Control of Complex Systems Using Neural Networks

Kumpati S. Narendra* Matthias J. Feiler† Zhiling Tian*

1 Introduction

1.1 Historical Background

The term artificial neural network (ANN) has come to mean any computer architecture that has massively parallel interconnections of simple processing elements. As an area of research it is of great interest due to its potential for providing insights into the kind of highly parallel computation performed by physiological nervous systems. Research in the area of artificial neural networks has had a long and interesting history, marked by periods of great activity followed by years of fading interest and revival due to new engineering insights [1]-[8], technological developments, and advances in biology. The latest period of explosive growth in pure and applied research in both real and artificial neural networks started in the 1980s, when investigators from across the scientific spectrum were attracted to the field by the prospect of drawing ideas and perspectives from many different disciplines. Many of them also believed that an integration of the knowledge acquired in the different areas was possible. Among these were control theorists like the first author, who were inspired by the ability of biological systems to retrieve contextually important information from memory, and process such information to interact efficiently with uncertain environments. They came to the field with expectations of building controllers based on artificial neural networks with similar information processing capabilities. At the same time they were also convinced that the design of such controllers should be rooted in the theoretical research in progress in different areas of mathematical control theory such as adaptive, learning, stochastic, nonlinear, hierarchical, and decentralized control.

1.2 Artificial Neural Networks (ANN)

From the point of view of systems theory, an artificial neural network (ANN-henceforth referred to as a neural network) can be regarded as a finitely parameterized, efficiently computable, and practically implementable family of transformations. The fact that they are universal approximators, involve parallel distributed processing, can be implemented in hardware, are capable of adaptation on-line, and easily applied to multivariable systems, made them attractive as components and subsystems in various applications. In the early 1980s, extensive computer simulation studies were carried out to demonstrate that such networks could approximate very well nearly all functions encountered in practical applications. As stated in their seminal paper, these claims led Hornik, Stinchcombe, and White to raise the question whether these were merely flukes, or whether the observed successes were reflective of some deep and fundamental approximating capabilities. During the late 1980s, as

*Center for System Science, Electrical Engineering Department, Yale University.
†Chair for Systems Design, ETH Zurich, Switzerland.

a result of the work of numerous authors [9]-[11] , it was conclusively shown that neural networks are universal approximators in a very precise and satisfactory sense. Following this, the study of neural networks left its empirical origins and became a mathematical discipline. Since approximation theory is at the core of many systems related disciplines, the new results found wide application of neural networks in such areas as pattern recognition, identification, and optimization and provided mathematical justification for them.

1.3 ANN for Control

Even as the above ground breaking developments in static optimization were taking place, it was suggested in 1990 [12] that feedforward neural networks could also be used as components in feedback systems, since the approximation capabilities of such networks could be used in the design of identifiers and controllers for unknown or partially known dynamical systems. This, in turn, gave rise to a frenzy of activity in the neural network control community, and numerous heuristic methods were proposed in the following years for the control of nonlinear processes. As in the case of function approximation, vast amounts of empirical evidence began to accumulate, demonstrating that neural networks could outperform traditional linear controllers in many applications. As in the past it once again became evident that more formal methods, grounded in mathematical systems theory, would have to be developed to quantitatively assess the capabilities as well as limitations of neurocontrol.

Linear Control and Linear Adaptive Control

The objective of control is to influence the behavior of dynamical systems. This includes maintaining the outputs of the system at constant values (regulation), or forcing them to follow prescribed time functions (tracking). The control problem is to use all available data at every instant and determine the control inputs to the system. Achieving fast and accurate control, while assuring stability and robustness in the presence of perturbations, is the aim of all control design. The best developed part of control theory deals with linear systems. The extensive results in linear control theory, and subsequently in linear adaptive control theory, have strongly influenced the evolution of neural network based control, and we recapitulate briefly in this section some of the principal concepts and results. Some further mathematical details concerning these are included in the following section.

Starting with the state description of linear systems, system theoretic properties such as controllability, observability, stability, stabilizability, and detectability were investigated in the 1960s and 1970s and the results were used to stabilize and control such systems using state feedback. Later, through the use of observers, the methods were extended to control both single-input single-output (SISO) and multi-input multi-output (MIMO) systems, in which all the state variables are not accessible. Many of the concepts and methods developed for the control of a single dynamical system were also extended to larger classes of systems where two or more subsystems are interconnected to achieve different objectives. In fact, current research in control theory includes many problems related to the decentralized control of linear systems using incomplete information about their interconnections.

Classical adaptive control deals with the control of linear time-invariant dynamical systems, when some of the parameters are unknown. Hence, all the theoretical advances in linear control theory were directly relevant to its development. In fact, the evolution of the field of adaptive control closely paralleled that of linear control theory, since the same control problems were attempted in the modified context. However, the principal difficulties encountered were significantly different, since adaptive systems are invariably nonlinear. During the period 1970-1980 the emphasis was on generating adaptive laws which would assure the stability of the overall system, and the asymptotic convergence of the performance of the adaptive system to that predicted by linear theory.

Control of Complex Systems

Our objective, as indicated by the title of the chapter, is to control complex systems using neural networks. The focus of the paper is on theoretical developments, primarily on the methods for generating appropriate control inputs. While the authors have carried out extensive simulation studies during the past 15 years to test many of these methods, no simulation results are included here.

In spite of numerous efforts on the part of researchers in the past, there is currently no universally accepted definition of a "complex system". Like many other terms in control theory, it is multi-faceted, and its definition cannot be compressed into a simple statement. At the same time, most researchers would agree on many of the characteristics that would make a system complex. Among these, the presence of nonlinear dynamics in the plant (or process) to be controlled, would be included as one of the more significant ones. The identification and control of an isolated nonlinear plant should therefore fall within the ambit of our investigations.

Other characteristics of complex systems would include uncertainties or time-variations in system behavior, and operation of the system far from equilibrium. Even if a stable equilibrium exists, the system may be prevented from approaching it by external disturbances or input signals in general. Complex systems are typically composed of many interconnected subsystems which mutually influence the evolution of their state variables. In some cases, the effect of the coupling dominates the dynamics of the system. Such problems have been studied extensively in the context of linear systems by means of matrix theory, and notions such as diagonal dominance have been coined to quantify the strength of the interconnections. The principal difficulty is, again, that the couplings may be nonlinear.

An additional source of complexity is the high dimensionality of the state and parameter spaces. In a large set of interconnected systems, the role of each individual system may be small, but together they constitute a powerful whole, capable of realizing higher–order functionality at the network level. This is sometimes referred to as emergent behavior and is one of the manifestations of complexity since it cannot be explained by simply aggregating the behaviors of the constituent systems. This means that the system cannot be modeled using "representative" variables of reduced dimensionality. The neural network itself is a prime example of such an interconnected system.

Nonlinear Adaptive Control: Stability and Design

It is a truism in control practice that efficient controllers for specific classes of dynamical systems can be designed only when their stability properties are theoretically well understood. This explains why controllers can be designed with confidence at the present time for both linear time-invariant plants with known parameters and those with unknown but constant parameters (i.e linear adaptive systems). The advantages of using neural networks as components in dynamical systems was stated earlier in this section. While the qualitative statements made in that context are for the most part valid and make neural networks attractive in static situations such as pattern recognition and optimization, their use in dynamical contexts involving control raises a host of problems which need to be resolved before they can be used with confidence. In the following sections, neural networks are primarily used as controllers in dynamical systems to cope with either known or unknown nonlinearities, making the overall system under consideration both nonlinear and adaptive. In spite of advances in stability theory for over two centuries, our knowledge of the stability of general nonlinear systems is quite limited. This makes the stability of nonlinear adaptive systems containing neural networks a truly formidable problem. Hence, while discussing the use of neural networks for identification and control, it is incumbent upon the authors to state precisely the class of plants considered, the prior information available to the designer concerning the system, the external perturbations that may be present, the domain of interest in the state space, and the manner in which new information concerning the unknown plant is acquired (i.e on-line or off-line) and utilized, and the conditions under which the results are valid.

Assumptions

Since the primary difficulty in most of the problems mentioned earlier arise due to the presence of nonlinearities in the representation of the system, it is not surprising that a wide spectrum of methods have been proposed in the literature by many authors making different assumptions. These assumptions determine the mathematical tractability of the problems, but at the same time also determine the extent to which the procedures developed will prove practically feasible. As is well known to experienced researchers, and succinctly stated by Feldkamp et al [13], apparently difficult problems can be made almost trivial by unreasonably optimistic assumptions.

1.4 Objectives of the Chapter

The first objective of the Chapter is to discuss in detail the methods that are currently available for the control of a nonlinear plant with unknown characteristics, using neural networks. In particular the chapter will examine the efforts made by different investigators to extend principles of linear control and linear adaptive control to such problems. It will examine the assumptions they have made, the corresponding approaches they have proposed, and the theoretical justification they provide for stability and robustness. In this context we also include our own approach to the same adaptive control problems, address the same issues mentioned earlier, and conclude with a statement concerning our position regarding the current status of the field of neurocontrol.

When the identifier and controller for a nonlinear plant are neural networks, we have the beginnings of interconnected neural networks. When many such are interconnected as described earlier, we have a network of neural networks. Since we believe that this is the direction in which the field is bound to evolve in the future, we include a typical problem for future investigation.

At the present time, there is considerable research activity in the use of neural networks in optimization and optimal control problems in the presence of uncertainty and we believe that it would be a great omission on our part if we failed to comment on it. We therefore devote a section to this important topic, merely to clarify the principal concepts involved.

Finally, our objective is to present and comment on some successful applications of the theory in practical control problems, as well as briefly touch upon some not so conventional applications that are currently under investigation which, if successful, will provide greater motivation for the use of neural networks in control.

1.5 Organization of the Chapter

In this chapter we attempt to discuss many of the issues related to neurocontrol that have arisen during the past fifteen years. Section 2 is devoted to mathematical preliminaries and includes results from linear and nonlinear control, as well as concepts for adaptive control that are useful for later discussions. The section concludes with a statement of the problems discussed in the chapter. Section 3 introduces feedforward and recurrent networks used to practically realize the control laws, and the methods used to update their parameters. Since neural network based control naturally leads to nonlinear control, and to nonlinear adaptive control when system characteristic are unknown, many of the current research problems are related to these areas. Results in nonlinear control theory, concepts and structures suggested by classical (linear) adaptive control, and the approximating capabilities of neural networks have to be judiciously combined to deal with the nonlinear adaptive control problems that arise in complex systems. Appropriate assumptions have to be made at every stage both to have well posed problems, as well as to make them mathematically tractable. These are contained in Section 4 which concludes with some critical comments concerning methods currently in vogue in the neurocontrol literature. In Section 5, global stabilizability questions are discussed, since the authors believe that such concepts are essential for our understanding of the nonlinear domain and will be increasingly encountered in neurocontrol in the future. Section 6 is devoted to optimization and optimal control over a finite time using neural networks. Finally, the current status of applications is discussed in Section 7.

2 Mathematical Preliminaries

Well known results from linear and nonlinear control that are used throughout the paper are presented in a condensed form in this section for easy reference. The section concludes with the statement of the identification and control problems that are investigated in the following sections.

2.1 Linear Time-invariant Systems

A general multiple input- multiple output (MIMO) linear time-invariant continuous-time system Σ_c (discrete-time system Σ_d) is described by the vector differential (difference) equation

$$\Sigma_c : \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array} \qquad \Sigma_d : \begin{array}{rcl} x(k+1) & = & Ax(k) + Bu(k) \\ y(k) & = & Cx(k) \end{array} \tag{1}$$

where $u(t) \in \mathbb{R}^r$, $y(t) \in \mathbb{R}^m$ and $x(t) \in \mathbb{R}^n$. A, B, and C are respectively constant matrices with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$ and $C \in \mathbb{R}^{m\times n}$, and $u(t)$, $y(t)$, and $x(t)$ are respectively the input, output, and the state of the system at time t. In the discussions that follow, we will deal with single-input single-output (SISO) systems (where $r = m = 1$) for clarity, and extend the results to the MIMO case. The SISO system is then described by the equation

$$\Sigma_c : \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + bu(t) \\ y(t) & = & cx(t) \end{array} \qquad \Sigma_d : \begin{array}{rcl} x(k+1) & = & Ax(k) + bu(k) \\ y(k) & = & cx(k) \end{array} \tag{2}$$

where b and c^T are constant vectors in $\mathbb{R}^n$.

Controllability, Observability, and Stability

Controllability, Observability, and Stability are system theoretic properties which play important roles in systems related problems. The following definitions and basic results can be found in standard text books on linear systems [14].

A system is said to be controllable if any initial state can be transferred to any final state by the application of a suitable control input. The SISO system $\Sigma_c(\Sigma_d)$ described in equation (2) is controllable if the matrix

$$W_c = [b, Ab, A^2 b, \cdots, A^{n-1} b] \tag{3}$$

is non-singular. The MIMO system $\Sigma_c(\Sigma_d)$ in equation (1) is controllable if the $(n \times nr)$ matrix $[B, AB, \cdots, A^{n-1}B]$ is of rank n.

The dual concept of controllability is observability. A system is said to be observable if the initial state (and hence all subsequent states) of the system can be determined by observing the system output $y(\cdot)$ over a finite interval of time. For an SISO system (2), the condition for observability is that the matrix W_o

$$W_o = [c^T, A^T c^T, \cdots, A^{(n-1)T} c^T] \tag{4}$$

be nonsingular. For MIMO systems, the $(n \times mn)$ matrix $[C^T, A^T C^T, \cdots, A^{(n-1)T} C^T]$ is of rank n.

The third system theoretic property which is crucial to all control systems is stability and depends on the matrix A. Σ_c is stable if the eigenvalues of A lie in the open left half plane (Σ_d is stable if the eigenvalues of A lie in the interior of the unit circle.)

Controllability and Stability: For LTI systems (2) it is known that if the pair (A, b) is controllable, it can be stabilized by state feedback i.e $u = k^T x$.

Estimation and Control: When $\Sigma_c(\Sigma_d)$ is represented by the triple (c, A, b) which is controllable and observable, an important result derived in the 1970s assures the existence of a control input that can stabilize the system. The state x of the system is estimated as $\hat{x}$ and used to determine the stabilizing input $u = k^T \hat{x}$.

ARMA Model

The proper representation of a discrete-time LTI system Σ_d in terms of only inputs and outputs is an important consideration in the mathematical tractability of many control problems. From equation (2), the following input-output relation can be obtained

$$y(k+n) = cA^n x(k) + \sum_{i=0}^{n-1} cA^{n-1-i} b \; u(k+i) \tag{5}$$

From the above equation the following ARMA (autoregressive moving average) representation can be derived for SISO systems

$$y(k+1) = \sum_{i=0}^{n-1} \bar{\alpha}_i y(k-i) + \sum_{j=0}^{n-1} \bar{\beta}_j u(k-j) \tag{6}$$

where $\bar{\alpha}_i$ and $\bar{\beta}_j$ are constants $(i, j = 1, 2 \cdots, n)$. The same also applies to MIMO systems where

$$y(k+1) = \sum_{i=0}^{n-1} \bar{A}_i y(k-i) + \sum_{j=0}^{n-1} \bar{B}_j u(k-j) \tag{7}$$

where $\bar{A}_i$ and $\bar{B}_j$ are constant $(m \times m$ and $m \times r)$ matrices.

If, in the SISO system (6) the input $u(k)$ at time k affects the output at time $(k+d)$ but not earlier, the system is said to have a relative degree d. For LTI systems this is merely the delay through the system). Since $cb, cAb, \cdots, cA^{d-1}b$ are zero, but $cA^d b \neq 0$, it can be shown that the system has a representation

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

where α_i and β_j are constants. For MIMO systems with m inputs and m outputs $(r = m)$, each output $y_i(\cdot)$ has a relative degree d_{ij} to the j-th input u_j. The relative degree d_i is then defined as $d_i = \min_j\{d_{ij}\}$, and represents the smallest time in which some input can affect the j-th output. Hence, each of the m outputs has a clearly assigned relative degree denoted by the elements of the vector $d = [d_1, d_2, \cdots, d_m]^T$. Using the same procedure as in the SISO case, we obtain the following input-output relation for the MIMO system

$$Y(k+d) = \begin{bmatrix} y_1(k+d_1) \\ y_2(k+d_2) \\ \vdots \\ y_m(k+d_m) \end{bmatrix} = \sum_{i=0}^{n-1} A_i y(k-i) + \sum_{j=0}^{n-1} B_j u(k-j) \tag{9}$$

where A_i and B_j are matrices of appropriate dimensions.

Minimum Phase Systems

A question that arises in control theory is whether internal signals in the system can become unbounded while the observed outputs remain bounded. In terms of equation (8), the question can be posed as follows: Is it possible for $\lim_{k\to\infty} y(k)$ to be zero while the input $u(k)$ grows in an unbounded fashion. Obviously such a situation is possible if

$$\beta_0 u(k) + \beta_1 u(k-1) + \cdots + \beta_n u(k-n) = 0 \tag{10}$$

has unbounded solutions. It can be shown that this is equivalent to the equation

$$\beta_0 z^n + \beta_1 z^{n-1} + \cdots + \beta_n = 0 \tag{11}$$

having at least one root outside the unit circle, or alternatively a necessary and sufficient condition for the question to have a negative answer is that all the roots of the equation (11) (representing the zeros of the transfer function of the SISO system) lie inside the unit circle. We refer to such a system as a "minimum phase system".

2.2 Nonlinear Systems

Finite dimensional continuous-time and discrete-time nonlinear systems can be described by the state equations of the form

$$\Sigma_c : \begin{array}{rcl} \dot{x}(t) & = & F(x(t), u(t)) \\ y(t) & = & H(x(t)) \end{array} \qquad \Sigma_d : \begin{array}{rcl} x(k+1) & = & F[x(k), u(k)] \\ y(k) & = & H[x(k)] \end{array} \tag{12}$$

Work in the area of nonlinear control has been in progress for many decades, and numerous attempts have been made to obtain results that parallel those in linear theory (refer to Section 5). We include here well established results concerning such systems which are related to their linearizations (refer to Section 4).

Controllability, Observability and Stability

The definitions of controllability, observability and stability in the nonlinear case are identical to those in the linear case. However obtaining general conditions to assure these properties in a domain D in the state space is substantially more complex.

Controllability: If the state $x(0)$ of the discrete-time system Σ_d in (12) is to be transferred to the state $x(n)$ by the application of a suitable input u, the following equation has to be satisfied.

$$x(n) = F[\cdots F[x(0), u(0)], u(1)] \cdots, u(n-1)] = \Phi[x(0), U_n(0)] \tag{13}$$

where $U_n(0) \triangleq \{u(0), u(1), \cdots, u(n-1)\}$ is an input sequence of length n. The problem of controllability at time $k = 0$ is evidently one of determining the existence of $U_n(0)$ which will satisfy equation (12) for any specified $x(0)$ and $x(n)$.

Observability: Similarly, observability can be defined by considering the equations

$$\begin{array}{rcl} y(k) & = & H[x(k)] = \Phi_1[x(k)] \\ y(k+1) & = & H[x(k+1)] = H[F[x(k), u(k)]] = \Phi_2[x(k), u(k)] \\ & \vdots & \\ y(k+n-1) & = & H[x(k+n-1)] = H[x(k+n-1)] = \Phi_n[x(k), u(k), u(k+1), , \ldots, u(k+n-2)] \end{array} \tag{14}$$

Given the sequence $Y_n(k) = \{y(k), y(k+1), \cdots, y(k+n-1)\}$ and the input $U_n(k) = \{u(k), u(k+1), \ldots, u(k+n-1)\}$ observability implies that the state $x(k)$ and hence $x(k+1), \ldots, x(k+n)$ can be determined.

Inverse Function Theorem and The Implicit Function Theorem

Both controllability and observability consequently involve the solutions of nonlinear algebraic equations. Two fundamental theorems of analysis that are useful in this context are the Inverse Function Theorem and the Implicit Function Theorem.

Inverse Function Theorem: Let U be an open set in $\mathbb{R}^n$ and let $f : U \to \mathbb{R}^n$ be a C^k function with $k \geq 1$. If a point $\bar{x} \in U$ such that the matrix $Df(\bar{x})$ is invertible, then there exists an open neighborhood of $\bar{x}$ in U such that $f : V \to f[V]$ is invertible with a C^k inverse.

Implicit Function Theorem: Let U be an open set in $\mathbb{R}^m \times \mathbb{R}^n$ and let $f : U \to \mathbb{R}^n$ be a C^k function with $k \geq 1$. Let $(\bar{x}, \bar{y}) \in U$ where $\bar{x} \in \mathbb{R}^m$ and $\bar{y} \in \mathbb{R}^n$ with $f(\bar{x}, \bar{y}) = c$. If the $(n \times n)$ matrix $D_y f(\bar{x}, \bar{y})$ of partial derivatives is invertible, then there are open sets $V_m \subset \mathbb{R}^m$ and $V_n \subset \mathbb{R}^n$ with $(x, y) \in V_m \times V_n \subset U$ and a unique C^k function $\phi : V_m \to V_n$ such that $f(x, \phi(x)) = c$ for all $x \in V_m$. Moreover $f(x, y) \neq c$ if $(x, y) \in V_m \times V_n$ and $y \neq \phi(x)$.

By the inverse function theorem if $\bar{x}$ is the solution of the vector equation $f(x) = c$, then the equation can also be solved in the neighborhood of $\bar{x}$ if the Jacobian matrix $Df(x)|_{x=\bar{x}}$ is nonsingular. The implicit function theorem extends this result to equations which are functions of x and y, and the solution y is desired as a unique function of x. The following important theorem derived using the implicit function theorem is the starting point of all the local results derived for nonlinear control, and is stated without proof.

Theorem: Let the linearized equations of (12) around the origin be

$$\begin{aligned} z(k+1) &= Az(k) + bu(k) \\ w(k) &= cz(k) \end{aligned} \tag{15}$$

where $A = \left.\frac{\partial f}{\partial x}\right|_{x=0,u=0}$, $b = \left.\frac{\partial f}{\partial u}\right|_{x=0,u=0}$ and $c = \left.\frac{\partial h}{\partial x}\right|_{x=0,u=0}$. If the linearized system (15) is controllable (observable), the nonlinear system (14) is controllable (observable) in some neighborhood of the origin.

Stability: If A in equation (15) is a stable matrix, from Lyapunov's works it is well known that the nonlinear system Σ_d is asymptotically stable in a neighborhood of the origin

The controllability and observability of the linearized system are merely sufficient conditions for the corresponding properties to hold for (12) and are not necessary. Yet the theorem is important, since according to it, the nonlinear system is well behaved in a neighborhood of the origin, if the linearized system is well behaved. The relevance of these comments will be made clear in Sections 4 and 5.

2.3 Adaptive Systems - Theoretical Considerations:

All the problems discussed in this paper can be considered as infinite-time or finite-time problems in nonlinear adaptive control. The term "adaptive control" refers to the control of partially known systems. Linear adaptive control deals with the identification and control of linear time-invariant systems with unknown parameters [15]. The class of nonlinear adaptive control problems of interest in this article are those in which the nonlinear functions $F(\cdot)$ and $H(\cdot)$ in the description of the controlled plant (12) are unknown or partially known. Obviously, this represents a very large class of systems for which general analytic methods are hard to develop. Many subclasses may have to be defined and suitable assumption may have to be made to render them analytically tractable.

In spite of the complexity of nonlinear adaptive systems, many of the questions that they give rise to are closely related to those in the linear case. While the latter seem simple (in hind sight), it is worth stressing that linear adaptive control gave rise to a multitude of difficult questions for a period of forty years and that some of them have not been completely answered thus far. Since the statement of the problems in the linear case, the assumptions made, the reasons for the difficulties encountered are all directly relevant for the issues discussed in this article, we provide a brief introduction to them in this section.

The Plant: to be controlled is described by the linear state equations (1) or (2) depending upon whether the system is MIMO or SISO. If only the inputs and outputs are accessible, the ARMA representations (5) and (6) are used. In all cases, the parameters of the plant are assumed to be unknown.

Identification and Control: Identification involves the estimation of the unknown parameters of the plant using either historical input-output data or on-line input-output measurements. Indirect adaptive control involves the adjustment of controller parameters based on the estimates of the plant parameters.

Comment 1: Parameters that are adjusted on-line become state variables. This also makes (linear) adaptive systems nonlinear. Hence, all system characteristics have to be discussed in a higher dimensional state space. For example the stability analysis of a linear plant of dimension n has to be discussed in a $3n$ dimensional space ($2n$ corresponding to the unknown parameters of the plant).

The Reference Model: Controller parameters can be adjusted either to minimize a specified performance criterion, or to track the output of a reference model (Model Reference Adaptive Control). In the latter case, the reference model has to be chosen properly so that the problem is well posed. Choosing a linear reference model to possess desired characteristics is relatively straightforward. Choosing a nonlinear reference model is considerably more difficult and requires a detailed knowledge of the plant. Therefore, it is not suprising that in most applications linear models are chosen.

Direct and Indirect Adaptive Control: In direct adaptive control the control input is determined directly from a knowledge of the output (control) error. To assure boundedness of all the signals strong assumptions such as positive realness of the plant transfer function have to be made. If indirect control is used, it must first be demonstrated that a controller structure exists which can result in the convergence of the output error to zero.

