AN ABSTRACT OF THE DISSERTATION OF

Radosław Romuald Zakrzewski for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on December 21, 1994.

Title: Neural Network Control of Nonlinear Discrete Time Systems

Abstract approved: Redacted for Privacy

Ronald R. Mohler

The main focus of this work is on the problem of existence of nonlinear optimal controllers realizable by artificial neural networks. Theoretical justification, currently available for control applications of neural networks, is rather limited. For example, it is unclear which neural architectures are capable of performing which control tasks. This work addresses applicability of neural networks to the synthesis of approximately optimal state feedback. Discrete-time setting is considered, which brings extra regularity into the problem and simplifies mathematical analysis. Two classes of optimal control problems are studied: time-optimal control and optimal control with summable quality index. After appropriate relaxation of the optimization problem, the existence of a suboptimal feedback mapping is demonstrated in both cases. It is shown that such a feedback may be realized by a multilayered network with discontinuous neuron activation functions. For continuous networks, similar results are obtained, with the existence of suboptimal feedback demonstrated, except for a set of initial states of an arbitrarily small measure. The theory developed here provides basis for an attractive approach of the synthesis of near-optimal feedback using neural networks trained on optimal trajectories generated in open loop. Potential advantages of control based on neural networks are illustrated on application to stabilization of interconnected power systems. A nearly time-optimal controller is designed for a single-machine system using neural networks. The obtained controller is then utilized as an element of a hierarchical control architecture used for stabilization of a multimachine power transmission system. This example demonstrates applicability of neural control to complicated, nonlinear dynamic systems.

Neural Network Control of Nonlinear Discrete Time Systems

by

Radosław Romuald Zakrzewski

A Dissertation submitted to

Oregon State University

in partial fulfillment of

the requirements for the

degree of

Doctor of Philosophy

Completed December 21, 1994

Commencement June 1995

Doctor of Philosophy dissertation of Radosław Romuald Zakrzewski

presented on December 21, 1994

APPROVED:

Redacted for Privacy

Major Professor, representing Electrical and Computer Engineering

Redacted for Privacy

Head of Department of Electrical and Computer Engineering

Redacted for Privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Radosław Romuald Zakrzewski, Author

Acknowledgements

I would like to thank Professor Mohler for all his help throughout my time at Oregon State University, Professor Kołodziej for many valuable discussions and research ideas, and Professor Pacut, of Warsaw University of Technology, for introducing me to the fascinating subject of artificial neural networks. Special thanks go to fellow graduate students, Rajkumar Vedam and Wenchun Zhu, for their moral support and assistance. Most importantly, however, I want to thank my wife, Elżbieta, for her unending patience in dealing with a PhD in statu nascendi.

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

List of Figures

Neural Network Control of Nonlinear Discrete Time Systems

Chapter 1
Introduction

For the past few years, the so called artificial neural networks have been an increasingly popular design paradigm in virtually all fields of technology concerned with some forms of information processing, including control systems. Artificial neural networks, or in short neural networks, are technical models that originated from biological research on functioning of a human brain. Research in neural networks was initially motivated by the desire to understand the cognitive processes of human beings through mathematical analysis and simulation of artificial models of elements of the neural system. In addition to this, a second motivation came from was the attempt to imitate those abilities of a human brain that are still out of reach of even the most powerful computing devices available. Accordingly, there exist two separate, although often overlapping, branches of research in the neural network community, which are focused on biological and technical aspects respectively. The biologically oriented branch is mainly concerned with research of human, or more broadly speaking, animal brains. The primary goal is to model biological neurons and neural systems as accurately as possible, and the resulting artificial neural network is treated most often as a tool for analysis and understanding of the actual neural systems. Consequently, the information processing mechanisms used in the artificial models are required to be biologically plausible. The engineering branch, on the other hand, is mostly concerned with imitation of certain abilities of human brains, with not much attention given to accuracy of the artificial model with respect to the biological prototype. The main objective is to obtain devices capable of performing difficult tasks in an "intelligent" way, which does not necessarily mean the way humans would perform them. In this approach the actual neural systems are merely a motivation, an initial step. Thus the artificial neural networks are not required to display biologically plausible features, and in fact often use information-processing mechanisms impossible in living organisms. Nevertheless, the basic structure of networks used in the two approaches is the same, and can be most succinctly characterized as a network of very simple processing elements, neurons, connected to each other through transmission channels called synapses. The structure is distributed, that is simple computations are performed independently in each neuron and the resulting signals are then sent to other neurons. The computations are also usually local, in the sense that the neuron uses only information from its set of inputs which most often represents signals from only a portion of other neurons. Because of this interconnected structure the neural networks used in technical applications are often called connectionist models, to accentuate their being only crude models of biological neural systems. Accordingly, the whole field of research is sometimes called connectionism. Here however, the more popular term neural networks, or sometimes simply networks, will be used.