Comment 2: Existence questions in linear adaptive systems lead to linear algebraic equations (e.g the Diophantine equation). In nonlinear systems, the corresponding equations would be nonlinear.

Algebraic and Analytic Parts: All conventional adaptive methods invariably consist of two stages. Demonstrating the existence of a controller mentioned earlier, constitutes the the first stage which is algebraic. Determining adaptive laws for adjusting the controller parameters so that the overall system is stable and the output error tends to zero, constitutes the second stage and is the analytical part. Both stages are relevant for all the problems treated in this article.

Comment 3: The resolution of the above problem in linear adaptive control in the late 1970s took several years. This was in spite of the advantage that could be taken of many of the subsystems being linear. For example, the error models relating parametric errors to output errors are linear (refer Section 3.2). Since this advantage is lost in nonlinear adaptive control, the problem is significantly more complex.

Nonlinear Plants with a Triangular Structure: A more general class of nonlinear systems, whose adaptive control has been investigated rigorously in the literature are those which are in canonical form with constant unknown parameters. Two types of plants that have been analyzed are defined below [16, 17].

Definition: A system is said to be in parametric pure-feedback (PPF) form if

$$\begin{aligned} \dot{z}_i &= z_{i+1} + \theta^T \gamma_i(z_1, \cdots, z_{i+1}), \quad i = 1, 2, \ldots, n-1 \\ \dot{z}_n &= \gamma_0(z) + \theta^T \gamma_n(z) + [\beta_0(z) + \theta^T \beta(z)]u \end{aligned} \tag{16}$$

Definition: A system is said to be in parametric strict-feedback (PSF) form if

$$\begin{aligned} \dot{z}_i &= z_{i+1} + \theta^T \gamma_i(z_1, \cdots, z_i) \\ \dot{z}_n &= \gamma_0(z) + \theta^T \gamma_n(z) + \beta_0(z)u \end{aligned} \tag{17}$$

where $z = [z_1, z_2, \cdots, z_n]$, $\theta \in \mathbb{R}^p$ is a vector of unknown parameters.

Stabilizing adaptive controllers have been developed for systems in both PPF and PSF forms where the result is local in nature in the former and global in the latter.

Comment 4: The proof of stability given in [16, 17] for the above problems are strongly dependent on the fact that the nonlinear functions $\gamma_0(\cdot)$, $\beta_0(\cdot)$ and $\beta(\cdot)$ are known and smooth (so their derivatives can also be used in the control laws) and the only unknown in the system is the constant vector θ. Naturally, the proofs are no longer valid if any of the above assumptions do not hold (see comments in Section 4.4).

2.4 Problem Statement

As stated in the introduction, the historical developments in the fields of neurocontrol have traversed the same paths as those of linear adaptive control, even as the latter have closely parallelled those of linear control theory. In this section we consequently confine our attention to the same sequence of problems which were resolved in the two preceding fields in the past four decades. These are concerned with the identification and control of nonlinear dynamical systems.

The Plant: We assume that Σ the plant (or process) to be controlled is described by the discrete-time state equations (12):

$$\Sigma: \quad \begin{array}{rcll} x(k+1) & = & F[x(k),u(k)] & F(0,0)=0 \\ y(k) & = & H[x(k)] & H(0)=0 \end{array} \tag{18}$$

and the functions F and H are smooth.

If F and H are known. the problem belongs to the domain of nonlinear control. If F and H are unknown or partially known, the problem is one of nonlinear adaptive control. Naturally, as in the case of linear adaptive control, we will be interested first in the questions that arise in the control problem when F and H are known, and later on how the methods proposed can be modified for the adaptive case.

A number of factors influence both the problems posed and the methods used for resolving them. Among these the most important are the assumptions about the function $F(\cdot)$ and $H(\cdot)$, the stability of the system Σ, and the accessibility of its state variables. If control has to be carried out using only the inputs and outputs of Σ, a different representation of the system will be needed. This would naturally call for a modification of the methods used for identification and control.

Three problems are presented below. In the first two problems the plant is assumed to be stable. In the third problem, the plant is assumed to be unstable, and identification and control proceed concurrently to make the overall system stable (this corresponds to the major stability problem of adaptive control resolved in 1980). In all cases, it is assumed that external inputs are bounded with known bounds, and that the region in the state space in which the trajectories of Σ should lie are also specified.

Problem 1 (Identification) The discrete-time plant Σ is described by the equations (12) where $F(\cdot)$ and $H(\cdot)$ are unknown. The input $u(\cdot)$ satisfies the condition $||u(t)|| \leq c_u$ and Σ is BIBO stable, so that $||x(t)|| \leq c_x$ and $||y(t)|| < c_y$, where c_u, c_x, and c_y are known constants. The state $x(k)$ of Σ is accessible at every time-instant k.

(i) Determine a suitable representation for a model $\hat{\Sigma}$ of the plant whose output $\hat{x}(\cdot)$ satisfies the condition $\lim_{k\to\infty} ||x(k) - \hat{x}(k)|| < \epsilon_1$.

(ii) If $y(k)$ but not $x(k)$ is accessible, determine an input/ output model $\hat{\Sigma}_{I/O}$ of the system such that the output $\hat{y}(k)$ of the model satisfies $\lim_{k\to\infty} ||y(k) - \hat{y}(k)|| \leq \epsilon_2$ for the set of input-output pairs provided, where ϵ_1 and ϵ_2 are prescribed constants.

Problem 2 (Control of a Stable Plant) Assuming that Σ is stable and that models $\hat{\Sigma}$ and $\hat{\Sigma}_{I/O}$ satisfying the conditions of problem 1 have been determined, the following control problems may be stated:

(i) Determine a feedback control law $u(k) = \gamma(x(k))$, where $\gamma(\cdot)$ is a smooth function, such that every initial condition x_0 in a neighborhood of the origin, is transferred to the equilibrium state in a finite number of steps.

(ii) Assuming that only the input and output of Σ are accessible, determine a control law such that $x(k)$ tends to the equilibrium state in a finite time.

(iii) (Set point Regulation) In problems (i) and (ii) determine a control law such that the output $y(\cdot)$ of Σ is regulated around a constant value.

(iv) (Tracking) Given a stable reference model Σ_m defined by

$$\Sigma_m : \quad \begin{array}{rcl} x_m(k+1) & = & F_m(x_m(k), r(k)) \\ y_m(k) & = & H_m(x_m(k)) \end{array} \tag{19}$$

where $x_m(k) \in \mathbb{R}^n$ and $y_m(k) \in \mathbb{R}^m$, F_m and H_m are known, determine a feedback control law such that

$$\lim_{k\to\infty} |y(k) - y_m(k)| < \epsilon \tag{20}$$

where ϵ is a specified constant.

Problem 3 (Control of an Unstable Plant) In this case identification and control of the plant Σ (which is assumed to be unstable), have to be carried out simultaneously. All four cases stated in Problem 2 can also be considered in this case.

Comment 5: As in classical adaptive control we will be interested in both the algebraic and the analytical parts of the solution. These are discussed Section 4.1.

All the problems stated above can be addressed either from a strictly theoretical point of view or as those which arise in the design of identifiers and controllers in real applications. In this chapter, we are interested in both classes of problems. In the latter case, the prior information that is available concerning the plant as well as mathematical tractability will dictate to a large extent the models used for both identification and control. Some of the questions that arise in this context are listed below:

a) structures of identifiers and controllers and the use of feedforward networks and recurrent networks to realize them.

b) the algorithms used to adjust the parameters of the neural networks, and

c) the questions of stability that arise in the various cases.

An Area for Future Research:

In control theory as well as in adaptive control, after problems involving isolated systems had been addressed, interest invariably shifted to problems in which multiple dynamical systems are involved. Decentralized control, distributed control, and hierarchical control come under this category. More recently, there has been a great deal of research activity in multi-agent systems in which many dynamical systems interact. Also interest in distributed architectures has increased, since researchers in control theory and computer science believe that they would enhance our ability to solve complex problems. The above comments indicate that interaction of dynamical systems can arise due to a variety of factors ranging from practical physical considerations to desire for increased efficiency.

When dealing with interacting or interconnected systems neural networks play a critical role similar to that in the problems described earlier. A generic problem of interconnected nonlinear systems may be stated as follows. The overall system Σ consists of a set of subsystems Σ_i $(i = 1, 2, \cdots, N)$

$$\Sigma : \quad \begin{array}{rrcl} \Sigma_1 : & x_1(k+1) & = & f_1[x_1(k), h_1[\bar{x}_1(k)], u_1(k)] \\ \Sigma_2 : & x_2(k+1) & = & f_2[x_2(k), h_2[\bar{x}_2(k)], u_2(k)] \\ & \vdots & & \vdots \\ \Sigma_N : & x_N(k+1) & = & f_N[x_N(k), h_N[\bar{x}_N(k)], u_N(k)] \end{array} \tag{21}$$

where $\bar{N} = \sum_{i=1}^{N} n_i$, is the dimension of the state space Σ, $x_i \in \mathbb{R}^{n_i}$ is the state of subsystem Σ_i and $\bar{x}_i \in \mathbb{R}^{\bar{N}-n_i}$ denotes the states of the remaining $N-1$ systems. Each system Σ_i is affected by other subsystems through an unknown smooth function $h_i(\cdot)$. Depending upon the nature of the problem, the different subsystems may compete or cooperate with each other to realize their overall objectives. How the various systems identify their dynamics in the presence of uncertainty, how they acquire their information, and whether communication is permitted between them constitute different aspects of the problems that arise.

For mathematical tractability, much of the research in progress on problems of the type described above are restricted to linear systems. In Section 4, one such problem dealing with decentralized adaptive control is discussed. However, since most real systems are in fact nonlinear, it is only reasonable to expect increased interest in the future in nonlinearly interconnected systems.

3 Neural Networks, Adaptive Laws, and Stability

In the following sections neural networks are used as identifiers and controllers in dynamical systems. The type of networks to be used, the adaptive laws for adjusting the parameters of the network based on available data, and the stability and robustness issues that have to be addressed, are all important considerations in their design. In this section we comment briefly on each of the above aspects.

3.1 Neural Networks

While numerous network architectures have been proposed in the literature, we will be concerned mainly with two broad classes of networks in this chapter: (i) Feed Forward Networks and (ii) Recurrent Networks. The former are static maps while the latter are dynamic maps. Even though both of them have been studied extensively in the literature, for the sake of continuity, we provide brief introductions to both of them.

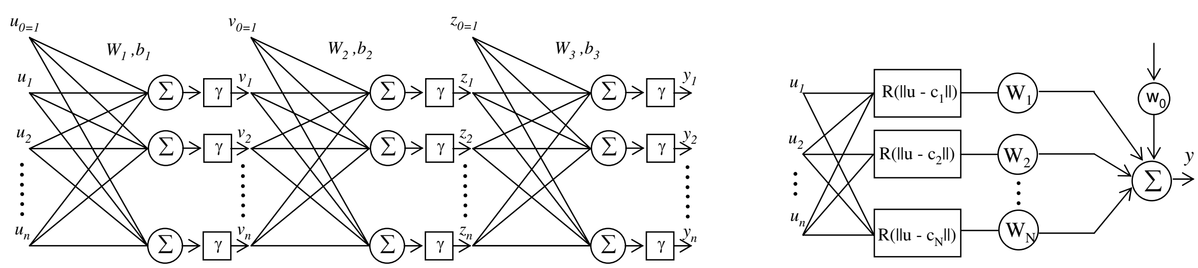
Feed Forward Networks

The most commonly used feedforward networks are the Multilayer Preceptron Network (MPN) and the Radial Basis Function Network. An N-layer MPN with input $u \in \mathbb{R}^n$ and output $y \in \mathbb{R}^n$ can be described by the equation

$$y = W_N \Gamma[W_{N-1} \cdots \Gamma[W_1 u + b_1] + b_2] + \cdots + b_{N-1}] + b_N \tag{22}$$

where W_i is the weight matrix associated with the i^{th} layer, the vectors $b_i (i = 1, 2, \cdots, N)$ represent the threshold values for each node in the i^{th} layer. Γ is a static nonlinear operator with an output vector $[\gamma(x_1), \gamma(x_2) \cdots, \gamma(x_n)]^T$ corresponding to an input $[x_1, x_2, \cdots, x_n]^T$, where $\gamma : \mathbb{R} \to [-1, 1]$ is a smooth function. A three layer networks is shown in Figure 1a. It is seen that each layer of the network consists of multiplications by constants (elements of the weight matrix) summation and the use of a single nonlinear map γ.

Radial basis function networks, which are an alternative to MPN, represent the output y as a weighted sum of basis (or activation) functions $R_i : \mathbb{R}^n \to \mathbb{R}$, where $i = 1, \ldots, N$. If $y \in \mathbb{R}$, the RBF network is described by $y = W^T R(u) + W_0$ where $W = [W_1, W_2, \ldots, W_N]^T$ is a weight vector multiplying the N basis functions having $u = [u_1, u_2, \ldots, u_n]^T$ as the input and w_0 is an offset weight. Quite often Gaussian functions are used as radial basis functions so that $R_i(u) = \exp\left[-\sum_{j=1}^{n} \frac{(u_j - c_{ij})^2}{2\,\sigma_{ij}}\right]$ where $c_i = [c_{i1}, c_{i2}, \ldots, c_{in}]$ is the center of the i^{th} receptive field, and σ_{ij} is referred to as the width of the Gaussian function. An RBFN is shown in Figure 1b. Since the function $R(u)$ is predetermined, the output is a linear function of the elements of W.



1a. A Multilayer Preceptron Network (MPN) 1b. A Radial Basis Function (RBF) Network

Figure 1: Neural Networks

For the purposes of this chapter, both MPN and RBFN enjoy two important characteristics. The first is their ability to approximate nonlinear maps. The second is the fact that for such networks, different methods of adjusting their parameters have been developed and are generally known. These methods will be discussed in Section 3.3.

Recurrent Networks

In contrast to the feedforward networks considered thus far which are static maps from one finite dimensional vector space to another, recurrent networks are dynamic maps which map input time signals into output time signals. This is accomplished by providing them with memory and introducing time–delays and feedback.

It is well known that any linear time-invariant discrete- time system can be realized using only the operations of multiplication by a constant, summation and a unit delay. A static feedforward network on the other hand includes multiplication by a constant and summation and a single appropriate nonlinear function (e.g. the sigmoid). Recurrent networks, which are nonlinear dynamic maps can be generated using all four operations described above i.e addition, multiplication by constant, delay and a sigmoid nonlinearity.

As in the case of static networks, interest in recurrent neural networks also grew from successes in practical applications. Through considerable experience, people in industry, became convinced of the usefulness of such networks for the modeling and control of dynamic systems. As shown later, recurrent networks provide a natural way of modeling nonlinear dynamical systems. It was also found that recurrent networks used as controllers are significantly more robust than feedforward controllers to changes in plant characteristics. It was argued by Williams [18] in 1990 that recurrent neural networks can be designed to have significantly new capabilities. Since then, it has been shown that recurrent networks can serve as sequence recognition systems, generators of sequential patterns, and nonlinear filters. They can also be used to transform one input sequence into another, all of which thus far have not been exploited in control theory.

Delays can be introduced anywhere in a feedforward network to make it dynamic. The number of such delays can vary from unity (if only the output signal is fedback to the input) to $\bar{N} = N^2$ where N represents the sum of the input, hidden, and output nodes (and delays exist between every node and all the other nodes in the system). For practical as well as theoretical reasons, compromises have to be made on the total number of delays used. Many different structures have been suggested in the neural network literature by both computer scientists and engineers. We present only two structures which will be needed for addressing the problems stated earlier. Both of them use the universal approximating property of multilayer neural networks. Consider first the general state equation (18) representing an n^{th} order nonlinear system. $F : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$ can be approximated using a multilayer neural network with $(n+r)$ inputs and n outputs, and delays as shown in Figure 2. Similarly, using a separate multilayer network, the function $H : \mathbb{R}^n \to \mathbb{R}^m$ can be approximated. The representation of the dynamical system given by equation (18) is shown in Figure 2. $\hat{F}$ and $\hat{H}$ represent the approximations of F and H respectively.

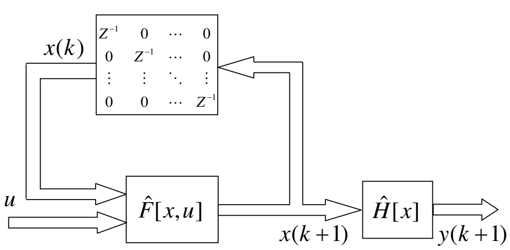


Figure 2: State Vector Model

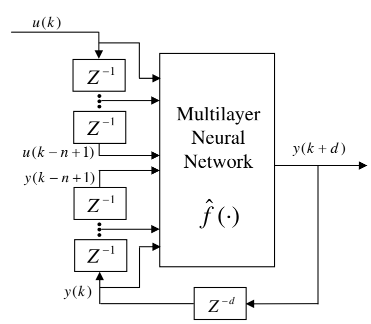


Figure 3: Input/Output Model

If the state variables are not accessible and an input-output model $\Sigma_{I/O}$ of the system (with relative degree d) is needed, it has been shown [19, 20] that an SISO system can be described by the equation.

$$y(k+d) = f[y(k), y(k-1), \cdots, u(k), \cdots, u(k-n+1)] \tag{23}$$

in a neighborhood Ω of the equilibrium state. Similarly, for a multivariable system with r inputs ($u(k) \in \mathbb{R}^r$) and m outputs $y(k) \in \mathbb{R}^m$, and relative degree d_i for the i^{th} output, it has been shown that a representation of the form

$$\begin{array}{rcl} y_1(k+d_1) & = & f_1[y(k), y(k-1), \cdots, y(k-v+1), u(k), u(k), \cdots, u(k-v+1)] \\ & \cdots & \\ y_m(k+d_m) & = & f_m[y(k), y(k-1), \cdots, y(k-v+1), u(k), u(k), \cdots, u(k-v+1)] \end{array} \tag{24}$$

exists in Ω. These are referred to as NARMA (Nonlinear ARMA) models.

In Figure 3, the realization of an SISO system is shown using tapped delay lines. The multivariable system (24) can also be realized in a similar fashion. $\hat{f}$ represents an approximation of f in equation (23). The recurrent network models shown in Figures 2 and 3 can be used either as identifiers or controllers in the problems stated earlier.

System Approximation

The two principal methods for approximating a system described by a recursive equation can be illustrated by considering the estimation of the parameters of a linear systems, described by equation (8)

$$y(k+d) = \sum_{i=0}^{n-1} \alpha_i y(k-i) + \sum_{j=0}^{n-1} \beta_j u(k-j) \tag{25}$$

where α_i and β_j are unknown and need to be estimated. A series-parallel identification model has the form

$$\hat{y}(k+d) = \sum_{i=0}^{n-1} \hat{\alpha}_i(k) y(k-i) + \sum_{j=0}^{n-1} \hat{\beta}_j(k) u(k-j) \tag{26}$$

where $\hat{\alpha}_i(k)$ and $\hat{\beta}_j(k)$ are the parameter estimates at time k. The output error equation has the simple form

$$\tilde{y}(k+d) = \sum_{i=0}^{n-1} \tilde{\alpha}_i(k) y(k-i) + \sum_{j=0}^{n-1} \tilde{\beta}_j(k) u(k-j), \tag{27}$$

where $\tilde{y}$, $\tilde{\alpha}_i$, and $\tilde{\beta}_j$ represent the output and parameter errors at time k. Since this has the standard form of error model 1 (described in Section 3.2), stable adaptive laws for adjusting $\hat{\alpha}_i(k)$ and $\hat{\beta}_j(k)$ can be determined by inspection. If however a recurrent (or parallel) identification model is used, the equation describing the model is no longer simple, and is a difference equation as shown below

$$\hat{y}(k+d) = \sum_{i=0}^{n-1} \hat{\alpha}_i(k) \hat{y}(k-i) + \sum_{j=0}^{n-1} \hat{\beta}_j(k) u(k-j) \tag{28}$$

since the estimate $\hat{y}(k+d)$ at time $k+d$ depends upon past estimates $\hat{y}(k)$, $\hat{y}(k-1)$,$\cdots$, $\hat{y}(k-n+1)$. The determination of stable adaptive laws for adjusting $\hat{\alpha}_i(k)$ and $\hat{\beta}_j(k)$ is substantially more complex in this case. In fact, such adaptive laws are not available and only approximate methods are currently known.

Comment 6: The following points are worth emphasizing. The series-parallel model is not truly a model but merely a predictor. In contrast to this, the recurrent model is a true model of the system with all the advantages of such a model (e.g. control strategies can be tried out on the model rather than the plant). If an efficient predictor is adequate for control purposes (as has been demonstrated in linear adaptive control) the simplicity of the series-parallel model, and the assured stability of the overall control system, based on the adjustment of the control parameters using plant parameter estimates, may outweigh the theoretical advantages of the recurrent model in some applications.

3.2 Stable Adaptive Laws: Error Models

The laws for adjusting the parameters derived in classical adaptive control are based on simple linear models, known as error models. These relate the parameter errors $\phi(t)$ to the output error $e(t) \in \mathbb{R}(\mathbb{R}^m$ for MIMO) between the actual output of the plant and a desired output (generally the output of a reference model). The objective is to determine $\dot{\phi}(t)$ using all the information available at time $'t'$ to make the system stable, so that $e(t)$ tends to zero. The study of error models is rendered attractive by the fact that by analyzing these models, which are independent of specific applications, it is possible to obtain insights into the behavior of a large class of adaptive systems.

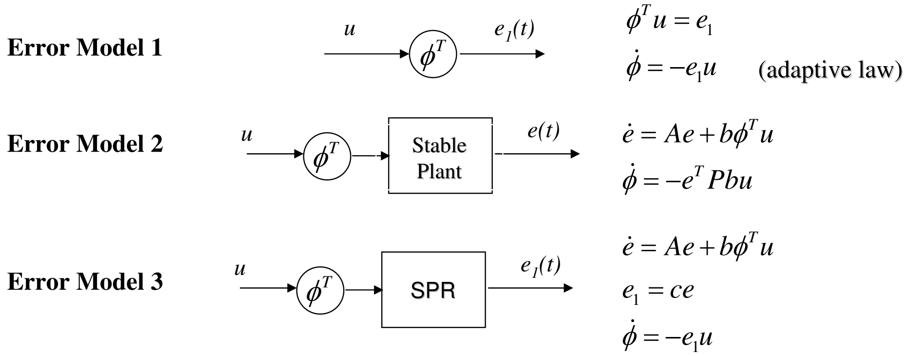


Figure 4: Error Models

The error models [15] are shown in Figure 4. The equations describing the models, the adaptive laws proposed, and the Lyapunov functions which assure stability in each case are given below.

<u>Error Model 1:</u> $\phi(t), u(t) \in \mathbb{R}^n$ and $\phi^T(t)u(t) = e(t)$.

Adaptive Law $\dot{\phi}(t) = -e(t)u(t)$ if the input $u(t)$ is bounded and $\dot{\phi}(t) = \frac{-e(t)u(t)}{1+u^T(t)u(t)}$ when it is not known a priori that $u(t)$ is bounded. $V(\phi) = \frac{1}{2}\phi^T(t)\phi(t)$ and $\dot{V}(t) = -e^2(t) \left(\text{or } \frac{-e^2(t)}{1+u^T(t)u(t)}\right) \leq 0$.

<u>Error Model 2:</u>

$$\dot{e}(t) = Ae(t) + b\phi^T(t)u(t) \tag{29}$$

where the matrix A and vector b are known, A is stable, and (A, b) is controllable.

Adaptive Law:

$$\begin{aligned} \dot{\phi}(t) &= -e^T(t)Pbu(t) \left(\text{or } \frac{-e^T(t)Pbu(t)}{1+u^T(t)u(t)}\right) \\ A^TP + PA &= -Q < 0 \end{aligned} \tag{30}$$

<u>Error Model 3:</u> In error model 2 $e(t)$ is not accessible, but $ce(t) = e_1(t)$ is accessible and $c[sI-A]^{-1}b$ is strictly positive real.

Adaptive Law: $\dot{\phi}(t) = -e_1(t)u(t) \left(\text{or } \frac{-e_1(t)u(t)}{1+u^T(t)u(t)}\right)$

$$V(e,\phi) = e^T Pe + \phi^T \phi \quad \text{and} \quad \dot{V}(t) = -e^T Qe \tag{31}$$

By the Kalman-Yakubovich Lemma [21], a matrix P exists which simultaneously satisfies the equations $A^T P + PA = -Q$ and $Pb = c^T$.

It is worth emphasizing that when the input $u(\cdot)$ is not known a priori to be bounded, the adaptive laws have to be suitably normalized.

Error Models for Nonlinear Systems

In some simple nonlinear adaptive control problems as well as more complex problems in which appropriate simplifying assumptions are made, it may be possible to obtain an error model of the form shown in Figure 5. This is the same as error model 3 in which $u = N(x)$ where x is the state of the unknown plant and $N(\cdot)$ is a continuous nonlinear function. If the same adaptive law as in error model 3 $\dot{\phi}(t) = -e_1(t)N(x(t))$ is used, by the same arguments as before, it follows that e and ϕ are bounded. This ensures that x and consequently $N(x)$ are bounded, and hence the output e_1 tends to zero.