One of the most popular areas of applications of artificial neural networks has been the control of nonlinear dynamical systems. The reason for this is that despite great advances in control theory, there are still numerous control problems that cannot be satisfactorily solved using more conventional, analytical methods. These include systems for which no accurate analytical model is available, or the existing model is too complicated for use in the process of controller synthesis. The existing nonlinear control theory offers design techniques dependent upon availability of a detailed model of dynamics, which is required to satisfy many difficult to verify and often restrictive assumptions. On the other hand, linearization techniques give results that are local in nature and often not sufficient enough for systems with wide range of operating conditions. Thus, a need exists for novel and effective control schemes, and neural networks have been proposed as such a method. The observation, that humans are often able to learn to successfully perform control tasks which are too difficult for even the most sophisticated control algorithms, has been projected onto artificial models of human neural systems. This optimistic belief in superior abilities of artificial neural networks is most often unfounded, as they are in no respect accurate representations of actual brains. Nevertheless, artificial neural networks did prove to be an effective design paradigm for the synthesis of nonlinear controllers. They have been shown to possess certain universal approximation properties which allow their use as control algorithms for a large class of nonlinear dynamical systems. Still, the theory behind such applications is unsatisfactory and does not provide enough justification for many of the apparent successes. In this work, an attempt is made to contribute to the growing theory of neural control with respect to a class of optimal control problems. Also, a heuristic approach to the control of uncertain systems using neural network methodology will be presented and illustrated on a simple example.

In the remaining part of this chapter some historical background will be given on neural networks, the applicability to control systems will be discussed, and the focus of this dissertation will be explained in some detail.

1.1 Historical overview

The origins of the research on artificial networks may be traced to early 1940s, when the first quantitative studies on the behavior of human neurons were published. In [1], a mathematical model was constructed which described a neuron as a simple information processing unit, transforming a set of external stimuli into a single response. That model, formulated as a simple mathematical function, is still currently used as a foundation for a majority of neural network architectures. Its main significance lay in the implication that it should be possible to express functions of the brain, and in particular the phenomenon of intelligent thinking, by a set of mathematical equations. Subsequently, a learning law was suggested in [2], which explained how the neuron function might change according to a prevailing type of stimuli, or, as an engineer might say, current operating conditions. The proposed model of the learning process consisted of adjustment of the synaptic strength, or weight, associated with particular stimuli. The idea of learning through weight adaptation has been since utilized in all training algorithms for artificial neural networks. Demonstration that the phenomenon of learning can be exhibited by a properly built mathematical model was quite a revolutionary achievement. In particular, it implied that any object behaving according to that mathematical description would also appear to learn from, or adapt in an intelligent way to its environment. Thus, a possibility of building a thinking machine was presented. With the advent of the first electronic computers around that time, this possibility appeared very realistic, and it was not very long before such attempts were undertaken.

The actual beginning of artificial neural networks as a separate discipline can be dated to the 1950s, and to the perceptrons [3]. Originating from efforts to model and imitate functions of the human retina, those early neural networks were intended mainly for recognition and classification of visual patterns. Publication of the then celebrated perceptron learning algorithm and of mathematical proof of its convergence lent much credibility to the successful application of perceptrons to various classification problems. Positive coverage by the popular press, and general public belief in almost unlimited possibilities of scientific progress, added to widespread enthusiasm about the newly developed machines. Accordingly, expectations were raised very high and construction of truly thinking and intelligent machines seemed imminent.

The abrupt end to that first heroic period of neural network research came in late 1960s and is usually credited to publication of [4]. That work exposed serious theoretical limitations of applicability of perceptrons to more complicated problems. In particular, the class of classification problems, which the most widely used single-layer perceptrons were able to solve, was shown to be very small. Thus, even though the single layer perceptron was theoretically guaranteed to learn fast, it could not learn much. On the other hand, for multi layer perceptrons, which could solve a much larger class of classification problems, no learning algorithms were available. This argument, rather than to prompt further research into learning of multi layer networks, was perceived as a proof that the perceptron architecture was unfit for application to any real life problems, and, as it is generally believed, lead to almost complete termination of research funding, and consequently of research itself. In the revised version of that book, the authors denied their responsibility for single-handedly killing perceptrons, arguing instead that the catastrophe had been already imminent due to incompatibility between overblown expectations and actual limited capabilities and achievements. Whatever the actual reason, by the end of the 1960s research in the field had ceased almost completely.

The revival of neural network research was observed in the mid of 1980s, and again is largely attributed to a single book [5], and in particular to the famous backpropagation learning algorithm included therein. Through slight modification of the structure of the artificial neurons, the problem with training of multilayer networks was successfully overcome and the spectrum of prospective applications of artificial neural networks widened almost unboundedly. It must be remarked however, that the same algorithm had been already published at least twice [6], [7], so that its reintroduction in [5] was a marketing success, rather than a research breakthrough. Also other neural architectures were being introduced at that time, most notably recurrent networks of Hopfield [8], whose prestige was instrumental in bringing artificial neural networks from oblivion back into the mainstream science and engineering. Yet, since the majority of practical applications involve in some form the backpropagation method or one of its derivatives, it is customary to regard [5] as the beginning of a new era in neural network research.

A few recent years witnessed a very high level of activity in the field of artificial neural networks. It may be said that in almost every field of technology there have been, reportedly very successful, applications of neural networks. These include areas as diverse as aircraft control, power systems operation, earthquake prediction or financial market analysis. Even though occasionally unsubstantiated claims are made, and applications are attempted to problems easily solved by other well established methods, the general atmosphere is much more calm than 30 years ago. Today, hardly anybody perceives neural networks as a panaceum for all difficult problems, and it is realized that the dream of a thinking machine must be postponed until an unspecified future. Also the field experienced a considerable influx of respected researchers from other fields, who brought in more of the much needed scientific rigor. In particular, much attention was given to the analysis of mathematical properties of the networks, resulting in significant advances with respect to representation and approximation properties. Therefore there seems to be no immediate danger of the field collapsing again, with its hidden flaws suddenly exposed by some groundbreaking publication. Neural networks now constitute a more mature and well established discipline which displays no signs of imminent demise. It may be said that the wave of activity in neural networks has subsided a little since its peak a few years ago. This, however, indicates rather a transition from the phase of overenthusiastic applications to everything, to a period of more systematic and thorough research, when the true advantages of neural networks will be exploited where really applicable. It is predicted and hoped that in the years to come the networks will be more often seen for just what they are - a very convenient and flexible tool to the solution of some difficult engineering problems.

1.2 Neural networks in systems and control

The majority of applications of artificial neural networks in control systems are based on treating the network as a specific kind of nonlinear mapping, and then using it to approximate input-output behavior of some elements of the system in question. The situations arising may be roughly categorized into identification problems and controller synthesis problems. In the first case a neural network is used as a model of a given dynamical system, for which analytical modeling techniques are not applicable, or may not be accurate enough. This may be caused either by insufficient knowledge about the system nature, or unavailability of proper parameters which determine its dynamics, or both. In such a case the network is trained to imitate the behavior of the system using the input-output measurements as input data. The so-constructed model may later be used for the purpose of prediction of the real system future behavior given certain outputs, estimation of signals that are corrupted by noise or unavailable in practice, or for the synthesis of the controller that will later be used on the actual system. The controller synthesis using a neural network is considerably more difficult than the identification task, for the reason that the desired input-output behavior of the controller is not available for the network as the training data. Thus, the learning process must involve interaction of the controller with the system to be controlled, or with its accurate model. These issues are only signaled here in an informal manner, since an excellent exposition of the involved theoretical problems may be found in [9], [10], [11] or [12]. For numerous references to other works, the survey paper [13] or books [14], [15] can be consulted.