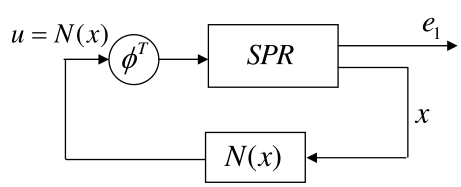


Figure 5: Error Models for Nonlinear Systems

<u>Comment 7:</u> Approximations of the type described in this model have been made widely in the neurocontrol literature without adequate justification. We shall comment on these at the end of Section 4.

Gradient Based Methods and Stability

During the early days of adaptive control in the 1960s, the adjustment of the parameters of an adaptive system were made to improve performance. Extremum seeking methods and sensitivity methods which are gradient based, were among the most popular methods used, and in both cases the parameters were adjusted on-line. Once the adjustment was completed, the stability of the overall system was analyzed, and conditions were established for local stability. Thus, optimization of performance preceded stability analysis in such cases.

In 1966 Parks [22], in a paper of great historical significance in adaptive control, conclusively demonstrated using a specific example that gradient methods for adjusting the parameters in an adaptive system can result in instability. At the same time he also showed that the system could be made globally stable using a design procedure based on Lyapunov's method. This clear demonstration that gradient based methods could become unstable, tolled the death knell of such systems, and witnessed a shift in the next decade to design based on stability methods described earlier. In the following forty years adaptive control aimed at first stabilizing the overall system using stable adaptive laws, and later adjusting the fixed controller parameters of the system to improve performance within a stability framework.

When neural networks are used in a system for identification and control, the overall system is nonlinear and it is very hard to derive globally stable adaptive laws for adjusting parameters. However, as shown in Section 4, numerous authors have continued to formulate problems in such a fashion that stable adaptive laws can still be determined.

In view of the difficulties encountered in generating stable adaptive laws most of the methods currently used for adjusting the parameters of a neural network are related to back propagation, and are gradient based as in classical adaptive control of the 1960s. Once again, it becomes necessary, to examine the reasons for discarding gradient methods in the past and explore ways of reconciling them with stability and robustness of the overall system.

In the example described by Parks in his classic paper, and later verified in numerous applications, it is the speed of adaptation that causes instability. If the frequency of adjustment of the parameters is such that the output error has a phase shift greater that $\frac{\pi}{2}$, adaptation may proceed exactly in a direction opposite to what is desired. Also, we note that in all the error models discussed earlier $\dot{\phi} \to 0$. Motivated by such considerations, researchers have been reexamining gradient based methods which are both theoretically and practically attractive. In the following sections we shall assume that gradient methods result in stability if the operating point is stable, and the adjustments are slow compared to the dynamics of the system.

3.3 Adjustment of Parameters: Feedforward and Recurrent Networks

The simplicity of the adaptive laws for adjusting the parameters of the series-parallel model described in earlier resulted from the fact that the error equation (relating parameter errors to output error) was linear. When the neural networks shown in Figure 1 are used to approximate a nonlinear function, the parameters of the networks are adjusted to minimize some specified norm $||y - y_d||$ where y is the output of the neural network and y_d the output of the given nonlinear function when both have the same input u. For radial basis function networks if w_i are adjusted, $\frac{\partial e}{\partial w_i}$ is merely the output of the i^{th} radial basis function network. For multilayer networks the element θ_i of the parameter vector θ (the elements of the matrices W_i and vectors b_i in Figure 1a) are in general nonlinearly related to the output error. While the output error depends linearly on the weights (W_3, b_3) in the output layer (if the activation functions $\gamma(\cdot)$ are omitted at the output), they depend nonlinearly on (W_2, b_2) and (W_1, b_1). Hence, one is forced to resort to gradient methods for adjusting these parameters. If w is a typical weight $\frac{\partial e^2}{\partial w} = 2e\frac{\partial e}{\partial w}$ so that $\frac{\partial e}{\partial w}$ has to be computed. Hence, a convenient method of computing $\frac{\partial e}{\partial \theta_i}$ is needed. "Back propagation" is such a method and is currently well established.

In [12], an architecture was proposed by which the partial derivatives of e with respect to parameters in an arbitrary number of layers can be realized. This is shown Figure 6. Since our interest is in the control of complex systems using neural networks, and the latter quite often involves the cascading of several networks, the architecture in Figure 6 is found to be very useful in practical applications.

Back propagation has been perhaps the most diversely used adaptive architecture in practical applications. In problems that do not involve feedback (such as function approximation and pattern recognition) it is simple to apply and does not pose any stability questions. If it is used in control problems, as described later, the adjustment of the parameters must be slow compared to the

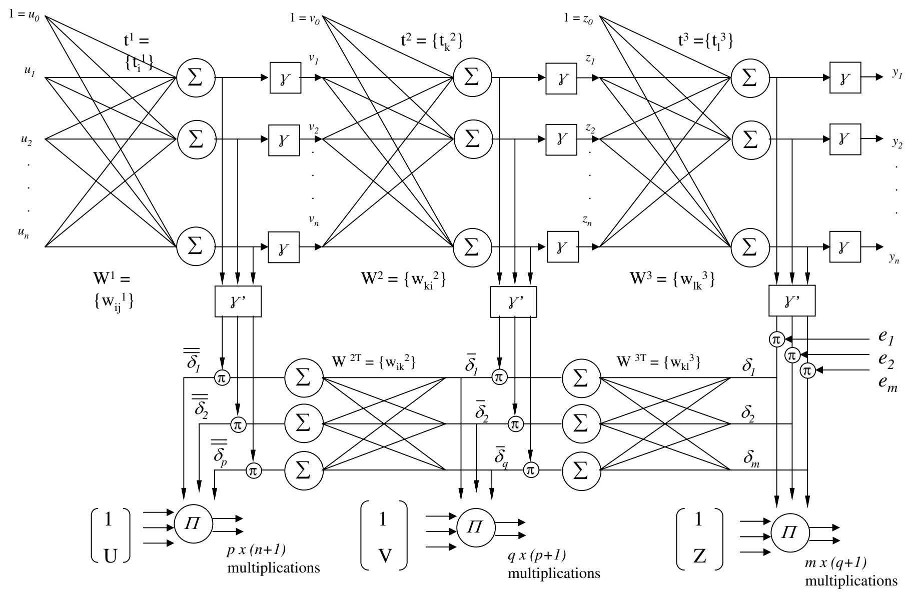


Figure 6: Architecture for Back Propagation

dynamics of the controlled system.

In contrast to the above, the recurrent network, as stated earlier, is a dynamic map and the inputs and outputs are not vectors in a finite dimensional vector space but time sequences. Mathematically they are substantially more complex than feedforward networks, but it is their very complexity that makes them attractive since they permit a wide variety of dynamical behaviors to be programmed by the proper choice of weights. In fact, it has been shown that a recurrent network with enough units can approximate any dynamical system [23]. Even though the mathematical tools for dealing with such networks are not sufficiently well developed, it is not surprising that they are being widely studied at the present time and there is every indication that their use in the control of complex systems is bound to increase in the future.

The problem of adjusting the parameters of a recurrent network for approximating a desired behavior has been addressed using supervised learning based on an output error signal, reinforcement learning based on a scalar reward signal, and unsupervised learning based on the statistical features of the input signal. Our interest here is in supervised learning algorithms. A network with a fixed structure (such as the ones described earlier) and a set of parameters, inputs, and desired outputs are specified. An error function is defined, and it is the gradient of this function with respect to the parameters that is desired.

Numerous authors in engineering and other fields (including computer science and computational neuroscience) have proposed different algorithms for adjusting the parameters of recurrent networks, and we refer the reader to papers by Pearlmutter [24] Rumelhart et al [8, 25] on this subject. In the engineering literature, the problem was considered independently by Werbos [26], Narendra and Parthasarathy [27] and Williams and Zipser [28] in the early 1990s. Werbos refers to it as “back propagation through time”, Narendra and Parthasarathy as “dynamic back propagation” and Williams and Zipser as “real-time recurrent learning”. In spite of the different origins and terminologies, the basic ideas are essentially the same and involve the computation of partial derivatives of time functions with respect to parameters in dynamical systems. A brief description of the principal ideas contained in each of the approaches is given below, and the reader is referred to the source papers for further details.

Back Propagation Through Time

The basic idea of this approach is that corresponding to every recurrent network it is possible to construct a feedforward network with identical behavior over a finite number of steps. The finiteness of the time-interval permits the neural network to be unfolded into a multilayer network, so that standard back propagation methods can be used for updating the parameters. Back propagation through time was first described in 1974 by Werbos and was rediscovered independently by Rumelhart et al (1986).

The principal idea of the method is best described by considering two steps of a recursive equation

$$x(k+1) = N(x(k), u(k), \theta) \tag{32}$$

where θ is an adjustable parameter vector. The system, over the instants 0,1,2 can be represented as shown in Figure 7.

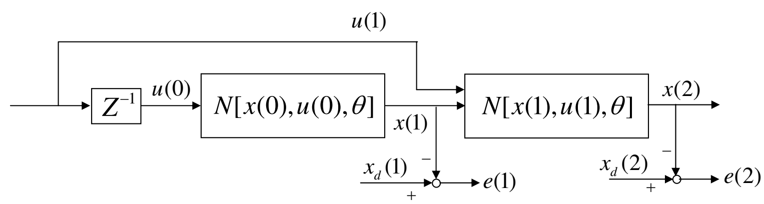


Figure 7: Back Propagation Through Time

If $u(0)$ and $u(1)$ are known and $x(0)$ is specified, the states $x(1)$ and $x(2)$ can be computed. If the desired states $x_d(1)$ and $x_d(2)$ are specified, static back propagation over two steps can be carried out to determine the gradient of an error function with respect to θ, and based on that, the parameter θ can be adjusted to decrease the error function over the interval.

The above method can be readily extended to a finite number of steps. Since the input, output and state information that has to be stored grows linearly with time, the interval over which optimization is carried out must be chosen judiciously from practical considerations. Having chosen the interval, the states are computed for the specified values of the inputs and the gradients computed using back propagation.

Comment 8: The architecture for back propagation shown in Figure 6, which applies to an arbitrary number of layers, is a convenient aid for computing the necessary gradient while using this method.

Dynamic Back Propagation

The origin of real-time recurrent learning may be traced back to a paper written in 1965 by McBride and Narendra [29] on "Optimization of time-varying systems". Determining gradients of error functions with respect to adjustable parameters, and adjusting the latter along the negative gradients was well known in the 1960s. This was extended to time-varying systems in the above paper and it was shown that it led to a whole gamut of optimization procedures ranging from self-optimizing (quasi) stationary systems to optimal programming and optimal control.

A procedure for determining the gradient of a performance index with respect to the parameters of a nonlinear dynamical system was proposed by Narendra and Parthasarathy in 1991 [27]. This work was naturally strongly influenced by numerous papers written in the 1960s by Narendra and McBride on gradient based methods for the optimization of linear systems. The main idea is best illustrated by a simple example of a continuous-time system (similar results can be readily obtained for discrete-time systems, as well as multivariable systems).

Let a second order system be described by the equation

$$\ddot{y} + F(\alpha, \dot{y}) + y = u \quad y(0) = y_1 \quad y'(0) = y_2 \tag{33}$$

The objective is to determine the value of α which minimizes

$$J(\alpha) = \frac{1}{T}\int_0^T [y(\tau,\alpha) - y_d(\tau)]^2 d\tau = \frac{1}{T}\int_0^T e^2(T,\alpha)d\tau \tag{34}$$

where $e(t,\alpha)$ is the error between $y(t,\alpha)$ and a specified desired time function $y_d(t)$. A gradient based method for adjusting α can be implemented if $\frac{\partial e(t,\alpha)}{\partial \alpha}$ $(= \frac{\partial y(t,\alpha)}{\partial \alpha}$) can be computed over the interval $[0,T]$.

Differentiating equation (33) with respect to α and denoting $\frac{\partial y(t,\alpha)}{\partial \alpha} = z$, we have (using $\frac{\partial F(\alpha,\dot{y})}{\partial \alpha} = \frac{\partial F}{\partial \alpha} + \frac{\partial F}{\partial \dot{y}}\frac{\partial \dot{y}}{\partial \alpha}$)

$$\ddot{z} + \frac{\partial F}{\partial \dot{y}}\dot{z} + z = -\frac{\partial F}{\partial \alpha} \tag{35}$$

If a time-varying sensitivity model described by equation (35) can be constructed, z (the desired gradient) can be generated as its output. When the parameters of a neural network need to be adjusted, the desired partial derivatives $\frac{\partial F}{\partial \dot{y}}$ and $\frac{\partial F}{\partial \alpha}$ are known signals, so that the desired gradient z, and the change in the parameter α can be computed at every instant of time.

Comment 9: The adjustment of α is assumed to be slow compared to the dynamics of the system and the methods were referred to as quasi-stationary methods in the 1960s.

Interconnection of LTI systems and Neural Networks

The method described above for determining the partial derivatives of the error functions with respect to the adjustable parameters in a recurrent network was used widely in the 1960s for optimizing LTI system. Since the methods used are identical in the two cases Narendra and Parthasarathy [27] suggested the use of dynamic backpropagation for use in complex systems in which LTI systems and static multilayer networks are interconnected in arbitrary configurations.

In all cases the method calls for the construction of a dynamic sensitivity model whose outputs are the desired partial derivatives.

Real-Time Recurrent Learning

In 1989 Williams and Zipser [28] suggested a method very closely related to dynamic back propagation described earlier for discrete-time recurrent networks.

Let a recurrent network consist of r inputs $u_i(\cdot)$ $(i = 1, 2, \cdots, r)$ and n state variables x_i $(i = 1, 2, \cdots, n)$ which are related by the state equations

$$\gamma(\sum_{j=1}^{n} w_{1,ij}x_j(n) + \sum_{j=1}^{r} w_{2,ij}u_j(n)) \tag{36}$$

or equivalently by the equation

$$x_i(n+1) = \gamma(\sum_{j=1}^{n+r} w_{ij} z_j(n)) \tag{37}$$

where $z_j = \begin{cases} x_j & 1 \le j \le n \\ u_{j-n} & j > n \end{cases}$ and γ is a squashing function.

The objective is to determine the weights w_{ij} so that the state $(x_1(n), x_2(n), \cdots, x_n(n))$ follows a desired trajectory $(x_{1d}(n), x_{2d}(n), \cdots, x_{nd}(n))$. To adjust the weights, the partial derivatives of $x_i(n)$ with respect to the weights have to be determined.

If w_{kl} is a typical weight, the effect of a change in it on the network dynamics can be determined by taking partial derivatives of both sides of equation (36). This yields

$$\frac{\partial x_i(n+1)}{\partial w_{kl}} = \gamma'(y_i(n)) \left[\sum_{j=1}^{n} w_{ij} \frac{\partial x_j(n)}{\partial w_{kl}} + \delta_{ik} z_l(n) \right] \qquad (i = 1, 2, \cdots, n) \tag{38}$$

where $y_i(n) = \sum_{j=1}^{n+m} w_{ij} z_j(n)$ and δ_{ik} is Kronecker's delta ($\delta_{ik} = 1,\ i = k$ and 0 otherwise). The term $\delta_{ik} z_l(n)$ represents the explicit effect of the weight w_{kl} on the state x_i, and the first term in the brackets the implicit effect on the state due to network dynamics. Equation (36) is a linear time-varying difference equation (which corresponds to the sensitivity network described in dynamic backpropagation) which can be used to generate the required partial derivatives in real time.

4 Identification and Control Methods

In this section, we consider several different methods which have been proposed for addressing identification and control problems of the form posed in the preceding section. The first method is based on linearization and represents the only one that we endorse without reservation. We discuss the approach in detail and comment on its practical implementation as well as its limitations. These comments are based on extensive simulation studies carried out in numerous doctoral thesis at Yale and the Technical University of Munich [30]-[35] carried out under the direction of the first author, but are not included here due to space limitations.

In addition to the above, alternate methods proposed by other researchers are included in this section. These make different assumptions concerning the representation of the plant. We comment on and raise both theoretical and practical issues related to these methods. Even though only some methods are presented here, they subsume to a large extent most of the approaches that have appeared in the neurocontrol literature. The latter are included in the list of references contained at the end of the chapter.

4.1 Identification and Control Based on Linearization

A vast body of literature currently exists on linear systems and linear control systems. The introductory chapters of most text books on these subjects emphasize the fact that almost all physical systems are nonlinear and that linear systems are obtained by linearizing the equations describing the system around an equilibrium state (linear time-invariant system (LTI)) or around a nominal time-varying trajectory (linear time-varying system (LTV)) so that they are valid in some neighborhood of the equilibrium state or trajectory. The fact that linear systems analysis and synthesis has proved extremely useful in a large number of practical applications attests to the fact that the neighborhoods in which the linearizations are good approximations of the dynamical systems are not always small. In this section we attempt, using similar methods, to obtain a more accurate representation of the dynamical system in a neighborhood of the equilibrium state, by including nonlinear terms which are small compared to the linear terms. Such representations enjoy many of the theoretical advantages of linear systems, while at the same time permitting improvements in performance using neural networks to compensate for nonlinearity.

System Representation

The developments in this section essentially follow those reported in [36] by Chen and Narendra. These are used to derive the main results developed at Yale during the period 1990-2004.

A nonlinear system Σ is described by equation (12). The local results for Σ are based on the properties of its linearized system Σ_L described by

$$\Sigma_L: \begin{array}{rcl} x(k+1) & = & Ax(k) + Bu(k) \\ y(k) & = & Cx(k) \end{array} \tag{39}$$

where $A = \frac{\partial F}{\partial x}|_{(x=0,u=0)}$, $B = \frac{\partial F}{\partial u}|_{(x=0,u=0)}$ and $C = \frac{\partial H}{\partial x}|_{(x=0,u=0)}$ are respectively the Jacobian matrices of F and H with respect to x and u. We now rewrite the actual system equations (12) as

$$\Sigma: \begin{array}{rcl} x(k+1) & = & Ax(k) + Bu(k) + f(x(k), u(k)) \\ y(k) & = & Cx(k) + h(x(k)) \end{array} \tag{40}$$

where f and h are called "higher order functions". A block diagram representation of system (40) is seen in Figure 8.

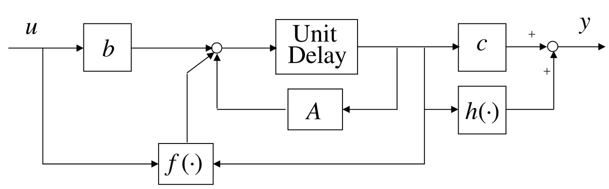


Figure 8

The representation shown in Figure 8 highlights the role played by the linear system in the synthesis of controllers, and suggests (as shown in this section) how methods proposed for linear systems can be modified for identifying and controlling the nonlinear plant, in some neighborhood of the equilibrium state.

Using the approach proposed, linear identifiers and controllers are first designed before an attempt is made to include nonlinear terms. Since this is in general agreement with procedures followed by the neurocontrol community, the approach has both pedagogical and practical interest.

Higher Order Functions

We shall address all the problems stated in Section 2 in the context of the nonlinear system Σ. In all cases we use the Inverse Function Theorem and the Implicit Function Theorem as the principal tools and indicate how the assumptions concerning the HOF, permit the application of the theorems.

Since the problems invariably lead to the addition, multiplication, and composition of "higher order functions", as functions of their argument, we formally define them as follows:

Definition: A continuously differentiable function $G(x) : \mathbb{R}^n \to \mathbb{R}^m$ is called a "Higher Order Function" if $G(0) = 0$ and $\frac{\partial G}{\partial x}|_{x=0} = 0$. We denote this class by $\mathcal{H}$.

Thus any smooth function can be expressed as the sum of a linear function and a higher order function. The following properties of functions in $\mathcal{H}$ are found to be useful and can be verified in a straightforward fashion: (i) If A is a constant matrix and $F(\cdot) \in \mathcal{H}$, then $AF(\cdot) \in \mathcal{H}$. (ii) If F_1, $F_2 \in \mathcal{H}$ then F_1F_2, $F_1 + F_2 \in \mathcal{H}$. (iii) If $F_1 \in H$ and $F_2(0) = 0$ and is continuously differentiable, $F_1F_2(\cdot) \in \mathcal{H}$.

If in some neighborhood $U_1 \subset \mathbb{R}^n$ of the origin the equation

$$Ax + f(x) = y \qquad A \in \mathbb{R}^{n\times n}, x \in \mathbb{R}^n \tag{41}$$

is defined, where A is nonsingular, by the Inverse Function Theorem (IFT), there exists a neighborhood $U \subset U_1$, containing the origin such that $V = AU + f(U)$ is open, and for all $x \in U$

$$x = A^{-1}y + g(y) \qquad g(\cdot) \in \mathcal{H}. \tag{42}$$

Similarly, it can be shown that if $U_1 \subset \mathbb{R}^{n+k}$ is an open set containing the origin, an element of U_1 is denoted by (x, y) with $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^k$, and $F(x, y) = Ax + By + f(x, y)$, A is nonsingular and $f(\cdot) \in \mathcal{H}$ is a function from U_1 to $\mathbb{R}^n$, then by the implicit function theorem there exists an open set $U \subset \mathbb{R}^k$ containing the origin such that

$$x = A^{-1}By + g(y) \quad y \in U \;\; g(\cdot \in \mathcal{H}) \tag{43}$$

satisfies the equation $F(x, y) = 0$.

From the above it is seen that when the functions involved in the equations are either linear or belong to $\mathcal{H}$ inverses can be obtained in some neighborhood of the origin. It is this fact that is exploited throughout this section. The above two results can be shown in block diagram form as in [36].

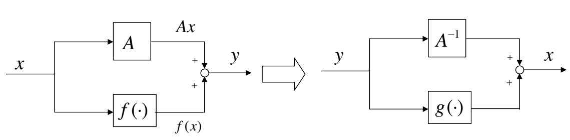


Figure 9

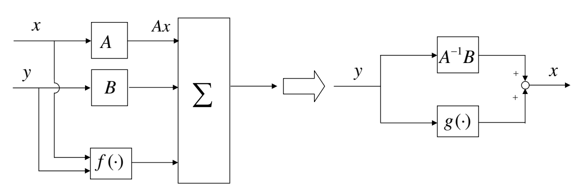


Figure 10

It is the existence of functions $g(\cdot)$ in the neighborhoods around the origin, which can be used in the inverse operation that provides an analytical basis for all the results that have been derived in [36]. We refer the reader to that paper for further details.

System Theoretic Properties

In Section 2 it was shown that if the linearized system Σ_L of Σ is controllable, observable, and stable, then Σ is also controllable observable and stable i.e there exist neighborhoods of the origin where these properties hold. The following discussions indicate how the additional nonlinear terms are determined in each case

Local Controllability: If the system Σ_L is controllable, then the system Σ is locally controllable i.e there exists a neighborhood Ω_c of the origin such that for any states $x_1, x_2 \in \Omega_c$ there is a finite sequence of inputs that transfer x_1 to x_2. For the linearized system we have the equation

$$x(n) = A^n x(0) + W_c U_{0,n} \tag{44}$$

where W_c is the controllability matrix and $U_{0,n} = [u(0), u(1), \cdots, u(n-1)]^T$. For the nonlinear system Σ, using the implicit function theorem it is shown in [36] that

$$U_{0,n} = W_c^{-1}[x(n) - A^n x(0)] + g(x(0), x(n)) \quad g(\cdot) \in \mathcal{H} \tag{45}$$

Local observability Similarly, we also have the result that if Σ_L is observable, Σ is locally observable. If the input-output sequences are defined as

$$\begin{aligned} Y_{(0,n-1)} &= [y(0), y(1), \cdots, y(n-1)]^T \\ U_{(0,n-2)} &= [u(0), u(1), \cdots, u(n-2)]^T \end{aligned} \tag{46}$$

Σ_L yields $x(0) = W_0^{-1}[Y_{(0,n-1)} - PU_{(0,n-2)}]$ while the application of the inverse function theorem yields

$$x(0) = W_o^{-1}[Y_{(0,n-1)} - PU_{(0,n-2)}] - \eta[Y_{(0,n-1)}, U_{(0,n-2)}] \tag{47}$$

where $\eta \in \mathcal{H}$, and P is a known matrix.

From the above it follows that there exists a neighborhood Ω_o of the origin in which the state $x(0)$ can be reconstructed by observing the finite sequence $Y_{(0,n-1)}$ and $U_{(0,n-2)}$.

Stability and Stabilizability:

It is well known that if the linear system described by

$$x(k+1) = Ax(k) \tag{48}$$

is asymptotically stable, then the nonlinear system

$$x(k+1) = Ax(k) + \eta(x(k)) \quad \eta(\cdot) \in \mathcal{H} \tag{49}$$

is (locally) asymptotically stable. This can be shown by demonstrating that a function $V(x) = x^T Px$ which is a Lyapunov function for (48), is also a Lyapunov function for (49) in a neighborhood of the origin.