The theoretical foundation for applications of neural networks to control problems is provided by function approximation results, which will be reviewed in chapter 2. However, exactly how much justification that theory provides, remains largely unclear. A recent paper [16] highlighted deficiencies of the available theory, stating that the most widely used neural architecture with continuous neurons is generally not sufficient for the most fundamental task of stabilization of nonlinear dynamical systems. It was shown that since the stabilizing feedback mapping cannot be guaranteed to be continuous, the available theorems on approximation of continuous functions by multilayered networks are in fact of little use. A symptomatic response to those findings may be found in a paper [12] by a leading researcher in the field, where first the problem was acknowledged, and then it was stated that only systems which allow continuous feedback would be considered. Similarly in other papers it is simply postulated that an appropriate feedback policy may be realized by the neural architecture being considered, and with this assumption already made other problems are addressed. The author of this dissertation also used the same path, in the paper [17], on the approximation of optimal control by neural networks. There it was "tacitly assumed" that the control policy in question can indeed be approximated by a continuous function, and consequently by a neural network. Such approach may be acceptable in application studies, where it is usually very difficult to verify mathematical properties of the controlled system, and it is routinely assumed that a chosen design paradigm is applicable to the given problem. It is important however to be sure, that the class of systems, for which this assumption actually holds, is not empty. Thus, there is an urgent need to identify classes of dynamical systems and types of control tasks for which application of artificial neural networks in the capacity of controllers is theoretically justified.

1.3 Focus of this work

The main part of this dissertation addresses the above mentioned existence problem in the context of optimal control. The developments are carried in the discrete time setting, which corresponds to a majority of modern computer-based controller implementations. The considered class of dynamical systems corresponds to physical continuous-time systems under sampling. It is true that, if the sampling is fast enough, it is possible to consider the discrete time control as a sufficiently accurate approximation of continuous-time control, and consequently perform the analysis in the continuous time. It turns out, however, that the analysis in the discrete time setting is more convenient for the problems studied here.

The controller design approach being investigated is an approximation of optimal control policies by neural feedback controllers. The general idea is that the solutions of the easier task of open-loop optimization be used as training data for a neural network to learn the underlying optimal control law. This approach has already been successfully applied, however with no supporting theory available. The question addressed here is whether, and when, it is indeed feasible to approximate an optimal control policy by a neural network controller. The controller structure considered is that of the state feedback. The problem is studied for the two most common cases of optimal control: time optimal control, which is investigated in chapter 3, and a generalization of quadratic optimal control considered in chapter 4. With fairly general assumptions, existence of near optimal controllers is demonstrated in a class of multilayered networks, which realize discontinuous mappings. Somewhat weaker approximation results are also established for controllers realizable by multilayered continuous networks. These results are obtained using the already standard theory on function approximation by neural networks, which is reviewed in chapter 2. All the neural architectures discussed here are also defined therein.

The last chapter of the dissertation presents a heuristic neural network based controller architecture, for which no relevant theory has been yet established. The approach is illustrated using an example of a complicated control problem related to stabilization of interconnected electric power transmission systems. The assumed controller structure, based on physical features of the modeled system, is that of output, rather than state feedback. While lacking theoretical justification for its applicability, the proposed controller is demonstrated to offer near optimal performance. This dichotomy of the dissertation in a way epitomizes the current condition of neural network control, where most of the successful practical applications lie outside the scope of available theory. Thus, the urgent need for closing this gap is further emphasized.

Chapter 2
Feedforward networks

The neural network architecture considered here are multilayered feedforward networks, also known as multilayered perceptrons. Even though numerous other neural network structures have been proposed, feedforward networks remain the most widely used paradigm. In control applications in particular, it may be claimed that the overwhelming majority of applications of neural networks involves multilayered perceptrons used with some version of supervised learning. The architecture is based on the original perceptrons [3], which were developed as an attempt to model functions of the human retina. The intended application of the perceptrons was classification of visual patterns into a finite number of prespecified categories, and consequently the output of the network was of binary nature. The same basic structure remains in use today, with the only substantial modification being the replacement of discontinuous binary valued activation functions with smooth ones. Thus the output signals change continuously with the inputs, and the feedforward network is essentially a particular form of heavily overparametrized nonlinear finite-dimensional mapping. Consequently, most questions about such networks can be posed as standard problems of real analysis, approximation theory or optimization theory, and the whole field may be viewed as a subdivision of the theory of real functions.

What sets neural networks apart from other families of nonlinear functions is their modular and layered structure. A network consists of a large number of very simple, one-dimensional processing elements, called neurons, that differ only by values of adjustable parameters. The data processing is distributed, in the sense that in each neuron only simple calculations are performed and the result is propagated to the next layer of neurons. Furthermore, calculation of the derivatives of the mapping with respect to both independent variables and parameters can be performed in a similar layered and distributed manner, thus allowing for convenient software and hardware implementations. These appealing structural features, combined with the biological foundations of the architecture, caused widespread interest in multilayered perceptrons and gave rise to numerous applications. These in turn prompted many specific questions not always sufficiently well addressed in standard theory. Therefore, it is justified to devote special attention to feedforward networks, and to study them as a separate class of functions, with specific applications in mind.