It is also well known that if the system $x(k+1) = Ax(k) + bu(k)$ is controllable, then it can be stabilized by a feedback control law of the form $u(k) = \Gamma x(k)$ where Γ is a constant row vector. When the nonlinear system Σ is considered, we have the equation $x(k+1) = Ax(k) + bu(k) + f[x(k), \Gamma; g(x(k))]$ and it follows that this is also asymptotically stable in some neighborhood of the origin Ω_s. It has also been shown that there exists a nonlinear feedback controller $u(k) =$

$\Gamma x(k) + \gamma(x(k))$ which stabilizes the system in a finite number ($\leq n$) of steps. (i.e transfers any state $x_0 \in \Omega_s$ to the origin in a finite number ($\leq n$) of steps).

In summary, the results presented thus far merely make precise in the three specific contexts considered that design based on linearization works locally for nonlinear systems. In each case the existence of a nonlinear function in $\mathcal{H}$ assures controllability, observability, or stability. Neural networks can consequently be used to practically realize these nonlinear functions.

Set-Point Regulation and Tracking:

The same procedures used thus far can also be used to demonstrate that well defined solutions exist for the set-point regulation and tracking problems stated in Section 3. We merely provide the main ideas, and refer the reader to the source paper [37], as well as the principal references [19, 32] contained in it, for further details.

Theorem 1: Set-Point Regulation: The output y of a nonlinear system Σ (whose state vector is accessible) can be regulated around a constant value r in a neighborhood of the origin if this can be achieved for the linearized system Σ_L.

For Σ_L, an input $u = \Gamma x + v$, where v is a constant can be used to regulate the output around a constant value if the transfer function $c[zI - A]^{-1}b$ does not have a zero at $z = 1$. For the nonlinear system Σ, the same input yields a constant state x^* asymptotically, where

$$x^* = [A + b\Gamma]x^* + bv + f[x^*, \Gamma x^* + v] \tag{50}$$

since $r = cx^* + h(x^*)$, using the implicit function theorem it can be shown that v can be expressed explicitly in terms of r. This consists of the sum of the input used in the linear case (i.e $r/c[I - \bar{A}]^{-1}b,\ \bar{A} = A + b\Sigma$) together with $\gamma(r)$ where $\gamma \in \mathcal{H}$.

Tracking an Arbitrary Signal $y^*(k)$: To pose the problem of tracking an arbitrary signal $y^*(k)$ precisely, we need concepts such as relative degree, normal form, and zero dynamics of the nonlinear system, which are beyond the scope of this paper. We merely state the results when the state of Σ is accessible, and when it is not, and in the latter case use the NARMA representation for Σ, and the corresponding ARMA representation for its linearization Σ_L.

Theorem 2: (Tracking: State Vector Accessible) If the nonlinear system Σ has a well defined relative degree, and the zero dynamics of the linearized system Σ_L is asymptotically stable, then a neighborhood Ω of the origin, and a control law of the form

$$u(k) = (cA^{d-1}b)^{-1}[y^*(k+d) - \bar{P}x(k)] + g[x(k), y^*(k+d)] \tag{51}$$

exist where $g(\cdot) \in \mathcal{H}$, such that the output $y(k)$ of Σ follows asymptotically the desired output $y^*(k)$ provided $x(k)$, $x^*(k) \in \Omega$.

Theorem 3: (Tracking: Inputs and Outputs Accessible) Let Σ have a well defined relative degree and an input-output representation of the form:

$$\begin{aligned} y(k+d) \quad = \quad & \alpha_0 y(k) + \cdots + \alpha_{n-1} y(k-n+1) + \beta_0 u(k) + \cdots + \beta_{n-1} u(k-n+1) \\ & + \omega\left(y(k), y(k-1), \cdots, y(k-n+1), u(k), \cdots, u(k-n+1)\right) \end{aligned} \tag{52}$$

where $\beta_0 \neq 0$ and $\omega(\cdot) \in \mathcal{H}$.

If the zero dynamics of Σ is asymptotically stable, then there exists a control law of the form

$$
\begin{aligned}
u(k) \quad = \quad & \frac{1}{\beta_0}\Big[y^*(k+d) - \alpha_0 y(k) - \alpha_1 y(k-1) - \cdots - \alpha_{n-1} y(k-n+1) \\
& -\beta_1 u(k-1) - \cdots - \beta_{n-1} u(k-n+1)) \\
& +g(y(k), y(k-1), \cdots y(k-n+1), y^*(k+d), u(k), \cdots, u(k-n+1))\Big]
\end{aligned} \tag{53}
$$

such that $y(k)$ will follow any reference signal $y^*(k)$ with a sufficiently small amplitude, while all the signals in the system remain in a neighborhood Ω of the origin.

Comment 10: Before neural networks are used in any system, as stated in the past sections, the existence of the appropriate input-output maps must be established. Theorems 1-3 establish the existence of such maps.

4.2 Practical Design of Identifiers and Controllers (Linearization)

Having established the existence of the appropriate functions for the identification and control of nonlinear plants, we proceed to consider the application of these results to Problems 1-3 stated in Section 2, and the practical realization of neural networks. We shall use multilayer networks and radial basis function networks, as well as recurrent networks for both identification and control, depending upon the available prior information and the objectives.

Problem 1 (Identification): A stable nonlinear plant Σ is operating in a domain $D \subset X$, containing the origin. The objective is to determine a model $\hat{\Sigma}$ of the plant using neural networks. Since the input set is compact, the state x and the output y of Σ also belong to compact sets. Identification can be carried out on-line, or off-line using data collected as the system is in operation.

Two networks N_F and N_H are used to identify the functions F and H respectively. Since the state $x(k)$ is accessible, the network N_H can be trained by standard methods from a knowledge of $y(k)$ and the estimate $\hat{y}(k)(= N_H(x(k)))$.

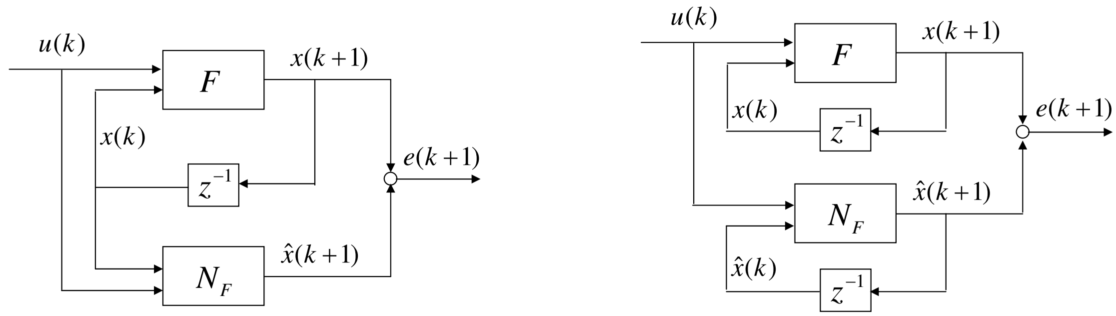
$N_F(k)$ can be identified using a series-parallel method as shown in Figure 11(a). Since the plant is known to be stable, and $x(k)$ is bounded, no stability questions arise in this case. If a parallel model is used to identify the system as shown in Figure 11(b), instability is possible. Hence, training of N_F should be carried out off-line using the weights obtained by the series-parallel method as initial values. Once the model is structurally stable, parameter adjustments can be carried out on-line, provided the adjustment is slow compared to the dynamics of Σ.

Comment 11: It may also be preferable to identify $F(x,u)$ as $Ax + Bu + f(x,u)$ to distinguish between the contributions of the linear and nonlinear parts.

Problem 2 (Control): Whether a multilayer network or a recurrent network is used to control the unknown plant, the stability question is invariably present. It is well known in industry that great caution is necessary while introducing nonlinear control, and that any increase in the nonlinear component of the control input has to be gradual.

Hence, as a first step, the region in the state space where $x(t)$ lies has to be limited and a linear controller designed first before the nonlinear component is added to it. Adjustment of network parameters has to be carried out using dynamic back-propagation.

Problem 3 (Plant Unstable): At present we have no methods available for controlling an unknown unstable nonlinear plant with arbitrary initial conditions. However, if the system is operating in the domain where linearization is valid, standard adaptive techniques can be used to stabilize the



a) Series-Parallel b) Parallel

Figure 11: Identification

system.

Increasing the domain of stability is a slow process and involves small changes in the nonlinear components of the input. The direction in which the parameter vector of the neural network is to be adjusted must be determined using one of the methods (of dynamic back propagation) described in Section 3.

If only the inputs and outputs of the plant are accessible in Problems 2 and 3, the NARMA model and the corresponding controller given by equations (52) and (53) have to be used. Figure 12 shows the structure of the indirect controller proposed in [12].

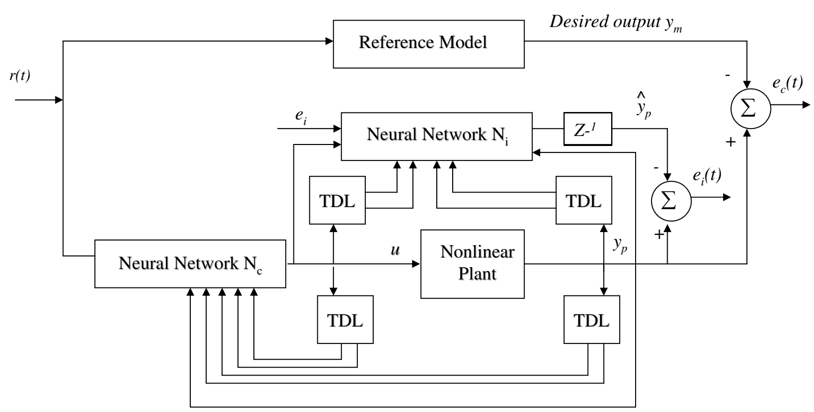


Figure 12: Indirect Control

<u>Comment 12:</u> The preceding sections indicate that the method of linearization is conservative but rigorous, assures stability using only the linearized equations, and improves performance using neural networks and a nonlinear component of the input that is small compared to the linear component. If the trajectories of the plant lie in a large domain where the above assumptions are not satisfied (i.e. $f(x, u)$ in equation (40) is comparable to the linear components) we do not have an adequate theory at the present time. This underscores the importance of Section 5, whose understanding is essential to formulate precisely adaptive control problems in regions where nonlinear effects are predominant.

Modeled Disturbances and Multiple Models for Rapidly Varying Parameters

In the three problems stated in Section 2 and discussed earlier there were no external disturbances. In addition, it was also assumed that the systems to be controlled are autonomous (or the plant parameters are constant) so that the plant could track a desired signal exactly as $t \to \infty$. However, external disturbances are invariably present, the plant generally contains dynamics not included in the identification model, and the characteristics of the plant may change with time. So the control has to be evaluated in the presence of external and internal perturbations. When small external perturbations and slow parameter variations affect the dynamics of the system, numerous methods such as the use of a dead zone, σ-modification, and $|\epsilon|$-modification have been suggested in the literature to assure robustness. These methods have also been suitably modified for use in neurocontrol. The reader is referred to the comprehensive volume on robust control by Ioannou and Sun [38] and monograph [39] by Tsakalis and Ioannou for details concerning such problems. From a theoretical standpoint, these methods can be rigorously justified if control based on linearization is used. Due to space limitations we do not consider them here. Instead, we discuss two cases: (a) where the disturbances are large and can be modeled as the outputs of unforced difference equations and (b) where the parameters vary rapidly and multiple models are used.

External Disturbances [40]

A SISO system Σ is described by the equations

$$\Sigma : \quad \begin{array}{rcl} x(k+1) & = & F[x(k), u(k), v(k)] \\ y(k) & = & h[x(k)] \end{array} \tag{54}$$

where $x(k)$ and $u(k)$ are the same as before and $v(k)$ is a disturbance which is the output of an unforced stable disturbance model Σ_v where

$$\Sigma_v : \quad \begin{array}{rcl} x_v(k+1) & = & g[x_v(k)] \\ v(k) & = & d[x_v(k)] \end{array} \tag{55}$$

where $x_v(k) \in \mathbb{R}^{\bar{n}}$. The state $x(k)$ of Σ as well as the disturbance are not accessible and it is desired to track a desired output $y^*(k)$ exactly (as $k \to \infty$) even in the presence of the disturbance. In classical adaptive control theory, when both the plant and the disturbance model are linear, it is known that exact tracking can be achieved by increasing the dimension of the controller. Mukhopadhyay and Narendra have used this concept for disturbance rejection in nonlinear dynamical systems. They have discussed the approach in detail in [40]. The following NARMA identification model is proposed by the authors.

$$\hat{y}(k+1) = N[y(k), y(k-1), \cdots y(k-(n+m)+1), u(k), u(k-1), \cdots, u(k-(n+m)+1)] \tag{56}$$

to identify the given system as an $(n+m)^{th}$ order system and control it as in Problem 3. It can be shown that this results in $y(k)$ (and hence $\hat{y}(k)$) tracking $y^*(k)$ exactly asymptotically in time. A large number of simulation studies have been successfully carried out on very different classes of nonlinear systems and a few of these are reported in [40].

Multiple Models

The authors of this paper are currently actively working in this area and without going into much detail we state the problem qualitatively in its simplest form and merely comment on the principal concepts involved.

A plant Σ can operate in any one of N "environments" at any instant. The NARMA models Σ_1, $\Sigma_2, \cdots$, Σ_N approximate the behavior of Σ in the different environments. In the simplest case the N models are assumed to be known so that N controllers (each corresponding to one of the models) can be designed. The plant Σ switches slowly and randomly between the N environments. The objective is to detect the change in the plant based on the available output data and use the appropriate controller corresponding to the existing model.

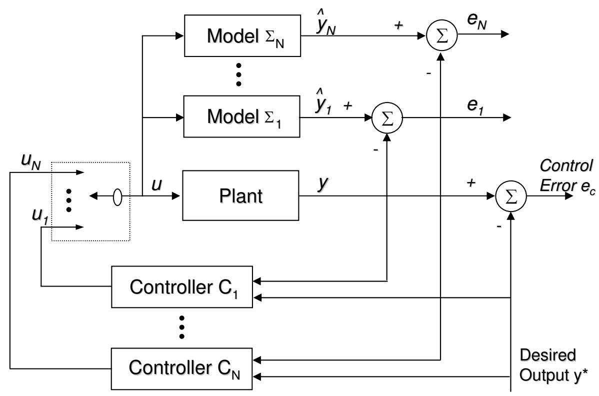


Figure 13: Multiple Models

As shown in Figure 13, at every instant, N errors $e_i(k)$ ($e_i(k) = y_i(k) - y(k)$, where $y_i(k)$ is the output of the i^{th} model) are computed, the model that corresponds to the smallest error according to an error criterion is chosen at that instant, and the controller corresponding to it is used. If the plant comes to rest after a finite number of switchings, convergence of the model to the plant has been demonstrated in the linear case [41]. It has also been shown that the same is true for nonlinear systems, provided all of them satisfy the linearization conditions stated earlier. Simulation studies have been very successful and the transient performance of the system is substantially improved.

When the plant characteristics vary, they may not correspond exactly to one of the predetermined models assumed in the previous case. In such situations tuning of both the model Σ_i and the corresponding controller C_i may be needed on-line. This has been referred to as the "switching and tuning" approach [42] and is widely used in many different fields (refer to Section 7). The results obtained for deterministic systems have also been extended to stochastic systems [43].

In the cases discussed thus far, all the systems Σ_i share the same equilibrium state, and all the trajectories lie in a neighborhood of the origin. In substantially more interesting problems, the models have different equilibrium states and operate in neighborhoods corresponding to those equilibrium states. However, transitions between equilibrium states may involve regions where the nonlinear terms are dominant. Switching to assure that the system transitions from one neighborhood to another raises questions which require concepts of nonlinear control that are contained in Section 5.

Control of Nonlinear Multivariable Systems [44]

Most practical systems have multiple inputs and multiple outputs, and the applicability of neural networks as practical adaptive controllers will be judged by their performance in such contexts.

The representation, identification and control of nonlinear multivariable systems are rendered very complex, as described in Section 2, due to the couplings as well as the delays that exist between the inputs and outputs. We comment briefly here on two questions which are relevant to the objectives of this paper. They are (a) tracking problem in multivariable control and (b) decoupling of multivariable systems. For details concerning these problems the reader is referred to [44].

Tracking: Given a controllable and observable nonlinear dynamical system with m inputs $u_i(\cdot)$ and m outputs $y_i(\cdot)\quad i = 1, 2, \cdots, m$ with well defined relative degrees d_i, it can be shown that it can be represented in a neighborhood Ω of the origin by the equations

$$y_i(k + d_i) = \Phi_i[x(k), u(k)] \quad i = 1, 2, \cdots, m \tag{57}$$

Since $x(k)$ (by the assumptions made) can be expressed in terms of the outputs and inputs $y(k), y(k-1), \cdots, y(k-n+1)$ and $u(k), \cdots, u(k-n+1)$, the system has a NARMA representation of the form (23). It is these equations that are used to determine controllers for tracking a desired output $y^*(k)$. Based on the existence of control inputs for LTI systems, it can be shown that the desired input $u(k)$ can be generated as the output of a multivariable nonlinear system

$$u(k) = \Phi_c[y(k), \cdots, y(k-n+1), r(k), u(k-1), \cdots, u(k-n+1)] \tag{58}$$

As stated earlier in this section, $\Phi_c(\cdot)$ can be approximated using a neural network N_c (or as the sum of a linear function of $u(k-i)$ and $y(k-i)$ and a nonlinear component which belongs to $\mathcal{H}$). Stability is guaranteed by the linear part while exact tracking is achieved using the nonlinear component.

Decoupling: An important practical question that arises in multivariable control is decoupling. Qualitatively, in its simplest form it implies that each output is controlled by one input. In mathematical terms the output y_i is invariant under inputs $u_j, j \neq i$, and is not invariant with respect to u_i. The desirability of decoupling in some practical applications is obvious. For linear multivariable systems described by the equation $x(k+1) = Ax(k) + Bu(k),\ y(k) = Cx(k)$, it is well known that the system can be decoupled if a matrix $E \in R^{m\times m}$ is nonsingular [45] and

$$\begin{aligned} u(k) &= E^{-1}\begin{bmatrix} c_1 A^{d_1} \\ c_2 A^{d_2} \\ \vdots \\ c_m A^{d_m} \end{bmatrix} x(k) + E^{-1} r(k) \\ &= Gx(k) + Fr(k) \end{aligned} \tag{59}$$

(E is the matrix whose i^{th} row is $c_i A^{d_i - 1} B$). If the linearization of the given nonlinear system can be decoupled, it can be shown using the same arguments as before that in a neighborhood of the origin, exact decoupling is possible using a controller of the form

$$u(k) = Gx(k) + Fr(k) + g(x(k), r(k)) \quad g \in \mathcal{H} \tag{60}$$

While approximate decoupling can be achieved using linear state feedback, exact decoupling can be achieved by approximating $g(\cdot)$ using neural networks. Similar arguments can also be given for the case when decoupling has to be achieved using only inputs and outputs.

Interconnected systems

All the problems considered thus far in this section are related to the identification and control of isolated nonlinear systems. As stated earlier, interest in the control field is shifting to problems that arise when two or more systems are interconnected to interact in some sense. In adaptive contexts, to make the problems analytically tractable, the different dynamical systems are assumed to be linear. Since the solution of the linear problem is the first approximation for nonlinear control problems that are addressed using linearization methods, we present here an important result that was obtained recently and that may have interesting implications for decentralized nonlinear control of the type stated in Section 2.

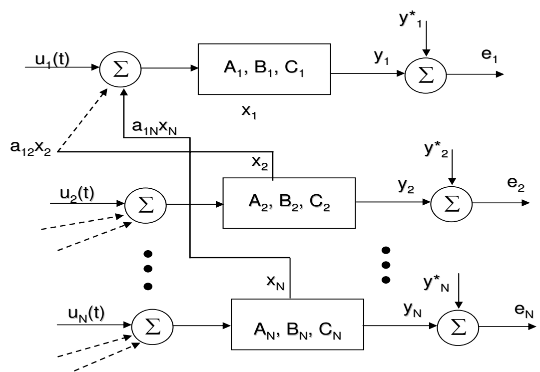


Figure 14: Interconnected System

The Problem: A system Σ shown in Figure 14 consists of N subsystems Σ_1, $\Sigma_2, \cdots$, Σ_N. Σ_i is linear and time-invariant, whose objective is to choose a control input $u_i(\cdot)$ such that its state $x_i(t)$ tracks a desired state $x_{mi}(t)$. Each system Σ_j affects the input to the system Σ_i by a signal $a_{ij}x_j$ where Σ_i has no knowledge of either a_{ij} or $x_j(t)$. The question that is raised is whether all the subsystem Σ_i can follow their reference inputs without having knowledge of the state vectors $x_j(t)$ of the other systems.

The above problem was answered affirmatively in [46, 47]. If it can be assumed that the desired outputs $x_{mi}(t)$ of the N subsystems is common knowledge, each subsystem attempts to cancel the perturbing signals from the other subsystems (e.g $h_{ij}x_j$) by using $\hat{h}_{ij}x_{mj}$ in place of $\hat{h}_{ij}x_j$, and adapting $\hat{h}_{ij}(t)$. It was shown that the overall system would be asymptotically stable and that all the errors would tend to zero.

Decentralized nonlinear control with nonlinear interactive terms between subsystems stated in Section 2 is obviously the next class of problems of interest and can be attempted using neural networks, provided that in the region of interest Ω in the state space, the dynamical systems satisfy the condition discussed earlier. For more general nonlinear interconnections of the type stated in Section 2, where the trajectories lie in larger domains in the state space, more advanced methods will be needed (refer Section 5).

4.3 Related Current Research

The literature on neural network based control is truly vast, and there is also intense activity in the area at the present time. Dealing with all the on-going work in any detail is beyond the scope of this paper. Since our main objective is to examine and present the basic principles used in simple terms, we present in this subsection some representative samples of methods, introduced by well known researchers.

a.) A general method for representing nonlinear systems for control purposes was introduced by Suykens and his co-workers in [48]. The authors claim that the NL_q system form that they introduce (alternating sequences of nonlinear elements (N), linear gains (L) having q layers) represents a large class of dynamical systems that can be used as identifiers and controllers. The plant and controller models (represented by M_i and C_i) have the general form

$$x(k+1) = \Gamma_1[A_1\Gamma_2[A_2 \cdots \Gamma_q[A_q x(k)(k)] \cdots + B_2 u(k)] + B_1 u(k)] \quad (61)$$

$$y(k) = \Lambda_1[c_1\Lambda_2[c_2 \cdots \Lambda_q[c_q x(k) + D_q u(k)] \cdots D_2 u(k)] + D_1 u(k)] \quad (62)$$

where $x(k) \in \mathbb{R}^n$ is the state, $y(k) \in \mathbb{R}^m$ is the output and $u(k) \in \mathbb{R}^r$ is the input of the recurrent network where the complexity increases with q. The same structure is used for both controllers and identifiers so that stability questions of the subsystems as well as the overall system are the same. The procedure used to control the plant is based on indirect adaptive control.

Taking advantage of the structure of the models and using Lure theory it is shown that sufficient conditions can be derived for asymptotic stability. These are expressed in terms of the matrices $(A_i(i = 1, 2 \cdots, q))$. The model is asymptotically stable if there exist diagonal matrices D_i such that

$$||D_i A_i D_{i+1}^{-1}|| \leq 1, \quad \forall\ i \quad (63)$$

The above condition assures the existence of a Lyapunov function of the form $V(x) = ||D_1 x||$.

To approximate the plant dynamics the parameters of the matrices are adjusted using dynamic back-propagation. To assure stability (satisfying inequalities) it is shown that the adjustment of A_i can be realized by solving a nonlinear constrained optimization problem, which results in a modified back-propagation scheme.

b.) A general structure is also introduced by Poznyak and his co-workers [49] in which identification, estimation, sliding mode control, and tracking are treated. The emphasis of the book is on recurrent networks and we briefly outline below the approach proposed for the identification problem. It is assumed that a general nonlinear system can be represented by the difference equation

$$x(k+1) = Ax(k) + W_1\sigma(x(k)) + W_2\phi(x(k))\gamma(u(k)) \quad (64)$$

where W_1 and W_2 are weight matrices and $\sigma(\cdot)$, $\phi(\cdot)$, and $\gamma(\cdot)$ are known nonlinear functions. Both σ and ϕ contain additional parameter vectors V_1 and V_2 corresponding to hidden layers. However, σ, ϕ, and γ are assumed to be bounded functions. Hence W_1 and W_2 determine a family of maps to which any given nonlinear system belongs. The objective is consequently to estimate W_1 and W_2 and then to determine the control input $u(\cdot)$.