The popularity of this particular neural network paradigm may be partly ascribed to theoretical approximation results available for this class of networks. It has been demonstrated that finite-dimensional functions realizable by feedforward networks are dense in several important function spaces, and thus can be used to approximate arbitrarily closely unknown nonlinear mappings. Thus neural networks may be applied to the modeling of nonlinear dynamical systems if models based on physical principles are inadequate or not available. Similarly, nonlinear controllers may be realized in the form of a neural network. The approximation results obtained for multilayered percetrons are only of an existential nature, and do not specify how to obtain the proper network (that is its size and values of the parameters) if the mapping to be approximated is not given in analytic form. Nevertheless those results lend at least some theoretical justification to the applications of multilayered networks, thus adding to their popularity.

The other reason behind the widespread use of multilayered perceptrons is the availability of training procedures allowing adjustment of the network parameters so that the obtained nonlinear mapping satisfies postulated criteria. While other neural paradigms offer unsupervised self-organizing learning algorithms, feedforward networks are used with supervised, error-driven training, best suited for applications where it is possible to specify the desired response of the network or at least a numerical measure of its performance. Virtually all training methods are modifications or enhancements of the widely publicized backpropagation algorithm, whose publication in [5] revolutionized neural network research. In retrospection it may be difficult to comprehend how such a straightforward development could have been a turning point in the history of the field. Backpropagation is a method of computing the sensitivities of the network output with respect to its parameters, based on the chain rule of calculating derivatives of compositions of functions, and the associated training method is a version of steepest descent optimization. Before backpropagation was finally popularized, the Rosenblatt's training algorithm which was in use, allowed training only of single-layer perceptrons, known to be sufficient only for so-called linearly separable classification problems [4]. The introduction of smooth neurons and of backpropagation training, suitable for multilayered networks, allowed the use of perceptrons in cases that are not linearly separable, which constitute an overwhelming majority of interesting classification problems. This provided the momentum that finally made neural networks one of the most fashionable areas of engineering research.

The remaining portion of this chapter includes the review of selected topics on feedforward networks, that may be useful in understanding the subsequent developments. First, the architecture will be described in detail. Then the available results on function approximation by neural networks will be reviewed. Finally training procedures will be discussed. All the material is standard and can be found in any introductory text on neural networks. Its inclusion here is intended only to provide a quick reference for the reader not acquainted with neural networks, with no claims to completeness being made.

2.1 Feedforward network architecture

A feedforward neural network is a nonlinear finite-dimensional mapping which can be interpreted as a composition of a number of simpler mappings, layers, each in turn consisting of several real-valued functions, neurons. All neurons in one layer are usually identical, except for values of parameters, which are called connection weights, or, using biologically motivated term, synaptic strengths. During the network operation the weights may be changed, or adapted, by a learning algorithm, resulting in changes in the nature of the mapping from the external inputs to the network outputs. Depending on the manner in which the inputs and the parameters are treated, several separate, but closely related, meanings may be associated with the term network. First, the network may be understood as the following joint function of both inputs and parameters

$$F : \Re^{N_i} \times \Re^{N_w} \to \Re^{N_o}, \quad y = F(x, w) \tag{2.1}$$

where $x \in \Re^{N_i}$ is the external input, $w \in \Re^{N_w}$ is the parameter vector, and $y \in \Re^{N_o}$ is the network output. Next, the network may mean the particular mapping obtained with specific value of parameter vector $w = w^\star$, so that

$$F_{w^\star} : \Re^{N_i} \to \Re^{N_o}, \quad y = F_{w^\star}(x) = \phi(x, w^\star) \tag{2.2}$$

Additionally, for a network F in form (2.1), it is possible to define a mapping $\tilde{F}$ associating, with every value of the parameter vector $w = w^\star$, a particular mapping $F_{w^\star}$ defined by

$$F_{w^\star}(x) = (\tilde{F}(w^\star))(x) = F(x, w^\star) \tag{2.3}$$

For each of these meanings the term feedforward network can be used, depending on the circumstances. When the approximation results are discussed, the network approximating a given function will be understood in sense (2.2) - that is with both structure and a particular parameter vector fixed. The training process may be viewed as an iterative search for the optimal value of the parameter vector $\hat{w}$, for which the obtained network $\tilde{F}(\hat{w})$ results in the best value of a performance criterion. Then the term network may be used to describe the association between parameter values and types of input-output behavior, or a mapping in the form (2.3). If the training of the net is performed on-line in the closed-loop control system, then inputs and parameters of the net exhibit mutual dependence and have to be treated identically, thus the network will mean the mapping of the form (2.1). Which particular meaning is used here, will be usually clear from the context, and will be explicitly stated should any confusion arise.