Identifiers using the approach in [49] are represented by recurrent networks described by

$$\hat{x}(k+1) = \hat{W}_1(k)\sigma(\hat{x}(k)) + \hat{W}_2(k)\phi(\hat{x}(k))\gamma(u(k)) \quad (65)$$

and the adaptive laws for adjusting $\hat{W}_1(k)$ and $\hat{W}_2(k)$ are derived. For series parallel models (where the arguments of σ and ϕ are $x(k)$ rather than $\hat{x}(k)$) standard adaptive laws can be derived directly from those found in [15]. Similar laws are also derived for recurrent networks. In classical adaptive control, deriving adaptive laws for such models was known to be a very difficult problem and was never resolved for the deterministic case. However, by making several assumptions concerning the boundedness of various signals in the system, the authors demonstrate that $V(e, \tilde{W}_1, \tilde{W}_2, \tilde{V}_1, \tilde{V}_2) = e^T Pe + \frac{1}{2} Tr[\tilde{W}_1^T \tilde{W}_1 + \tilde{W}_2^T \tilde{W}_2 + \tilde{V}_1^T \tilde{V}_1 + \tilde{V}_2^T \tilde{V}_2]$ is a Lyapunov function, which assures the convergence of the state error e to zero.

Assumptions:

As has been stated several times in the preceding sections, assumptions concerning the plant to be controlled are invariably made to have analytically tractable problems. In fact, in the next section, prior information that is assumed in linear adaptive control is indicated. Since nonlinear adaptive control is very complex, it was only natural that different assumptions were made, starting around the early 1990s. Gradually, these became accepted in the field and researchers began to apply them to more complex systems. At the present time they have become an integral part of the thinking in the field.

In Section 1, it was stated that assumptions can make complex problems almost trivial. The authors believe that the neurocontrol community should re-examine the conditions that are currently assumed as almost self-evident. In Section 4.4 the authors provide their own view on this subject. In the rest of this section the evolution of these assumptions to their present form is traced briefly by examining a sequence of typical papers that have appeared in the literature.

c.) Chen and Liu [50] (1994): The problem of tracking in a nonlinear system is considered. The system is described by the equation

$$\begin{aligned} \dot{x} &= f_0(x) + g_0(x)u \\ y &= h(x) \end{aligned} \tag{66}$$

where $x(t) \in \mathbb{R}^n$ can be measured and f_0, g_0 and h are smooth. The input and output are related by the equation $\dot{y} = f_1(x) + g_1(x)u$ where $f_1(x) = h_x^T f_0(x)$ and $g_1(x) = h_x^T g_0(x)$. Neural networks approximate f_1 and g_1 as $\hat{f}_1$ and $\hat{g}_1$ and the latter are used to determine the control input u.

Comment 13: The system is stable and gradient methods are used to approximate unknown functions.

d.) Polycarpou [51] (1996): This deals with a second order system

$$\begin{aligned} \dot{x}_1 &= f(x_1) + \phi(x_1) + x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{67}$$

It is assumed that $f(\cdot)$ is known while $\phi(\cdot)$ is unknown and that the state variables are accessible. The objective is to regulate the system around the equilibrium state $x_1 = x_2 = 0$. To make the problem tractable it is assumed that $\phi(x) = \theta^T \zeta(x_1)$, where $\zeta(x_1)$ is a vector of known basis functions. With this assumption the problem becomes a nonlinear (second order) version of the problems described in Section 2.3. Using similar methods, an adaptive law for obtaining an estimate $\hat{\theta}$ of θ, and a control law u are derived and are shown below. If $z_1 = x_1$, $z_2 = x_2 - \alpha(x_1, \hat{\theta})$ and $\alpha(x_1, \hat{\theta}) = -x_1 - \hat{\theta}^T \zeta(x_1)$.

$$\begin{aligned} u &= -z_1 - z_2 + \frac{\partial\alpha}{\partial x_1}(x_2 + f + \theta^T\zeta) + \frac{\partial\alpha}{\partial\theta}\dot{\theta} \\ \dot{\hat{\theta}} &= \Gamma\left[\zeta\left(z_1 - z_2\frac{\partial\alpha}{\partial x_1}\right) - \sigma(\hat{\theta} - \theta_0)\right] \end{aligned} \tag{68}$$

In spite of the assumption and the low order of the system, the resulting control is found to be quite complex which is typical of backstepping. The control used is seen to depend upon the partial derivatives of α which is estimated on-line.

Comment 14: As a mathematical problem the above is precisely stated. The assumption that ϕ can be approximated using basis functions, in our opinion, is justifiable in this case, since it is a function of only one variable. We discuss this further in the next section.

e.) Rovithakis [52] (1999): A system is described by the equation

$$\dot{x} = f(x) + g(x)u + \omega(x,u) \quad x(t) \in \mathbb{R}^n \tag{69}$$

where $\omega(x,u)$ is an external disturbance that is bounded and unknown. The objective is to regulate the system close to the equilibrium state. The following two assumptions are made:

(i) in the absence of the disturbance a control $\alpha(x)$ stabilizes the system, and a Lyapunov function for the nonlinear system is $V(x)$.

(ii) $\omega(x,u(x))$ lies in the range of the basis functions $S(x)$, i.e $\omega(x,u(x)) = W^TS(x)$ for any control law.

Based on these assumptions it is shown that a control input $u(x) = \alpha(x) + u_c(x)$ can be determined to stabilize the system. In 2004, an expanded version of this paper was presented in [53].

Comment 15: It is assumed that the nominal system (without the disturbance) is stable and that an explicit Lyapunov function $V(x)$ is known for it. Also while $\phi(x_1)$ in (d) was a function of a single variable, an arbitrary function $\omega(x,u(x))$ is approximated here using basis functions.

f.) Out of a very large collection of papers published in the literature [54]-[60] which are based on backstepping procedures and which utilize basis functions, we consider a representative sample of four papers here.

(i) In [54] Kwan and Lewis (2000) consider the tracking problem in a system described by

$$\begin{aligned} \dot{x}_i &= F_i(x_1, x_2, \cdots, x_i) + G_i(x_1, x_2, \cdots, x_i)x_{i+1} \quad i = 1, 2, \cdots, i \\ \dot{x}_n &= F_n(x_1, x_2, \cdots, x_n) + G_n(x_1, x_2, \cdots, x_n)u \end{aligned} \tag{70}$$

where the state variables are accessible, G_i's are known and sign definite and F_i's are unknown. The objective is to determine a control input such that $x_1(t)$ tracks a desired output asymptotically.

(ii) Ge and Wang (2002) consider a similar problem in [58] in which the system is described by the equations

$$\begin{aligned} \dot{x}_i &= f_i(x_1, x_2, \cdots, x_i) + g_i(x_1, x_2, \cdots, x_i)x_{i+1} \quad i = 1, 2, \cdots, i \\ \dot{x}_n &= f_n(x_1, x_2, \cdots, x_n) + g_n(x_1, x_2, \cdots, x_n)u \end{aligned} \tag{71}$$

As in (i) it is assumed that g_i are bounded away from 0. It is further assumed that $|\dot{g}_i| \leq g_{id}$ $i = 1, 2, \cdots, n$ and that a complex unknown function of $z_1, z_2, \cdots, z_n$ related to the unknown function f_i and g_i lies in the span of a set of known basis functions.

(iii) Li, Qiang, Zhuang, and Kaynak (2004) [59] consider the same system as in (ii) and attempt the same problem with the same assumptions an g_i and $\dot{g}_i$ but use two different sets of basis functions.

(iv) Wang and Huang (2005) [60] consider the system described by

$$\begin{aligned} \dot{x}_i &= f_i(x_1, x_2, \cdots, x_i) + x_{i+1} \quad i = 1, 2, \cdots, i \\ \dot{x}_n &= f_n(x_1, x_2, \cdots, x_n) + u. \end{aligned} \tag{72}$$

Tracking of a desired signal is achieved by computing a control input assuming that each element $f_i(x_1, \cdots, x_i)$ in equation (72) can be expressed as $\theta_i^T \xi(x_1, x_2, \cdots, x_i)$ where ξ_i are basis vectors.

The above typical papers clearly indicate the thrust of the research in the community and the emphasis on basis functions.

4.4 Theoretical and Practical Stability Issues

In Section 4.3 we discussed some typical methods that are representative of a large number of others that have also been proposed and share common features with them. Most of them claim that their approaches result in global asymptotic stability of the overall system, and robustness under perturbations with suitable modifications of the adaptive laws. Since all of them, in one way or another, attempt to emulate classical adaptive control, we shall start with the latter to provide a benchmark for examining and comparing the different methods. Since the approach based on linearization described in Sections 4.1 and 4.2 is closest to classical adaptive control, we shall briefly comment on that. Following this, we shall raise several questions and provide brief comments regarding each one of them to help us to critically evaluate the contributions made by the different authors.

Linear Adaptive Control: The theoretical study of linear adaptive control starts with the assumptions made concerning the plant to be controlled. The plant is assumed to be linear and time-invariant (LTI) of order n, with $2n$ unknown parameters. An upper bound on n is known. All the zeros of the plant transfer function are assumed to lie in the open left half of the complex plane (minimum phase). If the plant is to be identified (i.e. parameters are to estimated) it is normal to assume that it is stable with bounded inputs and bounded outputs. If the plant is to be controlled, it is generally assumed to be unstable and whatever adaptive scheme is proposed is expected to stabilize it. Since the controller parameters that are adjusted become state variables, the overall system is nonlinear with an extended state space. It is in this extended state space that all properties of the overall system are studied.

By a suitable choice of a Lyapunov function V and corresponding adaptive laws, it is first shown that the system is stable and that all signals and parameters are bounded. Asymptotic stability never follows directly since the time derivative $\dot{V}$ of V is always negative semi-definite and not negative definite. It is next shown that the control error tends to zero. Additional conditions on the reference input (persistent excitation) are needed to show that the parameter errors tend to zero (asymptotic stability). Theoretically, this is what is meant by the proof of stability in adaptive control. Another important consequence of the linearity of the plant is that all the results are global.

Nonlinear Adaptive Control Using Linearization Methods: As shown in Section 4.1, in the method based on linearization, we are operating in a domain where the linear terms dominate the nonlinear terms. Hence all the results of linear adaptive control carry over, but are valid only in this domain. If only linear adaptive control is used, the errors do not tend to zero due to the presence of the nonlinear terms. It is at this stage that neural networks are needed to compensate for the nonlinear terms and make the control error tend to zero. Also, since the overall nonlinear effect (due to plant and controller) is small compared to the linear terms, adaptation of both linear and nonlinear terms can be fast. However, to assure that the approximation is sufficiently accurate, the neural networks are invariably adjusted on a slower time scale. This is the same procedure adopted for multivariable control, control using multiple models, and interconnected systems.

Nonlinear Adaptive Control: The following simple adaptive control problem is a convenient vehicle for discussing many of the questions related to purely nonlinear adaptive control.

The Problem: A system Σ is described by the differential equation

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\cdots \\ \dot{x}_n &= f(x_1, x_2, \cdots, x_n) + u \end{aligned} \tag{73}$$

$f(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is smooth, the state variables x_i are accessible and the input u is to be chosen so that the output x_1 tracks a desired output y_{1d} which is the output of a stable n^{th} order differential equation

$$y_d^{(h)} + \sum_{i=0}^{n-1} \alpha_{i+1} y_d^{(i)} = r \tag{74}$$

where r is a known bounded reference input.

$f(\cdot)$ known: In the deterministic case where f is known, the choice of $u(\cdot)$ is simple. If

$$u = -\sum_{i=1}^{n} a_i x_i - f(x) + r \tag{75}$$

the error $e = (y - y_d)$ satisfies the same stable homogeneous differential equation and tends to zero asymptotically.

$f(\cdot)$ unknown: This is an adaptive control problem, and strictly speaking one for which a solution does not exist unless some assumptions are made concerning $f(\cdot)$. The prior information concerning $f(\cdot)$ determines the nature and complexity of the adaptive control problem. Consequently, it is with these assumptions that we are concerned here.

- If it is assumed that $f(x) = \alpha^T \eta(x)$ where $\eta(x)$ is a known vector function of x and $\alpha \in \mathbb{R}^n$ an unknown parameter vector, the problem is trivial from a theoretical standpoint. The theoretical solution is obtained by using the control input

$$u = -\sum_{i=1}^{n} a_i(t) x_i - \hat{\alpha}^T \eta(x) + r \tag{76}$$

  and adjusting $\hat{\alpha}$ using an adaptive law derived from the error model in Figure 4. This solution was known in the adaptive control field thirty years ago.

- Suggesting that $f(x)$, $x \in \mathbb{R}^n$, can be approximated by $\alpha^T \eta(x)$ by choosing $\eta_i(x)$ as basis functions and N sufficiently large is impractical even for small values of n (as stated in (d), $n = 1$,may be an exception). Instability may be caused by the residual error between $f(x)$ and $\hat{\alpha}^T \eta(x)$.
- If $f(\cdot)$ is a part of a nonlinear dynamical system to be controlled, information concerning the function $f(\cdot)$ in the strict adaptive control problem can be obtained only from the inputs and outputs of Σ.
- To measure the inputs and the outputs of Σ to estimate $f(\cdot)$, it must be assumed that the system is stable. Obviously, this assumption is not valid since stabilizing the system is one of the main objectives of adaptive control.

If $f(\cdot)$ is unknown the problem has *not* been solved rigorously thus far. However, a large number of papers dealing with very complex nonlinear adaptive control problems with multiple unknown nonlinearities and high dimensional state space have appeared in the literature. In all cases the nonlinearities are approximated as $\alpha^T \eta(x)$ where $\eta(x)$ is known.

In the following paragraph we abstract from the above sample problem a set of general questions that we feel have to be answered before a nonlinear adaptive control problem is addressed.

Questions Related to Nonlinear Adaptive Control:

(i) Is the representation of the plant sufficiently general?

It is clear that assumptions have to be made to render the problem tractable. One extreme assumption would be to assume that the plant is linear! But this will limit the class of plants to which the methods can be applied. In the twentieth century, starting with the work of Volterra, numerous representations for nonlinear systems have been proposed. If truncated models are used to identify a nonlinear system, the magnitudes of the residuals are evident in such cases. With the models proposed in [48] and [49] this is not the case. For example, the effect of making the model more complex is not clear.

(ii) Are basis functions a generally acceptable way to approximate a nonlinear function?

While it is true that any continuous function $f(x)$, $x \in \mathbb{R}^n$ can be approximated as $f(x) \approx \alpha^T \eta(x)$ $\alpha \in \mathbb{R}^N$ using a sufficiently large number N of basis function $(\eta_i(x))$, it is not clear how the approximation error scales with the dimension n. The authors have considerable experience with approximation methods and have carried out extensive numerical identification for many years using different approaches including neural networks. These have shown that N must be very large even for simple functions. The number of basis functions increases dramatically with the dimension of the space over which the function is defined. Hence, from both theoretical and practical standpoints $f(x) = \alpha^T \eta(x)$ is not a satisfactory parametrization of the approximator (though it may be convenient to derive adaptive laws). The many theoretical proofs given in the literature are consequently not acceptable in their present forms.

(iii) Is the plant stable or unstable?

As stated earlier, following adaptive control, we will assume that the plant is stable only if identification is of interest, and unstable if the principal objective is stability. In our opinion, at present, only the method based on linearization can be used to stabilize an unknown unstable system.

(iv) If the plant is unstable, is it a continually evolving process (such as an aircraft or a chemical process) or can it be stopped or interrupted and re-initiated (like a broom balancer, or a biped robot (as in Section 7))?

These two represent very different classes of problems. On-line adaptive control refers only to the first class. Repetitive learning, as in the second case, is not on-line adaptive control but is important both practically and theoretically, as seen from the next question.

(v) If the plant is stable, is the region S in the state space in which the trajectories lie known?

We shall assume that this is indeed the case (though it is not a simple assumption in higher dimensions). If this is all the prior information available, a neural network can be trained (slowly off-line or on-line), to identify the system in S. However, the behavior of the system outside S being unknown, any input which drives the system outside S can result in instability. The reference trajectory should therefore lie inside S and during the adaptation process, the plant trajectories should also lie in S. This accounts for the great care taken in industry while trying new control methods.

If a process can be interrupted, regions outside S can be explored and a feedback controller can be designed through repetitive learning. In the case of on-line adaptive control regions outside S can be explored incrementally using continuity arguments and approximating properties of neural networks. To the authors' knowledge such investigations have not been carried out thus far.

(vi) Are the controllers to be designed off-line or on-line?

As seen from earlier comments, stability questions arise only in the latter. Stability questions also arise in computer simulations but very little cost is attached to them, and they can be re-initiated.

(vii) Are gradient methods or stability based methods used in the adjustment of the controller?

The essential difference between the two is in time-scales. The latter operate in real time (i.e the dynamics of adaptation is as fast as the dynamics of the plant). No gradient method in real time has been demonstrated to be stable. Therefore, such method operating in a slow time scale cannot stabilize an unstable plant. However, almost all of them can be shown to work satisfactorily if the plant is stable and the adjustments are sufficiently slow. This accounts for neural networks performing very well in practical applications. Once again this demonstrates that successful applications do not necessarily imply sound theory.

In the authors' opinion, there have been very few real theoretical results in nonlinear adaptive control using neural networks. The solutions for the most part are not mathematically precise. Obviously better theoretical formulations of problems are needed. Much greater emphasis has to be placed in the future on the prior information assumed about $f(x)$, and the choice of the basis functions dictated by it. This has not impeded the use of neural networks in practice to improve the performance of systems which have already been stabilized using linear controllers. The applications described in Section 7 attest to this.

5 Global Control Design

A basic problem in control is the stabilization of an equilibrium state. In Section 4, nonlinear stabilizers were developed which are valid in the neighborhood of such a point. An obvious question of both practical and theoretical importance is whether the region of validity can be extended to larger domains of the state space.

The study of the global properties of nonlinear systems and their control is much more complicated than the local approaches employed so far. Nevertheless, the development of adequate mathematical tools has always been guided by linear intuition and aimed at finding analogies of the concepts developed in linear systems theory. As Brockett put it, even as the state space of a system becomes a differentiable manifold, the characterization of observability, controllability and realization "is not more difficult than linear systems of the usual type in $\mathbb{R}^n$ "[62].

The mathematical machinery used in the study of global system theory consists of differential geometric methods, the theory of foliations, and the theory of topological groups. Our objective in this section is to point the reader to the excellent and insightful body of literature that exists on the subject as well as to convey the intuition behind the principal ideas involved. This will permit the neural network community to formulate well–posed problems in the design of nonlinear controllers using neural networks as well as to chart future directions for research.

We begin by characterizing the natural state space of a nonlinear dynamic system. The first question that arises naturally is the essential difference between linear and nonlinear systems, from a geometric viewpoint. The state space $\mathbb{R}^n$ of a linear system is "flat" in the sense that it expands to infinity along the direction given by the vectors of a basis of $\mathbb{R}^n$. The space of nonlinear systems, on the other hand, is *curved* and is defined as the manifold M, where a point $p \in M$ if there exists an open neighborhood U of p and a homeomorphic map $\varphi : U \rightarrow \varphi(U) \subset \mathbb{R}^n$, called the (local) coordinate chart of M. In other words, the manifold "looks" locally like $\mathbb{R}^n$. This simple fact has an important consequence: Many coordinate systems may be needed to describe the global evolution of a nonlinear dynamic system. While the manifold is an abstract geometric object, the coordinate system is the physical handle on that object through which we have to interact with the system when we control it. It is important to keep this in mind when designing global nonlinear controllers.

5.1 Dynamics on Manifolds

An excellent textbook on the subject is Boothby 1975 [63]. One important idea is the following: The flow of a system is a C^1-map $\phi :\ \mathbb{R} \times U \rightarrow M$ sending an initial value $p \in U$ defined on some *open* neighborhood $U \subset M$ to a value $\phi(t,p) \in M$ (at time $t \in [t_0, t_1]$). It defines a vector–field X_p as follows:

$$X_p = \frac{d}{dt}\,\phi(t,p)\bigg|_{t=t_0} \tag{77}$$

X_p is the tangent vector to the curve $t \rightarrow \phi(t,p) \in M$ at $t = 0$. Given a coordinate chart (U, φ), we obtain the usual differential equation

$$f(x) = \dot{x} \tag{78}$$

Notice that $f(x)$ is merely the local representative of the vector field X_p defined in (77) on the smooth manifold M. Furthermore $\varphi(p) = x$ for all $p \in U$. Once the solution leaves the neighborhood U in which the representation is valid, we have to find a new set of local coordinates.

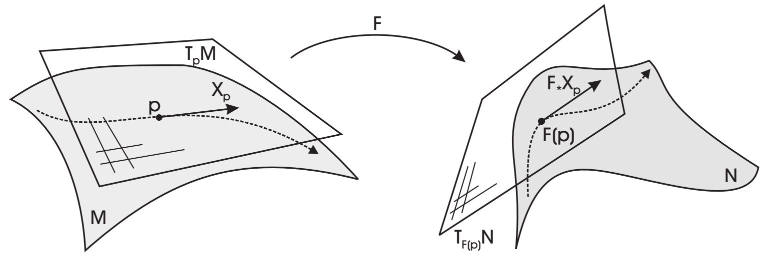


Figure 15: Tangent maps

Denote by $C^\infty(x)$ the set of smooth functions defined on a neighborhood of $x \in \mathbb{R}^n$. Any vector field defines a *linear operator* assigning to any $h \in C^\infty(x)$

$$(L_f h)(x) = \sum_{i=1}^{n} f_i(x) \left.\frac{\partial h}{\partial x_i}\right|_x \tag{79}$$

which is called the directional (Lie-) derivative of h along f. Geometrically, the vector field assigns to each $p \in M$ a tangent vector given as an element of a linear space of *mappings* called *tangent space* T_pM to the manifold M at p. The mappings are denoted by

$$X_p : \quad C^\infty(p) \to \mathbb{R} \tag{80}$$

Given a smooth map $F : \ M \to N$ it is clear that for any point $p \in M$ we have $F(p) \in N$. But what happens to the tangent vectors attached to p? We define a map (called the tangent map of F at p) $F_* : \ T_pM \to T_{F(p)}N$ as follows:

$$F_* X_p(h) = X_p(h \circ F) \tag{81}$$

where h is (again) a smooth scalar valued function $h : M \to \mathbb{R}$. $h \circ F$ denotes the composition of h and F (at p). Hence the argument of X_p in equation (81) is simply the function $h(\cdot)$ evaluated at the point $F(p)$. It is easily checked that F_*X_p is indeed an element of $T_{F(p)}N$, i.e. the tangent space to N at $F(p)$ see Figure 15.

The tangent map $\varphi_* : \ T_pM \to T_{\varphi(p)}\mathbb{R}^n$ is used to define local representatives of the tangent vectors X_p at $p \in M$. The tangent bundle defined as

$$TM = \bigcup_{p \in M} T_pM \tag{82}$$

is $2n$-dimensional with natural coordinates $(\varphi(p), \varphi_*(X_p)) =: (x, f(x))$, i.e. it is composed of the point p on M and the corresponding tangent vector $X_p \in T_pM$ attached at that point.

We are now ready to define a (smooth) vector field as the mapping

$$X : \ M \to TM \tag{83}$$

assigning to every $p \in M$ a tangent vector $X_p \in T_pM$ (in a smooth way). In view of the above a nonlinear control system of the general type can be defined in local coordinates as follows

$$\Sigma : \quad \dot{x} = f(x, u) \tag{84}$$

where $\varphi(p) = x \in \mathbb{R}^n$ and $\varphi_*(X_p(u)) = f(x,u) \in T_x\mathbb{R}^n$ both defined in a neighborhood U of M, a smooth connected manifold. Notice that the tangent space $T_x\mathbb{R}^n$ to the Euclidean space at any point x is actually equivalent to $\mathbb{R}^n$. The vectorfield is parameterized by the controls $u \in V \subset \mathbb{R}^r$. We assume that the solution for (84) exists up to infinite time for any fixed u.

5.2 Global controllability and stabilization

Keeping in mind that the representation of a dynamical system on curved spaces requires many local coordinate systems, we set $M = \mathbb{R}^n$ in this section. We are interested in designing a globally stabilizing feedback controller for a general nonlinear system of the form (84).

Among the many ideas developed for nonlinear feedback control (see e.g. [64] and references therein) we select one that closely builds upon the results obtained in chapter 4, in fact it will allow us to extend the results in the very direct meaning of the word, see [65].

Given the control system (84) where $x \in \mathbb{R}^n$. The solution for (84) exists up to infinite time for any $u \in V \subset \mathbb{R}^r$ fixed. The problem is to find a feedback control $u \in V$ such that $[x^*,\, u^* = u(x^*)]$ is an asymptotically stable fixed point of the closed–loop system $f(x, u(x))$. We restrict ourselves to the case of semi–global stabilization, i.e. the region of attraction of x^* is a compact subset $K \subset \mathbb{R}^n$. A first question is whether a *smooth* feedback can be found which stabilizes the system. This is fundamental if neural networks are to be employed to approximate and implement the control law. Let

$$f^{-1}(0) = \{(x,u) \in \mathbb{R}^n \times \mathbb{R}^r \mid f(x,u) = 0\} \tag{85}$$

denote the equilibrium set of the control system. It turns out that for some point $[x^*,\, u(x^*)] \in f^{-1}(0)$ to be smoothly stabilizable $f^{-1}(0)$ must be an unbounded set. As an example [65], the system

$$\begin{array}{rcl} \dot{x}_1 & = & x_1^2 + x_2^2 - 1 \\ \dot{x}_2 & = & u \end{array} \tag{86}$$

is not smoothly stabilizable in the large since its equilibrium set defined by $x_1^2 + x_2^2 = 1$, $u = 0$ is bounded. Moreover, a general smooth system defined on a compact set $K \subset \mathbb{R}^n$ is never globally smoothly stabilizable since its equilibrium set is evidently bounded. Another necessary condition for C^∞ stabilizability obtained by Brockett [66] is that $f(x,u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ maps every neighborhood of (x^*, u^*) onto a neighborhood of zero. As an example, the system

$$\begin{array}{rcl} \dot{x}_1 & = & u_1 \\ \dot{x}_2 & = & u_2 \\ \dot{x}_3 & = & x_2 u_1 - x_1 u_2 \end{array} \tag{87}$$

does not have a continuous stabilizing feedback, since no point of the form $x = [0\ 0\ \varepsilon]^T$ is in the image of $f(x,u)$. The conditions motivated the introduction of discontinuous feedback laws. To this end, we are interested in a special kind of controllability (discussed in section 2): A point $x_0 \in K$ is *piecewise constantly* steered into a point $x \in \mathbb{R}^n$ if

$$\begin{array}{c} \text{for } x_0 \in K, x^* \in \mathbb{R}^n \qquad \exists T \in \mathbb{R},\ T > 0 \\ u(t) : [0,\, T] \to V \subset \mathbb{R}^r, \quad u \text{ p.w. constant} \\ \text{such that} \\ \phi_u(T,\, x_0) = x^* \end{array} \tag{88}$$

where $\phi_u(t, x_0)$ is the flow of $f(x, u)$ on $\mathbb{R}^n$ with initial value x_0. $V \subset \mathbb{R}^r$ is a finite set. If (88) holds for every point $x_0 \in K$ then x^* is said to be piecewise constantly *accessible* from the set K. Accessibility in general is the property that the above holds for arbitrary $u : [0, T] \to V \subset \mathbb{R}^r$. Equivalently, controllability means that every point $x_0 \in K$ can be steered into $x \in \mathbb{R}^n$.

It is clear that the controls $u_i \in V = \{u_1, \ldots, u_N\}$ generate different vector fields $f_i = f(x, u_i)$ where $i = 1, \ldots, N$. Every vector field when applied to the system will cause the state variable x to evolve in the direction tangent to it. A fundamental property of discontinuous controls is that it may generate additional directions (i.e. other than f_i) where the system may evolve.

Example (adapted from [64]): Consider a kinematic model of a car (front axis) with position $[x_1, x_2] \in \mathbb{R}^2$ and the angle of rotation $x_3 \in S^1$, i.e. the state space of the car is given by $M = \mathbb{R}^2 \times S^1$. The two applicable events are "drive" and "rotate" the corresponds to the two vector fields

$$f_1 = \begin{pmatrix} \sin x_3 \\ \cos x_3 \\ 0 \end{pmatrix} \quad \text{"drive"} \qquad f_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{"rotate"} \tag{89}$$

As the experienced driver knows, in order to park the car, one has to use the switching sequence given by "*roll – rotate – roll back – rotate back*", i.e. the resulting flow is obtained as the composition of the flows induced by the fields f_1, f_2, $-f_1$, and $-f_2$. It can be shown that the system moves infinitesimally in the direction orthogonal to the drive direction.

The new direction corresponds to the *Lie bracket* of the vector fields f_1 and f_2. The Lie Bracket of two vector fields is another vector field which measures the non-commutativeness of the flows induced by both vector fields. The Lie Bracket is instrumental in understanding nonlinear control systems since it implies that the points attainable from some point x_0 by the vector fields f_i lie not only in the directions given by linear combinations of f_i but also in the direction of the (iterated) Lie brackets of f_i. In local coordinates, the Lie bracket writes

$$[f_1,\, f_2]_x = \left.\frac{\partial f_2}{\partial x}\right|_x f_1(x) - \left.\frac{\partial f_1}{\partial x}\right|_x f_2(x). \tag{90}$$

In the above example we obtain $[f_1, f_2] = [-\cos x_3 \;\; \sin x_3 \;\; 0]^T$. For the interested reader some of the more formal mathematical statements regarding Lie brackets are included at the end of the section (see also [67, 68]). At present, we state the main result of the section, which is due to [65].

Theorem: Let $w = w(x)$ be a smooth feedback which locally stabilizes Σ at $(x^*,\; w^*) \in f^{-1}(0)$. Let K be a compact set and $w \in \operatorname{Int} V$ where V is the set of admissible controls. Then $w = w(x)$ has a piecewise smoothly stabilizing extension $u = \bar{u}(x) : \mathbb{R}^n \to V$ over K if and only if N_{x^*} is p.w. constantly accessible from K, where N_{x^*} is an open neighborhood of x^* such that its closure $\bar{N}_{x^*}$ is an invariant set of the closed loop system $\dot{x} = f(x, w(x))$ and $w = w(x)$ smoothly stabilizes Σ in $(x^*,\, u^*)$ over $\bar{N}_{x^*}$.

Let us highlight the conditions of the theorem. It is required that

(i) The system must be locally stabilized at the point $(x^*,\, u^*) \in f^{-1}(0)$.

(ii) The point x^* must be piecewise constantly accessible from a compact set K.

In order to verify (i) and realize the local stabilizing controller the methods described in chapter 4 are used. We know that if the linearized system $\Sigma_L = \left(\frac{\partial f}{\partial x}|_{x^*,u^*},\, \frac{\partial f}{\partial u}|_{x^*,u^*}\right) := (A,\; B)$ is stabilizable, i.e.

$$\operatorname{rank}(sI - A, B) = n \quad \text{whenever} \quad Re\, s \geq 0 \tag{91}$$

then the original system Σ is locally C^∞ stabilizable at (x^*, u^*).

Condition (ii) refers to our above discussion and equation (88). The theorem states that the semi–globally stabilizing control–law will be given by a set of smooth controls u_i, $i \in \Omega = \{1, \ldots, N\}$ and a switching sequence $\sigma(x) : \mathbb{R}^n \to \Omega$ that decides which control law is to be used depending on the state of the system. Notice that one of the controllers u_i corresponds to the local stabilizer at x^* while the others serve to enlarge the region of attraction of x^* in the closed loop system. Based on this existence theorem, $N+1$ neural networks can be used to implement u_i, $i = 1, \ldots, N$ and the switching function. The first N networks play the role of function approximators while the $N+1^{\text{st}}$ network is used as a classifier of the state space of the system.

Example: Find a global controller which stabilizes the origin of the system

$$\begin{aligned} \dot{x}_1 &= -x_1 + x_2\,(1 + x_1^2) \\ \dot{x}_2 &= u \end{aligned} \tag{92}$$

We verify condition (i) and find that the controllability matrix of the linearized system (at zero) has full rank. We define a linear feedback

$$u = -x_2 \tag{93}$$

to stabilize the nonlinear system (92) in a neighborhood of the origin. It is clear that the control law cannot be extended to an arbitrary compact domain $K \subset \mathbb{R}^2$, since the quadratic term x_1^2 will eventually dominate the stable part "$-x_1$" on the right hand side of (92).

Assuming global accessability of the origin (the question is addressed later) we define the piecewise smooth feedback

$$u = -\text{sign}(x_1)(1 + x_1^2) - \text{sign}(x_2) \tag{94}$$

which steers the system state to the region where the linear stabilizing control law is valid. This can be verified using the strict Lyapunov function

$$V = |x_1| + 0.5\,x_2^2 \tag{95}$$

The time derivative along the trajectories of the closed–loop system is $\dot{V} = -|x_1| - |x_2| < 0$. Figure 16 displays the local stability region and a sample trajectory which starts from outside this region and is steered to the origin using a piecewise smooth control. This is possible since the system is in fact semi–globally controllable.

Controllability of a nonlinear system depends upon the way the family of vector fields $F_V = \{f(x,\, u)\,|\, u \in V \subset \mathbb{R}^r\}$ generates a Lie algebra of differentiations of $C^\infty(\mathbb{R}^n)$. Consider the control system

$$\dot{x} = \underbrace{f(x)}_{\text{drift vf}} + \underbrace{g_1(x)u_1 + \ldots + g_r(x)u_r}_{\text{control vectorfield}}. \tag{96}$$

Given an initial point $x(0)$, we wish to determine the set of points which can be reached from $x(0)$ in finite time by a suitable choice of the input functions $u_1, \ldots, u_r$. The set of vector fields of (92) that can be obtained by applying different controls u spans a linear space Δ_x at any point $x \in \mathbb{R}^n$. Δ_x is called a *distribution* of vector fields at x and is a subspace of the tangent space $T_x\mathbb{R}^n$ of $\mathbb{R}^n$ at x. The set of points reachable from $x_0 \in K$ lie on the integral manifold of Δ_x. Frobenius' theorem states that Δ is integrable if and only if it is of constant rank and involutive, i.e. closed under Lie

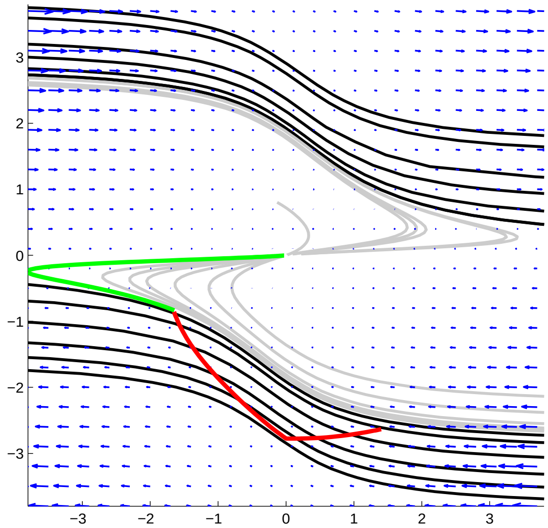


Figure 16: Piecewise smooth stabilization in the large. The controller consists of a piecewise smooth feedback of the form (94) (outside the local stability region) and (93) (in a neighborhood of the origin)

brackets: $f_1, f_2 \in \Delta \;\Rightarrow\; [f_1,\, f_2] \in \Delta$ at every point $x \in \mathbb{R}^n$. Thus, Σ is controllable if it is possible to construct an integrable distribution Δ of dimension n. This is achieved by successively including "new directions" to Δ obtained by forming higher–order Lie brackets of the vectorfields in F_V.

In the above example we have

$$f(x) = \begin{pmatrix} -x_1 + x_2(1+x_1^2) \\ 0 \end{pmatrix} \qquad \text{and} \qquad g(x) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{97}$$

We form a distribution of vector fields,

$$\Delta_x = \operatorname{span}\{g,\, [f,\, g]\} = \operatorname{span}\left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1+x_1^2 \\ 0 \end{pmatrix} \right\} \tag{98}$$

We include only those Lie brackets that are not already contained in lower–order Lie brackets of vectorfields in F_V. Given any point $x \in \mathbb{R}^n$, $Lie_x\, F_V$ is the linear space spanned by the tangent vectors of $Lie\, F_V$ at that point. The dimension of that space is called the *rank* of the Lie algebra $Lie\, F_V$ at the point $x \in \mathbb{R}^n$. The system Σ defined in (84) is globally controllable provided that

$$\operatorname{rank} Lie_x\, F_V = \dim \mathbb{R}^n \qquad \forall\, x \in \mathbb{R}^n \tag{99}$$

The condition is evidently fulfilled in our example, see equation (98). The construction of the distribution enables us to identify the set of reachable states *without* specifying the control input $u_1 \ldots u_r$. The set depends exclusively on geometrical properties of the system (96).

5.3 Global Observability

In the above, it was assumed that the state x of the system is accessible. Stabilization of a nonlinear system is more difficult if only a function of the state is available, in the way introduced earlier

$$y = h(x) \tag{100}$$

where $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is a C^∞ function representing some measuring device. As is well known from control theory and has also been stressed in section 2 of this paper, the critical property in this case is observability. Unlike the linear case it is impossible to pass directly from controllability conditions to observability because in the nonlinear domain there is no clear notion of duality.

The question is whether, given two initial states x_1 and x_2, one distinguish these initial states by observing the values of the "output" function $h(x)$ for any input sequence of length l. This is referred to as strong observability. A weaker form is generic observability which requires that almost any input sequence of length greater or equal to l, will uniquely determine the state. In Aeyels [69] it is demonstrated that almost all smooth output functions pair with an almost arbitrarily chosen smooth vectorfield to form a globally observable system provided that $l = 2n+1$ samples are taken. His proof makes use of a "generic transversality" theorem of differential topology to characterize observable flows. An extension of this result to "universal observability" has been given in [70] in which the output function is only continuous. In a paper by the first author [19], conditions under which strong observability holds have been investigated and it is shown how this can be used to construct global input-output models of the nonlinear system.

Comment 17: In this section we found that the extension of familiar concepts such as controllability, observability and stabilization to the nonlinear domain requires new mathematical tools and insights. Over the last thirty years a rich body of literature has been created at the intersection of differential geometry, topology and control theory which addresses the questions of global nonlinear control. However, this literature has not yet entered the engineering literature. Constructive methods for actually realizing global controllers based on geometric control theory are only sparely available and often involve a high level of mathematical formalism which is hard to grasp intuitively. In practice, our ideas for successful and ingenious design come from the "feel" we have about the effect of the chosen design on the system behavior. We have not reached this stage yet in the nonlinear domain but the material presented in this section is meant to be the first step in that direction. The fact that switching may be involved in order to overcome the topological obstruction to global stabilization must become common knowledge in much the same way as state space properties are in linear systems. The multiple model approach described in Section 4 provides the architecture for orchestrating the action of the neural networks involved in global nonlinear control.

6 Optimization and Optimal Control Using Neural Networks

As stated in Section 1.4, there is currently a great deal of interest in the use of neural networks in optimization and optimal control problems. These are problems in which optimization is carried out over a finite time. Even though the authors are not actively involved in research in this area at present, a few years ago the first author had an active program in the area of optimal control using neural networks, and consequently has some familiarity with such problems. Hence, for the sake of completeness, we wish to discuss this topic briefly and clarify the concepts involved.

In Section 6.1, the system to be optimized is assumed to be completely known. Optimal control is the principal tool, and is used to determine optimal control inputs as functions of time. The information collected is used to design neural networks to act as feedback controllers. In Section 6.2 the principal mathematical vehicle is Dynamic Programming. More importantly, the system to be controlled is either unknown or partially known. Hence it addresses problems of optimization under uncertainty and bears the same relation to Section 6.1 that adaptive control problems discussed in

Section 4 bear to feedback control theory. Finally, while the problems treated in this section involve optimization over a finite time interval unlike the adaptive control problems treated earlier, much of the motivation for using different approximation schemes, as well as the analytical difficulties encountered are very similar.

6.1 Neural Networks for Optimal Control

A question that arises in all decision making in both biological and engineering systems concerns the extent to which decisions should be based on memory and on on-line computation. For example, in some cases retrieval of stored answers may be preferable; in other cases solutions may have to be computed on-line, based on data obtained at that instant. In this section, we describe methods proposed in the last decade which utilize the above concepts for solving optimal control problems using neural networks.

Theoretical methods such as Pontryagin's Maximum Principle and Bellman's Dynamic Programming exist for determining optimal controls for nonlinear dynamical systems with state and control constraints. However, solutions for specific problems can rarely be determined on-line in practical problems. In optimal control theory the solutions are obtained as functions of time (i.e $u(t)$) which makes them nonrobust in practical applications, where feedback controllers (i.e controllers of the form $u(x)$) are required. During the period 1994-2000 very promising methods were proposed by Narendra and Brown for circumventing these difficulties. The authors suggested that open loop solutions of optimal control problems computed off-line could be used to train neural networks as on-line feedback controllers. Substantial progress was made, and the authors succeeded in proposing and realizing solutions to relatively complex optimal control problems. However, even as efforts to improve the scope of the approach were proving successful, the research was terminated due to a variety of reasons. Since the concepts may prove attractive to future researchers in the field, and also as an introduction to Section 6.2, they are briefly discussed here.

Function Approximation Using Neural Network

The gradual evolution of the "solve and store" approach from function approximation to optimal feedback control is best illustrated by considering several simple examples. The approach itself was motivated by a problem in a transportation system involving N electrical vehicles whose positions and velocities determine voltages which are critical variables of interest. A simplified version is given in Problem 6.1.

Problem 6.1: Consider the network shown in Figure 17a representing two electrical vehicles operating on a track. R_1, R_2, and R_3 are fixed resistances, while three other resistances depend on variables x_1 and x_2. The voltages V_1 and V_2 across R_1 and R_2 are nonlinear functions of x_1 and x_2 and are the values of interest. The objective is to estimate V_1 and V_2 for given values of x_1 and x_2. This is a standard application of a neural network as a function approximator. The problem was solved 1000 times for randomly chosen values of x_1 and x_2 in a compact set and a two input two output neural network shown in Figure 17b was trained. The network outputs were compared with true values for test inputs and had standard deviations of 0.0917 and 0.0786 respectively.

Problem 6.2: All the resistances in problem 6.1 were linear making the computations of currents and voltages simple for any choice of x_1 and x_2. In problem 6.2, one of the resistances was made nonlinear, making the computation of $V_i(x_1, x_2)$ $(i = 1, 2)$ substantially more complex. For every choice of x_1 and x_2 the problem had to be solved iteratively and the results used to train the neural

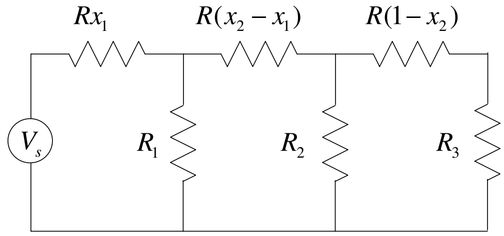


Figure 17a

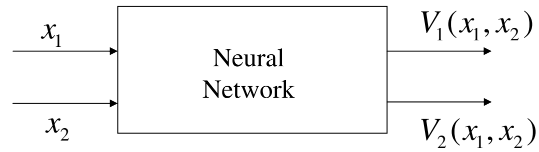


Figure 17b

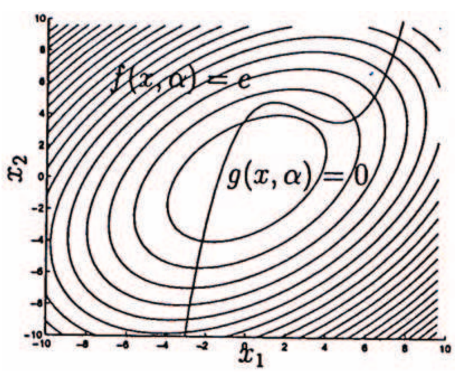
network. This problem introduces the principal difficulty encountered in many of the problems that follow, where available data has to be processed at many levels to obtain the relevant inputs and outputs necessary to train neural networks.

Parameter Optimization: The next stage in the evolution of the method was concerned with static optimization in which parameter values have to be determined which optimize a performance criterion in the presence of equality or inequality constraints. As shown below, a number of optimization problems have to be solved to obtain the information to train a neural network.

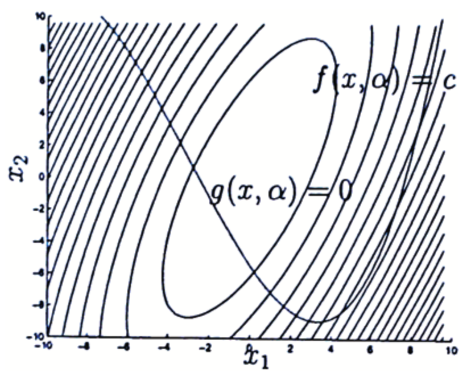
Problem 6.3: A function $f(x, \alpha)$ has to be minimized subject to an inequality constraint $g(x, \alpha) = 0$, where $x = [x_1, x_2]^T \in \mathbb{R}^2$, and α is a parameter. α can assume different values in an interval [0,1] and a neural network has to be trained to obtain the corresponding optimal values of $x_1(\alpha)$ and $x_2(\alpha)$.

$$\begin{aligned} f(x, \alpha) &= (\alpha + 0.5)x_1^2 + \tfrac{x_2^2}{(1+\alpha)^2} - \tfrac{1+\alpha}{2} x_1 x_2 \\ g(x, \alpha) &= (1 - \alpha)[10 + 0.1(x_1 + 10\alpha - 5)^3 + \alpha(-3x_1 - 10) - x_2] = 0 \end{aligned} \tag{101}$$

plots of $f(x, \alpha) = c$ (a constant) and $g(x, \alpha) = 0$ are shown for two typical values of α i.e $\alpha = 0.2$ and $\alpha = 0.8$, from which the optimal values $x_1(0.2)$, $x_2(0.2)$ and $x_1(0.8)$, $x_2(0.8)$ can be computed. The constrained parameter optimization problem was solved 100 times for 100 values of α to train the network to obtain optimal solutions for values of α not used before. It is seen from Figure 18 that the optimal values are discontinuous function of α.



(a) $\alpha = 0.2$



(b) $\alpha = 0.8$

Figure 18: Contour plots of $f(x, \alpha) = c$, where c is a constant, and constraint curves $g(x, \alpha) = 0$ for two values of α.

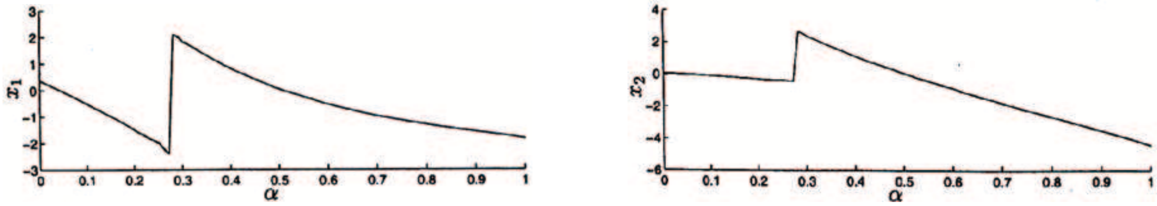


Figure 19: Optimal values of x_1 and x_2 as functions of α. Precomputed solutions (dotted lines) coincident with neural network approximations (solid lines).

Problem 6.4: (Dynamic Optimization) Problem 6.3 sets the stage for dynamic optimization in which neural networks can be used effectively as feedback controllers in optimal control problems. The general statement of such problems is first given and two examples are included to illustrate the different specific forms it can take. A system is described by the differential equation

$$\dot{x} = f[\alpha(t), u(t)] \quad x(t_0) = x_0 \tag{102}$$

where $f(\cdot)$ satisfies conditions to assure the existence and uniqueness of solutions in an interval $[0, T]$. $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^r$. The input $u(\cdot)$ is amplitude constrained and must lie in a unit cube $c \subset \mathbb{R}^r$. (i.e $||u_i(t) \leq 1 \; i = 1, 2, \cdots, r)$. The initial state $x_0 \in \mathbb{S}_0$ and the objective is to determine a control input which transfer x_0 to x_T and minimizes a performance criterion

$$J[u] = \int_0^T L[x(t), u(t)]dt \tag{103}$$

The optimal input and optimal trajectory are denoted by $u^*(t)$ and $x^*(t)$ respectively.

The above problem reduces to the solution of $2n$ differential equations of the form

$$\begin{aligned} \dot{x}(t) &= H_\lambda[x(t), \lambda(t), u(t)] \quad x(0) = x_0 \\ \dot{\lambda}(t) &= -H_x[x(t), \lambda(t), u(t)] \quad x(T) = x_T \end{aligned} \tag{104}$$

and the optimal input $u^*(t)$ is determined from the optimality condition

$$Inf_{u(t) \in C} H[x^*, \lambda, u] = H[x^*, \lambda, u^*] \tag{105}$$

This necessary condition confines the optimal solution to a small set of candidates. In the problems that we shall consider, it will be unique. Once $u^*(t)$ is known as a function of $x^*(t)$ and $\lambda(t)$, equations (104) correspond to a two-point boundary value problem (TPBVP) that can be solved off-line through successive approximations. This yields $x^*(t)$, $\lambda(t)$ and the corresponding $u^*(t)$ as functions of time.

Comment 18: Our interest is in training neural networks as feedback controllers for the above problem. Following the procedure we have adopted thus far, the above problem must be solved for numerous values of $x_0 \in S_0$ to obtain the necessary information.

By Bellman's Principle of Optimality given the optimal trajectory from x_0 to x_T, if $x^*(t_1)$ is on this trajectory, the optimal control from t_1 to T is merely $u^*(t + t_1)$ over the remaining $T - t_1$ units of time. If a family of optimal controls can be generated off-line for different values of the state, can be stored and used to train a neural network. Such a neural network will have for its inputs, $x(t)$ the state of the system, x_T the final state and T_r, the time to go (i.e $T - t$).

The following two problems are considered in [71], and only brief descriptions of the problems and the corresponding solutions are presented here.

Problem 6.4.1: (Minimum Time) A second order system is described by the differential equations $\dot{x}_1 = x_2,\ \dot{x}_2 = u$. The scalar input $u(\cdot)$ satisfies the amplitude constraint $|u(t)| \leq 1$, and the trajectories should lie outside a circular region in the state space. A specified initial state x_0 must be transferred to a final state x_f in minimal time.

It can be shown that the solution to the above problem either a) does not exist, or b) is bang bang, or c) piecewise continuous. The optimal trajectory in the state space and the corresponding input for a typical pair of states x_0 and x_f (generated by a neural network) is shown in Figure 20.

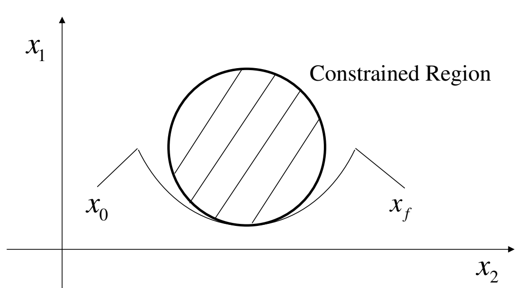


Figure 20a

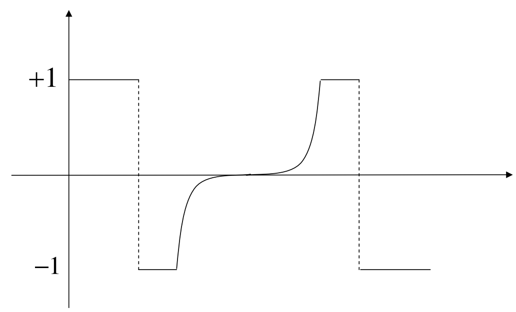


Figure 20b

Problem 6.4.2: (Minimum Energy) Another problem that is generally included in standard textbooks on control systems deals with minimum energy control, for which a closed form solution exists. Our interest is in determining a feedback controller for such a problem using a neural network and comparing the computational effort involved in the two cases.

A system is described by the linear state equation

$$\dot{x} = Ax(t) + bu(t) \tag{106}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$, and (A, b) is controllable. The objective is to transfer an initial state x_0 at time $t = 0$ to a final state $x(T) = x_T$ where T is specified, while minimizing the energy $E = \frac{1}{2}\int_0^T u(\tau)d\tau$.

The open loop optimal control $u^*(t)$ is given by

$$u^*(t) = -b^T W(T_r)^{-1}[x(t) - e^{-AT_r}x_f] \tag{107}$$

where T_r is the remaining time i.e $T - t$ and

$$W(T_r) = \int_0^{T_r} e^{-A\tau}bb^T e^{-A^T\tau}d\tau. \tag{108}$$

Assuming that the optimal control input and the corresponding response $x^*(t)$ have been generated for a number of triple (x_0, x_T, and T) a neural network having the structure shown in Figure 21 can be trained as a feedback controller.

Computational Advantage: A comparison of the computational times required to evaluate the Grammian matrix W, using standard techniques and a neural network reveals that the latter has a significant advantage (0.27 ms vs 600 ms).

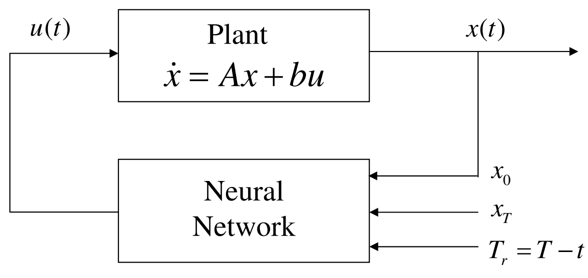


Figure 21

Comment 19: The method proposed here, when applied to a slowly varying terminal state also performed remarkably accurately.

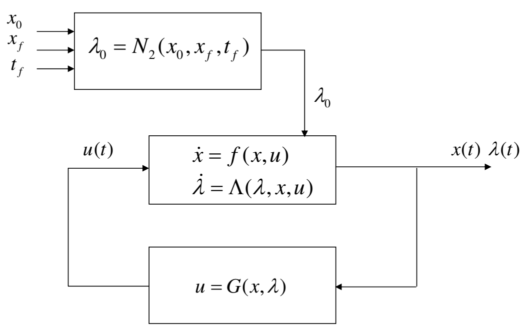


Figure 22a

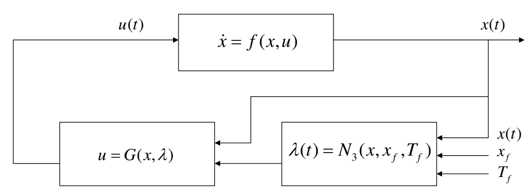


Figure 22b

Other Formulations: Thus far, the basic philosophy has been solve the optimal control problem off-line a number of times to obtain optimal solutions, and use the information at different levels to train a neural network to act as a feedback controller. Many other formulations were suggested in the late 1990s to reduce the computational effort. Two of them are shown in Figure 22. The idea is to convert the TPBVP into an initial value problem, which is substantially simpler to solve.

In particular, if given x_0, x_T and T a neural network can map them on to λ_0, the initial value of $\lambda(t)$, the $2n$ equations (104) can be integrated forwards in time, yielding $x(t)$ and $\lambda(t)$ simultaneously. This is shown in Figure 22a. A more robust method is shown in Figure 22b where a neural network yields $\lambda(t)$ corresponding to the triple (x_t, x_T and $T_r(= T - t)$).

6.2 Dynamic Programming in Continuous and Discrete Time

While optimization using dynamic programming is well known, we review briefly in this section the problem to be addressed and the principal concepts involved to aid us in the discussions that follow involving approximate methods that have been proposed in the literature. We first state the problem in continuous-time and later switch to discrete-time systems for the practical realization of the solutions.

Continuous Time (no uncertainty): A performance function $J[x, u, t]$ is defined as

$$J[x(t), u(t), t] = \phi(x(t_f)) + \int_{t_0}^{t_f} L[x(\tau), u(\tau), \tau] d\tau \tag{109}$$

and the objective is to determine a control law which minimizes the performance function. The state $x(t) \in \mathbb{R}^n$ of the system is governed by the differential equation

$$\dot{x}(t) = f[x(t), u(t), t] \tag{110}$$

and $u(t) \in \mathbb{R}^m$ denotes the control input. The initial and final times t_0 and t_f as well as the initial and final states $x(t_0) = x_0$ and $x(T) = x_T$ are assumed to be specified (in which case ϕ in (109) can be omitted) or the cost of the final state can be explicitly included in ϕ. $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is the instantaneous cost as in standard optimal control theory.

If J^* represents the optimum value of J we have

$$J^*[x(t), t] = \min_{t_0 \le \tau \le t_f} \left\{ \phi(x(t_f) + \int_{t_0}^{t_f} L(x(\tau, u(\tau), \tau)) d\tau) \right\} \tag{111}$$

If the integral in (111) is over the interval $[t, t_f]$ expressed as the sum of two integrals $\int_t^{t+\Delta t} L d\tau + \int_{t+\Delta t}^{t_f} L d\tau$, by the Principle of Optimality, the second integral must be optimal independent of the value of the first integral. Extending this argument to the case $\Delta t \to 0$, we obtain the Hamilton-Jacobi-Bellman equation

$$\frac{\partial J^*[x(t), t]}{\partial t} = -\min_{u(t)} \left\{ L(x(t), u(t), t) + \frac{\partial}{\partial x} J^*[x(t), t]^T f(x(t), u(t), t) \right\} \tag{112}$$

Partial differential equations of the form given in (112) are in general extremely hard to solve and exact solutions have been derived mainly for linear systems (110) with quadratic performance criteria. This why researchers became interested in neural networks, since they are universal approximators, and they permit the problem to be stated as a parameter optimization problem.

Comment 20: Before we discuss the use of neural networks to carry out the computation, it is important to bear in mind that, in general, the above problem admits only a non-smooth viscosity solution. Therefore assumptions about the performance measure J and the system (110) have to be carefully examined before attempting the use of neural networks.

Discrete-time (no uncertainty) In discrete time the procedure is substantially more transparent. The analogous problem can now be stated as follows:

$$J_T = J_{[t_0,T]}[x(k), u(k), k] = \phi(x(T)) + \sum_{k=k_0}^{k=T} L(x(k), u(k), k) \tag{113}$$

subject to

$$x(k+1) = f[x(k), u(k), k] \quad x(0) = x_0 \tag{114}$$

since $J_{[T,T]}[x(k), u(k), k] = \phi(x(T))$, we start by considering the transfer from $k = T - 1$ to $k = T$

$$J^*_{[T-1,T]} = J_{[T,T]} f(x, u, T-1) + L(x, u, T-1) \tag{115}$$

where (x, u) implies $(x(T-1), u(T-1))$. This is a one step optimization problem whose solution yields the optimal $u(T-1)$ for any initial state $x(T-1)$. The expression for $J_{[T-2,T]}$ is similar to (115), except that the optimal cost from $T-1$ to T has to be used in place of $J_{[T,T]}$. We therefore have

$$J^*_{[T-2,T]} = \min_{u(T-2)} [L(x, u, T-2) + J^*_{T-1}(x, T-1)] \tag{116}$$

Proceeding backwards in time, we have the general equation

$$J^*_{[T-k,T]}[x, T-k] = \min_{u(T-k)} \left\{ L(x, u, T-k) + J^*_{[T-k+1,T]} f[x, u, T-k] \right\} \tag{117}$$

where $k = 1, 2, \cdots, T$ is the stage number. Since the procedure involves computations backwards in time (as generally in optimal control), the computations have to carried out off-line. At state $T-k$, one obtains the optimal control law $u^*(T-k)$, by the minimization of the function in (113) as function $g(x(T-k), T-k)$ of the state and time. This in turn is used to compute the optimal performance $J^*_{[T-k,T]}[x, T-k]$, which in the next step is used ti derive $u^*(T-k-1)$. As stated earlier, except for the LQ (linear-quadratic) problem which has been extensively studied, an explicit expression of $g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ is vary hard to obtain. In such a case the state space is discretized and an exhaustive search over the space of admissible solutions is performed to find the optimum. The number of computations grow exponentially with the dimension of the state space, and this is generally referred to as the curse of dimensionality.

Discrete-time (system unknown): In the problems of interest to us in this chapter, accurate knowledge of complex nonlinear systems are not available. When $f(\cdot)$ in equation (117) is unknown or partially unknown, the methods described thus far cannot be applied and incremental minimization methods of the type treated in previous sections become necessary. These are collectively referred to as Approximate Dynamic Programming. In this section we confine our attention to discrete-time methods, in which most of the current work is being carried out. In these methods, which have been strongly influenced by reinforcement learning, one proceeds forward in time to estimate future rewards based on state and action transitions. Incremental dynamic programming is combined with a parametric structure (using neural networks) to reduce the computational complexity of estimating cost.

Adaptive Critics: In view of the complexity of the problems and the computations they entail, it is not surprising that a variety of methods have been proposed which include Heuristic Dynamic Programming (HDP), Dual Heuristic Programming (DHP) Globalized Dual Heuristic Programming (GDHP) and the so-called Action Dependent (AP) variants of HDP and DHP. All of them involve the use of neural networks to approximate the value function, the decision law, and the system dynamics, so that the problem is reduced to one of parameter optimization.

As in previous sections we shall try to address the basic ideas of the different methods. Following this, we shall briefly outline the differences between the various methods proposed and proceed to comment on convergence and stability questions they give rise to [72].

Unlike classical dynamic programming we proceed forward in time but nevertheless determine the optimal control law by using the same recurrence equation (117) where $J^*_{[T-k+1,T]}$ is replaced by an estimate of the optimal cost-to-go $\hat{J}^*_{[T-k+1,T]}$. So, at $k = T$ we solve

$$J^*_{[0,T]} = \min_{u(0)} \{L(x, u, 0) + \hat{J}^*_{[1,T]}[\hat{f}(x, u, 0)]\} \tag{118}$$

where $\hat{f}$ is the estimate of f at time $T-k = 0$. At step 1 the procedure is repeated, i.e

$$J^*_{[1,T]} = \min_{u(1)} \{L(x, u, 1) + \hat{J}^*_{[2,T]}[\hat{f}(x, u, 1)]\} \tag{119}$$

Again, $\hat{J}^*_{[2,T]}$ is used instead of $J^*_{[2,T]}$. The estimate of the optimal cost J^* as a function of the state has been updated using the cost that was actually caused by the control $u(0)$ in the previous instant. Repeating this process at every stage $k = T, \cdots, 1$, the estimate of the optimal policy $\hat{u}(x)$, the estimate of the plant dynamics $\hat{f}(x, u, k)$ and the estimate of the optimal cost-to-go $\hat{J}^*_{[T-k,T]}$ are evaluated.

The evaluation of all three functions at any stage k based on $x(k)$ is carried out iteratively over l cycles (so that k denotes the time instant, while l denotes the number of iterations at that time instant). It is claimed that this procedure will result in the convergence of $\hat{u}(x, k)$ to $u(k)$ and $\hat{J}^*$ to J^*.

While the optimization procedure described above contains the essential ideas of the four methods mentioned earlier, they differ in the exact nature in which the various steps are executed (e.g the nature of the functionals and/or their derivatives that they attempt to realize) as shown below.

Heuristic Dynamic Programming (HDP) is essentially the procedure outlined above, and represents conceptually the simplest form of design. It uses two neural network to approximate the value function and the decision law. In Dual Heuristic Programming (DHP) neural networks are used to approximate the derivatives of the value function with respect to the state variables (used in the computation of the control law). Empirically, the resulting updating laws have been shown to converge more rapidly than HDP, although at the cost of more complexity, since a vector function is approximated rather than a scalar. This also results in the relationship between the updatings of the value function and the control law becoming more complicated. The Globalized Dual Heuristic Programming attempts to combine the fast convergence of DHP with the easier implementation of HDP. Action Dependent (AD) Methods methods modify the above three methods by using a value function $V(x, \alpha)$ in place of $V(x)$, where α is the control action. Once again empirically this has been shown to result in improved convergence rate. The reader is referred to [73] for details concerning all four methods.

Comment 21: (Convergence) The control law is determined as the one that minimizes the instantaneous cost $L(x, \hat{u}(x), T-k)$ in the value function $J^*_{[T-k,T]}$. At every stage, only an estimate $\hat{J}^*_{[T-k,T]}$ is available and hence the quality of the resulting control depends on how close the estimate is to the actual optimal trajectory. Hence, $\hat{u}(x)$ is suboptimal in general. But then, if it is applied in the next step of the recursive procedure, it does not really minimize the instantaneous cost $L(x, \hat{u}(x), T-k)$. Hence L does not contribute to the cost in the same way that the optimal cost L^* would, and this, in turn, may distort the estimate of the value function. A word must be added regarding the logic behind the forward programming scheme. The improvement of $\hat{J}^*_{T-k,T}$ is of no use for determining the next control input. As expected, the procedure approximates the cost only in hindsight, i.e. after the control has been applied. However, in the procedures proposed many iterations are carried out for the same time instant.

Comment 22: (Stability) Since $\hat{u}(x)$ applied to the system is suboptimal we have to assume that it does not destabilize the system.

Comment 23: (Stability) If J^* is seen as the output of a system and $\hat{J}^*$ is its estimate, then clearly, by testing the system and comparing the output to the estimate, one gains information that can be used to adjust the estimate at the next instant of time. This is similar to the viewpoint adopted in adaptive control where the adjustment is performed on a suitably parameterized system model. Accepting this analogy temporarily, we recall that a series of questions had to be answered in order to prove stability of the adaptive process. These questions are concerned with the way in which the adjustment of parameters and the application of the control are to be interwoven so as to keep all the signals in the system bounded. A similar issue arises in the present case, since the system is controlled at the same time that an estimate of the cost to go is generated.

Conclusion: (i) At the present time, we do not believe that the methods described in Section 6.2 can be used to stabilize an unknown nonlinear dynamical system online while optimizing a performance function over a finite interval. However, as in Section 4, stability may be arrived at provided that the uncertainty regarding the plant is sufficiently small and initial choices are sufficiently close to the optimal values. It should also be mentioned that a large body of empirical evidence exists which demonstrates the success of the method in practical applications.

(ii) To the authors' knowledge the stability of even a linear adaptive system optimized over a finite interval has not been resolved thus far. Since it is likely that conditions for the latter can be derived, we believe that it should be undertaken first so that we have a better appreciation of the difficulties encountered in the nonlinear case.

(iii) In problems discussed in Section 6.1 it was assumed that the dynamics of the plant was known. In light of the discussions in Section 6.2, it may be worth attempting the same problem (i.e with plant uncertainty) using optimal control methods described in Section 6.1. By providing an alternative viewpoint it would complement much of the work that is currently in progress using dynamic programming methods.

(iv) Finally, the authors also believe that the multiple model based approach [41] (which is now used extensively in many different fields) may have much to offer to the problems discussed in this section. While it is bound to be computationally intensive, the use of multiple trajectories at each stage would increase the probability of convergence. The reader is referred to [41]-[43] where a switching scheme is proposed to achieve both stability and accuracy.

7 Applications of Neural Networks to Control Problems

A computer search carried out by the first author about ten years ago revealed that 9955 articles with the title "neural networks" were published in the engineering literature over a five year period of which over 8000 dealt with problems related to function approximation. Of the remaining 1500, approximately 350 were related to applications, which were concerned with theory, experiments in the laboratory, and primarily simulation studies. Only 14 of the roughly 10000 articles dealt with real applications.

The authors have once again searched the engineering literature and have concluded that matters are not very different today. For a comprehensive and systematic classification of neural network based control applications they refer the reader to the paper by Agarwal [74] They are also aware that a number of exciting applications of neural networks have existed in the industrial world during the entire period, and that many of them do not appear in journal articles for proprietary reasons.

In this section we present a few carefully chosen applications. These have been collected from books and technical articles, and from friends in industry through private communication. These fall into three distinct categories. The first consists of applications which are in one way or another related to the issues raised in Sections 3 and 4, and indicate the extent to which theory plays a role in design. Sufficient details are provided about these problems. In the second category are included those applications which emphasize the practical considerations which determine the choices that have to be made. The third category consists of novel applications where the emphasis is not on mathematical techniques but on the ingenuity and creativity of some of our colleagues.

Application 1: Controller in a Hard Disk Drive [75]

This concerns a high performance servo controller for data acquisition in a hard disk drive using an electromechanical voice-coil motor (VCM) actuator. Such high performance controllers are needed due to the rapid increase in data storage density.

The Model: If q is the position of the actuator tip and $\dot{q}$ is its velocity, the model of such a system is described by the equation

$$M\ddot{q} + F(q,\dot{q}) = u \tag{120}$$

where M is the system inertia and is unknown, and the smooth function $F(\cdot)$ is also unknown but bounded with a known bound K_F i.e $|F| \leq K_F$. Further, $(q,\dot{q}) \in S$ where S is a compact set and is known.

RBF Network: Since the domain over which F needs to be approximated is compact and F is bounded, the approximation can be achieved using a radial basis function network such that

$$F(q,\dot{q}) = \theta^T R(q,\dot{q}) + E_F \tag{121}$$

where R is a vector of radial basis functions, θ is an unknown constant vector and $|E_F| \leq K_E$ is an error term. Since the state is bounded, the principal concern is accuracy.

The Objective: If $\hat{M}$ and $\hat{\theta}$ are estimates of M and θ, the objective is to determine adaptive laws for updating the estimates and at every instant use the estimates to determine the control input to transfer the initial state to the desired final state.

Adaptive Laws and the Control Law: The following adaptive laws for determining the parameter estimates and control law for generating the input u were used.

$$\begin{array}{rcl} \dot{\hat{M}} & = & -\gamma \ddot{q}_r r \\ \dot{\hat{\theta}} & = & -\Gamma \phi(q,\dot{q}) r \\ u & = & \hat{M}\ddot{q}_r + \hat{\theta}^T \phi(q,\dot{q}) - (K_d \dot{r} + K_r + K_i \int_0^t r d\tau) - K_E \mathrm{sgn}(r) \end{array} \tag{122}$$

where $\epsilon = q - q_d$, $\dot{q}_r = \dot{q}_d - \lambda_\epsilon$, and $r = \dot{q} - \dot{q}_r = \dot{\epsilon} + \lambda\epsilon \qquad \lambda > 0$ are error signals.

Comment 24: This is a well defined problem for which neural networks can be designed since the compact region in which the trajectories lie is known a priori, and all radial basis functions were carefully chosen to cover the operational range of the controller.

Application 2: Lean Combustion in Spark Ignition Engines [76]

This problem discussed by He and Jagannathan is a very interesting application of neural networks in control. At the same time it also raises theoretical questions related to those discussed in Section 4. It deals with spark ignition (SI) engines at extreme lean conditions.

The control of engines at lean operating conditions is desirable to reduce emissions and to improve fuel efficiency. However, the engine exhibits strong cyclic variations in heat release, which is undesirable from the point of view of stability. The objective of the design is consequently to reduce cyclic variations in heat release at lean engine operation.

The system to be controlled is described by the equations

$$\begin{array}{rcl} x_1(k+1) & = & f_1(x_1(k), x_2(k)) + g_1(x_1(k), x_2(k))x_2(k) + d_1(k) \\ x_2(k+1) & = & f_2(x_1(k), x_2(k)) + g_2(x_1(k), x_2(k))u(k) + d_2(k) \end{array} \tag{123}$$

where $x_2(k)$ is the mass of fuel before kth burn, $x_1(k)$ is the mass of air before kth burn, and u, the control variable, is the change of mass of fuel per cycle. f_1, f_2, g_1, and g_2 are smooth unknown functions of their arguments and $g_i(k)$ are known to lie in the intervals $[0, g_{mi}]$, $i = 1, 2$. $d_1(\cdot)$ and $d(\cdot)$ are external disturbances.

The Objective: is to maintain $x_1(k)$ at a constant value (Problem 2) and reduce the variations in the ratio $x_2(k)/x_1(k)$ over a cycle, by using u as the control variable.

The Method: The authors use a nonlinear version of back stepping (which we have discussed in Section 4) and use $x_2(k)$ as the virtual control input and $u(k)$ as the control input. The former requires the adjustment of the weight vector w_1 of a neural network, and the latter that of the weight vector w_2 of a second network. The adaptive laws are

$$\begin{array}{rcl} w_1(k+1) & = & w_1(k) - \alpha_1\phi(z_1(k))(w_1^T(k)\phi(z_1(k)) + k_1e_1(k)) \\ w_2(k+1) & = & w_2(k) - \frac{\alpha_2}{k_2}\sigma(x_2(k))(z_1^T(k)\sigma(z_2(k)) + k_2e_2(k)) \end{array} \tag{124}$$

where $z_1(k) = [x_1(k)\ x_2(k)\ x_{1d}]^T$ and $z_2(k) = [x_1(k)\ x_2(k)\ w_1(k)]^T$. The authors claim that the objectives set forth earlier are achieved and that the performance is highly satisfactory.

Comment 25: While this is a very ingenious application, we do not agree with the theoretical arguments used to justify the structure of the controllers. Significantly more information about the manner in which the basis functions were chosen, and the conditions under which the neural networks were trained need to be known before questions of theoretical correctness can be argued.

Application 3: MIMO Furnace Control [77]

An application of neural networks for temperature control was developed by Omron Inc in Japan over ten years ago. It is included here since it exemplifies the set-point regulation problem of a nonlinear system posed in Section 3, and discussed in Section 4. It deals with an MIMO temperature control system. The range of the temperatures, and the accuracy of the set point control affect the final products in an industrial process. The objective is to regulate temperature in three channels by bringing up the set point during startup as quickly as possible avoiding overshoots.

The system is open loop stable so that neural networks can be trained to obtain both forward and inverse models of the three channels of interest. The identification models, in turn, are used to train the controllers. A comparison of the proposed scheme with a self-tuning controller and a PID controller demonstrated that neuro-controllers are considerably more robust than the other two, and can also cope with changes in the dynamics of the plant.

Application 4: Fed-batch Fermentation Processes [78]

Bio-fermentation processes, in which microorganisms grown on a suitable substrate synthesize a desired substance, are used widely to produce a large number of useful products. The application of neural network based controllers discussed by Bǒsković and Narendra in 1995 for the control of fed-batch fermentation processes is an appropriate one for examination in the present context.

It reveals clearly that the efficacy of a control strategy in a practical application depends upon a number of factors which include the prior information needed to implement the controller, the difficulty encountered in choosing design parameters, stability and robustness issues, the effect of measurement noise, and the computational aspects involved in the implementation of the control algorithm.

The paper deals with the control of a distinctly nonlinear system whose dynamics is not known precisely, whose initial conditions and parameters can vary, and whose inputs have saturation constraints. The system is open loop stable and all its five state variables are accessible. The two inputs of the system $u_1(\cdot)$ and $u_2(\cdot)$ are to be determined to maximize the production of microorganisms (i.e $x_1(k)$) in the interval $[0, T]$, while assuring that one of the state variables $x_4(k)$ (ethanol) is small.

On the basis of the results presented in the paper it became clear that the method to be used for controlling a fermentation process would depend upon several factors including the extent to which parameter values and initial conditions may vary, the error the designer would be willing to tolerate, and prior knowledge concerning nonlinearities.

It was concluded that linear adaptive controllers and neuro-controllers were the only two viable alternatives. Even among these, if accuracy and robustness are critical issues, neural network based control is distinctly preferable.

Application 5: Automotive Control Systems

The research team at Ford Research Laboratory, under the leadership of Feldkamp has been investigating for about fifteen years the efficacy of neural network techniques for addressing different problems that arise in automotive systems. The experiments that they have carried out under carefully controlled conditions, the meticulous manner in which they have examined various issues, as well as their candid comments concerning the outcomes have had a great impact on the confidence of the neural network community in general, and the authors in particular, regarding the practical applicability of neural networks. The reader is referred to [13][79]-[82] which are papers presented by Feldkamp and his co-workers at the Yale Workshops during the period 1994-2005.

Idle speed control implies holding the engine speed at or near an externally supplied target speed in the presence of disturbances. The latter include load from the air conditioning system, load from electrical accessories such as windows, and load from power steering. Some of these loads may be large enough to stall a poorly controlled engine. The controls to the system are throttle position and spark advance. Since these have different dynamics and control authority, and interesting aspect of the control problem is in coordinating the control actions effectively.

In [83] an attempt was made to develop a recurrent neural network idle speed controller for a 4-cylinder engine. An identification stage preceded the design of a controller which was trained on-line. Training was carried out with no disturbance, single disturbance, and combinations of disturbances during which engine speed measurements were compared to the target speed, and extended Kalman filter rather than simple gradient updates were used, The latter involved truncated back propagation through time.

A variation of the above procedure was to identify the system using a recurrent network and use it off-line to train a controller.

Over the 1990s, the same approach was used to obtain nonlinear models for active suspension and anti-lock braking. In all cases the best estimates of noise, parameter uncertainty, measurement error and actuator delays were used.

Of current interest to the group is a setting in which a nominal model is given and the objective is to develop a controller with an adjustable tradeoff between performance and robustness.

Feldkamp has been arguing strongly for a long time for the use of recurrent networks as controllers. Training such networks off-line eliminates stability issues provided an initial set of values can be found which will eventually yield a satisfactory reduction in the chosen cost function. Based on this, the first author is starting a program at Yale to study the theoretical properties of recurrent neural networks

Application 6: Biped Walking Robot [84]

An exciting, ambitious, and very complex application, whose development over many years was followed by the first author was the biped walking robot designed, implemented, and tested by Miller and his co-workers at the University of New Hampshire.

It encompassed almost all the difficulties enumerated in Section 4, including unstable nonlinear dynamics, time delays in the control loops, nonlinear kinematics that are difficult to model accurately, and noisy sensors. In spite of these difficulties walking control strategies were tested and extended in studies over generations of bipeds. From the point of view of this paper it brings to focus some of the theoretical questions concerning stability raised in Section 4.

Dynamic walking promises higher speeds, improved walking structures, and greater efficiency. However, this also implies exploring new regions in the state space to learn better strategies, which automatically brings in its wake stability questions.

Three neural networks controlled front to back and side to side balance, and good foot contact. The first network controlled instantaneous front/back position of the hips relative to the feet, the second network predicted the correct amplitude and velocity of side to side lean during each step. The third network was used to learn kinematically consistent postures.

All three networks have to operate in new regions to learn the proper strategies, and in all cases instability is a distinct possibility. Frequent human support was needed to keep the biped from falling during the learning process when it was learning to perform satisfactorily in unfamiliar regions.

Application 7: Real-time Predictive Control in the Manufacturing Industry

The following is a recent application from industry, but for proprietary reasons details concerning the process controlled are not provided.

In a manufacturing plant, the sensors used to measure the relevant outputs do not report actual readings frequently enough for effective feedback control. Further, the plant has unmodeled dynamics which consist of several parts that are very hard to model from first principles. Hence, the system presented the right opportunity to develop a dynamic model using the neural network methods described in Section 4.

The neural network model was used as a virtual sensor, and based on the estimate of the state provided by the model, an intelligent set point generator was developed to determine set points for two control variables. A 32% improvement in performance of the overall system was achieved.

Application 8: Multiple-Models: Switching and Tuning

The multiple-model switching and tuning method proposed in [41] is currently widely used in very different areas. Brief descriptions of two applications are given below:

(i) Anomaly Detection in Finance

The stock markets in the United States employ the industry's most sophisticated real-time surveillance systems to ensure investor protection and a fair and competitive trading environment. They have designed and developed unusual market activity detecting alerts for real-time market surveillance systems. The "Switch and Tune" approach has been used to develop piecewise models of various regions in the underlying operating space. New regions are explored and flagged by the use of anomaly detection methods using neural networks, while well traversed spaces are learned using a local classifier which classifies activity as an anomaly or not.

(ii) Reconfigurable Control of Space Structural Platforms

In a broad spectrum of aerospace applications, achieving acceptable performance over an extended operating range may be difficult due to a variety of factors such as high dimensionality, multiple inputs and outputs, complex performance criterion and operational constraints. The multiple model based "switching and tuning" philosophy described in Section 4 is ideally suited for such problems.

One example of such a system is described in [85]. The system considered is a flexible structure with nonlinear and time-varying dynamics. An adaptive radial basis function network is used to identify most of the spatio-temporal interaction among the structure members. A fault diagnosis system provides neurocontrollers with various failure scenarios, and an associative memory compensates for catastrophic changes of structural parameters by providing a continuous solution space of acceptable control configurations. As stated in Section 4, the latter are designed a priori.

Application 9: Biological Control Structures for Engineering Systems

In the introduction it was sated that control theorists were attracted to the new field of neurocontrol in the 1980s, inspired by the ability of biological systems to interact with uncertain complex environments. Thus far, most of the systems that we have discussed are engineering systems. The following interesting application by Doyle et al. [86] is a notable exception. In it, the authors attempt reverse engineering by using biological structures for applications in control systems.

The baroreceptor vagal reflex is responsible for short-term blood pressure control. The authors make a very good case that it provides an excellent biological paradigm for the development of control strategies for multiple-input single-output (MISO) Processes. The components of the system have well defined control analogues. The central nervous system is the "controller". The transducers in the major blood vessels (baroreceptors) are the "'sensors", and the sympathetic and vagal postganglionic motor neurons in the heart and vessels are "actuators". Demand (eating, exercise), external inputs (cold weather) emotional state (joy, anger), and anticipated action (postural adjustment) correspond to "time-varying environments". Maintaining blood pressure around a set point dictated by cardiovascular demands is the objective. The control system performs a variety of tasks which includes integration of multiple inputs, noise filtering, compensation for nonlinear features of cardiovascular function, and generation of a robust control input. The primary objective of the authors is to understand the above system and then to mimic the functions for process control application.

The MISO control architecture employed in the baroreceptor reflex consists of two parallel controllers in the central nervous system: the sympathetic and the parasympathetic systems. While the response of the latter is relatively fast [2-4.5 secs], the response of the former is slow [10-80 secs]. However the faster control is "expensive" while the slower control is acceptable. The brain coordinates the use of the two controllers to provide effective blood pressure control while minimizing the long-term cost of the control actions. This is one of many functions discussed by the authors which is used for reverse engineering. We consider this a truly noteworthy application.

Application 10: Preconscious Attention Control

An application that is distinctly different from those discussed thus far, which combines engineering and biology, and which was brought to the attention of the first author, concerns preconscious attention control. Automobile collision avoidance is an example of a problem where improved recognition results in reduced reaction time, but where visual distraction decreases overall performance. In this application, the authors [87] seek to aid an operator in recognizing an object requiring attention by first presenting an object image to the preconscious region of visual awareness (the region in which neural activity registers without the user's awareness). Preconscious exposure results in subsequent processing of identical images to require less neural activity in identification (known as visual priming). Visual priming results from the plasticity of the brain and is known to facilitate recognition, since recognition becomes a function of memory as well as awareness.

Since each individual is assumed to have unique visual sensitivities that evolve with all prior experience, a neural network is used to derive an operator specific sensitivity to images presented outside of awareness.

From an engineering point of view, the interesting feature of the indirect control procedure is that the output of the reference model is a function of the state of the plant.

The effect of visual priming is found by comparing an operator's reaction time with that of a non-primed operator. The observed reduction in reaction time is directly proportional to the reduction in neural activity. For further details concerning this application the reader is referred to [87].

8 Comments and Conclusions

The paper examines the current status of neurocontrol and the methods available for infinite time and finite time adaptive control of nonlinear systems. The theoretical basis for their design, the determination of the existence of appropriate maps, and the realization of identifiers and controllers when such maps exist are discussed. The emphasis of the paper is on the simple ideas that underlie the different methods. Spread throughout the paper are comments to relate the procedures proposed to well established concepts in adaptive control.

Section 2 presents many of the basic results from control theory and adaptive control theory for easy reference, and concludes with a statement of problems for investigation in the following sections.

Since neural networks are the principal components used to cope with nonlinearities, their structures as well as the adaptive laws used to train them are discussed in Section 3.

Section 4 deals with the methods that have been proposed to identify and control nonlinear dynamical systems. The authors believe that the method based on linearization is one that is general and can be rigorously justified. It is valid in a neighborhood Ω of the equilibrium state where the linear terms dominate the nonlinear components. In such cases, using the inverse function theorem and the implicit function theorem all the problems investigated in the past in adaptive control theory can be revisited after including nonlinear terms in the description of the plant. Stability can be assured using linear methods and fast adaptation is possible. Slow adjustment of the parameters of the neural networks can be used, without affecting stability, to achieve greater accuracy. Extensive simulation results have shown that Ω need not be small.

While the emphasis in Section 4 is on linearization based methods, several others proposed by different authors are also discussed. The assumptions made by them concerning the dynamics of the plant are critically examined. In particular, the authors believe that the assumptions regarding the existence of basis functions needs considerably more justification, before they can be accepted as being theoretically rigorous. They also believe that a vast number of contributions made in recent years extending the well known back stepping procedure to general nonlinear systems cannot be theoretically justified.

An introduction to nonlinear control concepts is included in Section 5. These become relevant when the nonlinear terms dominate the linear terms in regions of the state space far from equilibrium, and will be needed when attempts are made in the future to control dynamical systems in those regions. While these concepts have not, become part of the main stream thinking in the neural network community, the authors believe that the latter will take to them enthusiastically in the future, when their power and scope become evident as experience with distinctly nonlinear phenomena increases.

Finite time optimization and optimal control is the topic of Section 6. Optimal control theory and dynamic programming are the principal mathematical tools used here. In Section 6.1, the dynamics of the plant is assumed to be known and optimal control theory is used to design feedback controllers. In Section 6.2, the dynamics of the plant are assumed to be unknown, and approximate dynamic programming methods are proposed. The same questions that arise in nonlinear adaptive control also arise in this case, and the authors are of the opinion that the arguments used can be rigorously justified only when the initial trajectories are in the neighborhoods of the optimal trajectories.

The paper concludes with a section on applications. Some of the applications are based on the analytical principles developed in the paper, while others have only a tenuous relation to them. However, most of them are both novel and creative and, like many other similar applications in the history of control, call for theoretical explanations and thereby catalyze research.

Nonlinear control using neural networks is still in its infancy, but there is little doubt that great opportunities abound both in theory and in practice.

Acknowledgement

The first author would like to thank Kannan Parthasarathy, Asriel Levine, Snehasis Mukhopadhyay, Sai-Ming Li, João Cabrera, Lingji Chen, and Osvaldo Driollet, who were his former graduate students, and Jovan Bŏsković his former Postdoctoral Fellow, all of whom collaborated with him on different aspects of the research reported here. In particular, he would like to thank Snehasis, João, and Lingji for many insightful discussions in recent years.

He would also like to acknowledge the help received from Lee Feldkamp, Santosh Ananthram, and Alex Parlos concerning recent applications, and the generous support from the National Science Foundation (through Paul Werbos) over a period of fifteen years when much of the work on linearization was done. Finally, the authors would like to thank the editors Andreas Pitsillides and Petros Ioannou for their invitation to contribute to this book and for their patience and support.

References

[1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[2] N. Wiener, *Cybernetics.* Cambridge, MA: M.I.T Press, 1948.

[3] D. O. Hebb, *Organization of Behavior: A Neuropsychological Theory.* New York: Wiley, 1949.

[4] F. Rosenblatt, "The perceptron: A perceiving and recognizing automaton," *Technical Report 85-460-1, Cornell Aeronautical Laboratory*, 1957.

[5] B. Widrow, "Generaliztion and information stroage in networks of adaline "neurons"," *Self-Organizing Systems, M. Yovitz, G.Jacobi, and G. Goldstein, Editors, Spartan Books, Washington, D.C*, pp. 435–461, 1962.

[6] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry.* Cambridge, MA: M.I.T Press, 1969.

[7] P. J. Werbos, *Beyond Regression: New tools for prediction and analysis in the behavioral sciences.* Cambridge, MA: PhD thesis, Harvard University, 1974.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by error propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, D.E. Rumelhart and J.L. McClelland, Editors*, vol. 1, pp. 318–362, 1986.

[9] G. Cybenko, "Approximation by superpositiion of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303–314, 1989.

[10] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[12] K. S. Narendra and K. Parthasarathy, "Identification and control of dyanmical systems using neural netwroks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, 1990.

[13] L. A. Feldkamp, G. V. Puskorius, L. I. David, and F. Yuan, "Enabling concepts for applications of neuraocontrol," in *Proceedings of the Eighth Yale Workshop*, (New Haven, CT), pp. 168–173, 1994.

[14] W. J. Rugh, *Linear System Theory.* Englewood Cliffs, NJ: Prentice Hall, 1995.

[15] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems.* Englewood Cliffs, NJ: Prentice Hall, 1989 (Dover 2005).

[16] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design.* New York: John Wiley and Sons, 1995.

[17] D. Seto, A. M. Annaswamy, and J. Baillieul, "Adaptive control of nonlinear systems with triangular structure," *IEEE Transactions on Automatic Control*, vol. 39, pp. 1411–1428, 1994.

[18] R. J. Williams, "Adaptive state representation and estimation using recurrent connectionist networks," *Neural Networks for Control, W. T. Miller, R. S. Sutton, and P. J. Werbos, Editors, Cambridge: MIT Press/Bradford Books*, 1990.

[19] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks: Observability, identification, and control," *IEEE Transactions on Neural Networks*, vol. 7, pp. 30–42, 1996.

[20] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for non-linear systems. i. deterministic non-linear systems," *International Journal of Control*, vol. 41, pp. 303–328, 1985.

[21] K. S. Narendra and J. H. Taylor, *Frequency Domain Criteria for Absolute Stability.* Academic Press, 1972.

[22] P. C. Parks, "Lyapunov redesign of model reference adaptive control systems," *IEEE Transactions on Automatic Control*, vol. 11, pp. 362–367, 1966.

[23] K. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Networks*, vol. 6, pp. 801–806, 1993.

[24] B. A. Pearlmutter, "Gradient calculations for dynamic recurrent neural networks: A survey," *IEEE Transactions Neural Networks*, vol. 6, pp. 1212–1228, 1993.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by backpropagating errors," *Nature*, vol. 6, pp. 533–536, 1986.

[26] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc IEEE*, vol. 78, pp. 1550–1560, 1990.

[27] K. S. Narendra and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 252–262, 1991.

[28] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neur. Computation*, vol. 1, pp. 270–280, 1989.

[29] L. E. McBride and K. S. Narendra, "Optimization of time-varying systems," *IEEE Transactions, PGAC*, 1965.

[30] A. U. Levin, *Neural Networks in Dynamical Systems.* Doctoral Dissertation: Yale University, 1992.

[31] S. Mukhopadhyay, *Synthesis of Nonlinear Control Systems Using Neural Networks.* Doctoral Dissertation: Yale University, 1994.

[32] J. B. D. Cabrera and K. S. Narendra, "Issues in the application of neural networks for tracking based on inverse control," *IEEE Transactions on Automatic Control*, vol. 44(11), pp. 2007–2027, 1999.

[33] S.-M. Li, *Pattern Recognition and Control.* Doctoral Dissertation: Yale University, 1999.

[34] L. Chen, *Nonlinear Adaptive Control of Discrete-tme Systems Using Neural Networks and Multiple Models.* Doctoral Dissertation: Yale University, 2001.

[35] M. J. Feiler, *Adaptive Control in the Presence of Disturbances.* Doctoral Dissertation: TU Munich, 2004.

[36] L. Chen and K. S. Narendra, "Identification and control of a nonlinear discrete-time system based on its linearization: A unified framework," *IEEE Transactions on Neural Networks*, vol. 15(3), pp. 663–673, 2004.

[37] L. Chen and K. S. Narendra, "Nonlinear adaptive control using neural networks and multiple models," *Automatica*, vol. 37, pp. 1245–1255, 2001.

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

[39] K. S. Tsakalis and P. A. Ioannou, *Linear Time Varying Systems: Control and Adaptation.* Upper Saddle River, NJ: Prentice-Hall, 1993.

[40] S. Mukhopadhyay and K. S. Narendra, "Disturbance rejection in nonlinear systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 4, pp. 63–72, 1993.

[41] K. S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *IEEE Transactions on Automatic Control*, vol. 42, pp. 171–187, 1997.

[42] K. S. Narendra, O. A. Driollet, M. Feiler, and K. George, "Adaptive control using multiple models, switching and tuning," *International Journal of Adaptive Control and Signal Processing*, vol. 17, pp. 87–102, 2003.

[43] K. S. Narendra and O. A. Driollet, "Stochastic adaptive control using multiple models for improved performance in the presence of random disturbances," *International Journal of Adaptive Control and Signal Processing*, vol. 15, pp. 287–317, 2001.

[44] K. S. Narendra and S. Mukhopadhyay, "Adaptive control of nonlinear multivariable systems using neural networks," *Neural Networks*, vol. 41, pp. 737–752, 1994.

[45] P. L. Falb and W. A. Wolovich, "Decoupling in the design and synthesis of multivariable control systems," *IEEE Transactions on Automatic Control*, vol. 12, pp. 651–659, 1967.

[46] K. S. Narendra and N. O. Oleng, "Exact output tracking in decentralized adaptive control systems," *IEEE Transactions on Automatic Control*, vol. 47, pp. 390–395, 2002.

[47] B. M. Mirkin, "A new decentralized model reference adaptive control scheme for large scale systems," in *Proceedings of the 4th IFAC Int. Symp. Adaptive Systems Control Signal Processing*, (Grenoble, France), 1992.

[48] J. A. Suykens, J. P. Vandewalle, and B. L. DeMoor, *Artificial Neural Networks for Modeling and Control of Non-Linear Systems*. New York: Springer-Verlag, 1995.

[49] A. S. Poznyak, E. N. Sanchez, and W. Yu, *Differential Neural Networks for Robust Nonlinear Control - Indentiifcation, State Estimation and Trajectory Tracking*. Singapore: World Scientific Publishing Co., 2001.

[50] F.-C. Chen and C.-C. Liu, "Adaptively controlling nonlinear continuous-time systems using multilayer neural networks," *IEEE Transactions on Automatic Control*, vol. 39, pp. 1306–1310, 1994.

[51] M. M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 41, pp. 447–451, 1996.

[52] G. A. Rovithakis, "Robustifying nonlinear systems using high order neural network controllers," *IEEE Transactions on Automatic Control*, vol. 44, pp. 104–107, 1999.

[53] G. A. Rovithakis, "Robust redesign of a neural network controller in the presence of unmodeled dynamics," *IEEE Transactions on Neural Networks*, vol. 15, pp. 1482–1490, 2004.

[54] C. Kwan and F. L. Lewis, "Robust backstepping control of nonlinear. systems using neural networks," *IEEE Trans. Syst., Man, Cybern. A*, vol. 30, pp. 753–765, 2000.

[55] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*. London: World Scientific, 1998.

[56] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*. Norwell, USA: Kluwer Academic, 2001.

[57] F. L. Lewis, J. Campos, and R. Selmic, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. Philadelphia: Society of Industrial and Applied Mathematics Press, 2002.

[58] S. S. Ge and C. Wang, "Adaptive NN control of uncertain nonlinear pure-feedback systems," *Automatica*, vol. 38, pp. 671–682, 2002.

[59] Y. Li, S. Qiang, X. Zhuang, and O. Kaynak, "Robust and adaptive backstepping control for nonlinear systems using rbf neural networks," *IEEE Transactions on Neural Networks*, vol. 15, pp. 693–701, 2004.

[60] D. Wang and J. Huang, "Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form," *IEEE Transactions on Neural Networks*, vol. 16, pp. 195–202, 2005.

[61] K. S. Narendra, Y.-H. Lin, and L. Valavani, "Stable adaptive controller design, part ii: Proof of stability," *IEEE Transactions on Automatic Control*, vol. 25, pp. 440–448, 1980.

[62] R. W. Brockett, "System theory on group manifolds and coset spaces," *SIAM J. Control*, vol. 10, pp. 265–284, 1972.

[63] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1975.

[64] E. D. Sontag, "Stability and stabilization disturbances: Discontinuities and the effect of disturbances," *Nonlinear Analysis, Differential Equation, and Control, F. H. Clarke and R. J. Stern, Editors, Montral*, 1998.

[65] S. Nikitin, *Global Controllability and Stabilization of Nonlinear Systems*. World Scientific, 1994.

[66] R. W. Brockett, "Asymptotic stability and feedback stabilization: in differential geometric control theory," *R. W. Brockett, R. S. Millman,H. J. Sussmann, Editors, Boston:Birkhauser*, pp. 181–191, 1983.

[67] A. Isidori, *Nonlinear Control Systems*. London: Springer, 1995.

[68] H. J. Sussmann, "Orbits of families of vector fields and integrability of distribution," *AMS Trans*, vol. 180, pp. 171–188, 1973.

[69] D. Aeyels, "Generic observability in differentiable systems," *SIAM Journal of Control and Optimization*, vol. 19, pp. 596–603, 1981.

[70] M. Nerurkar, "Observability and topological dynamics," *Journal of Dynamics and Differential Equations*, vol. 3, pp. 273–287, 1991.

[71] K. S. Narendra and S. J. Brown, "Neural networks in optimal control: Part I," *Technical Report 9703, Center for Systems Science, Yale University*, 1997.

[72] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches, D. A. White and D. A. Sofge, Editors, Van Nostrand Reinhold*, 1992.

[73] S. Ferrari and R. F. Stengel, "Model-based adaptive critic designs," *Learning and Approximate Dynamic Programming: Scaling Up to the Real World, J. Si, A. Barto, W. Powell, D. Wunsch, Eds., IEEE Press and John Wiley & Sons*, pp. 65–92, 2004.

[74] M. Agarwal, "A systematic classification of neural-network based control," *IEEE Control Systems Magazine*, vol. 17, pp. 75–93, 1997.

[75] G. Herrmann, S. S. Ge, and G. Guo, "Practical implementation of a neural network controller in a hard disk drive," *IEEE Transactions on Control Systems Technology*, vol. 13, pp. 146–154, 2005.

[76] P. He and S. Jagannathan, "Neuro-controller for reducing cyclic variation in lean combustion spark ignition," *Automatica*, vol. 41, pp. 1133–1142, 2005.

[77] M. Khalid, S. Omatu, and R. Yusof, "MIMO furnace control with neural networks," *IEEE Transactions on Control Systems Technology*, vol. 1, pp. 238–245, 1993.

[78] J. D. Bŏsković and K. S. Narendra, "Comparison of linear, nonlinear and neural-network-based adaptive controllers for a class of fed-batch fermentation processes," *Automatica*, vol. 31, pp. 814–840, 1995.

[79] L. A. Feldkamp, G. Puskorius, K. A. Marko, J. V. James, T. M. Feldkamp, and G. Jesion, "Unravelling dynamics with recurrent networks: Application to engine diagnostics," in *Proceedings of the Ninth Yale Workshop*, (New Haven, CT), pp. 59–64, 1996.

[80] D. Prokhorov and L. A. Feldkamp, "Bayesian regularization in extended Kalman filter training of neural networks," in *Proceedings of the Tenth Yale Workshop*, (New Haven, CT), pp. 77–84, 1998.

[81] L. A. Feldkamp, D. Prokhorov, and T. M. Feldkamp, "Conditioned adaptive behavior from a fixed neural network," in *Proceedings of the Eleventh Yale Workshop*, (New Haven, CT), pp. 78–85, 2001.

[82] L. A. Feldkamp and D. Prokhorov, "Recurrent neural networks for state estimation," in *Proceedings of the Twelfth Yale Workshop*, (New Haven, CT), pp. 17–22, 2003.

[83] L. A. Feldkamp and G. V. Puskorius, "A signal processing framework based on dynamic neural networks with application to problems in adaptation, filtering, and classification," *Proceedings of the IEEE*, vol. 12, pp. 651–659, 1998.

[84] A. Kun and T. Miller, "Adaptive dynamic balance of a biped robot using neural networks," in *Proceedings of the 1996 IEEE International Conference On Robotics and Automation*, (Minneapolis, MN), 1996.

[85] G. G. Yen, "Reconfigurable neural control in precision space structural platforms," *Neural Networks for Control, O. Omidvar and D.L. Elliott, Editors, Academic Press*, pp. 289–316, 1997.

[86] F. J. Doyle, M. A. Henson, B. A. Ogunnaike, J. S. Schwaber, and I. Rybak, "Neuronal modeling of the baroreceptor reflex with applications in process modeling and control," *Neural Networks for Control, O. Omidvar and D.L. Elliott, Editors, Academic Press*, pp. 89–130, 1997.

[87] A. Subramanian and D. Gerrity, *US Patents 6967594, 6650251 and US Patent Application 60840623.*