2.1.1 Artificial neuron

The common feature of all artificial neural network architectures is that they are composed of a large number of simple processing elements called neurons. The structure of a typical artificial neuron is patterned after a model of a human neuron initially introduced in [1]. Even though biological neurons have long been known to exhibit much more complicated behavior than such a model, the latter is still widely used in artificial neural networks. Its simplicity and relatively easy implementation make it an appealing choice in those technical applications which do not aim at exact modeling and analysis of functioning of human brain. The basic neuron is a multi-input, single-output function $g : \Re^N \rightarrow \Re$, parametrized by $N+1$ real numbers, w_i, $i = 1, \ldots, N$ and θ

$$g(x) = \Psi(\sum_{i=1}^{N} w_i x_i \ + \ \theta) \tag{2.4}$$

Parameters w_i represent weights, or synaptic strengths, associated with each of the incoming signals x_i. The resulting weighted sum of the inputs x_i is the total activation of the neuron caused by the external signals. The parameter θ is called a bias or a threshold and represents the critical level of the activation at which the neuron changes its response. It is common practice to modify the above model of the neurons, so that all the parameters w_i are treated uniformly as weights associated with the input signals. This is done by augmenting the input vector x and the weight vector w with an artificial unity bias signal and with threshold θ, respectively, as follows

$$\tilde{x} = [x_1, \ldots, x_N, 1]^T$$

$$\tilde{w} = [w_1, \ldots, w_N, \theta]^T$$

With this notation the model of an artificial neuron takes the following form

$$g(\tilde{x}) = \Psi(\sum_{i=1}^{\tilde{N}} \tilde{w}_i \tilde{x}_i) \tag{2.5}$$

with $\tilde{N} = N + 1$. This simplified neuron model, with the threshold included in the weight vector, is often more convenient to use. For example the roles of inputs and weight are symmetric, which allows for use of the same formulae in the calculation of sensitivities with respect to inputs and with respect to weights. On the other hand, it requires inclusion of additional neurons with constant outputs in each layer of the network, and care must be taken to distinguish between its actual and augmented input. As the two notations are equivalent, both will be used here, depending on which is more convenient.

The function $\Psi : \Re \rightarrow \Re$ is commonly called a transfer function, as it defines the manner in which total activation of the neuron is transformed into its output signal. Since the term transfer function is reserved in control theory for input-output representation of a linear dynamical system, function Ψ is called here the activation function of the neuron. The original activation function used in early neural networks was the Heaviside, or step function

$$\Psi(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{2.6}$$

In this case the overall function computed by the neuron is the characteristic function of the halfspace determined by vector w, and the threshold parameter θ defines the critical level of activation signal necessary for the neuron to produce positive output. Artificial neurons with binary outputs approximate the functioning of human neurons, which either "fire" or not. The basic application of networks using this type of neurons was initially for the classification of input patterns (that is values of the input vector) into separate and well defined classes. Their main disadvantage lies in the discontinuity of the activation function Ψ, which prevents use of gradient based training methods. To overcome this, smooth approximations of the step function have been introduced. The most often used activation function is the logistic function

$$\Psi(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

or its scaled equivalent

$$\Psi(x) = \tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{2.8}$$

Both functions are bounded, strictly increasing, and the threshold parameter θ of the neuron (2.5) determines the activation level for which the output signal is halfway between its maximum and minimum value. The common name used for functions similar to (2.6), (2.7) and (2.8) is sigmoidal or squashing function. A function $\Psi : \Re \rightarrow \Re$ is called sigmoid, if is nondecreasing, and satisfies $\lim_{x \rightarrow -\infty} \Psi(x) > -\infty$ and $\lim_{x \rightarrow +\infty} \Psi(x) < +\infty$. A sigmoid function is not required to be continuous, but the definition allows at most countably many discontinuities. An often used case of a non-sigmoid activation function is identity

$$\Psi(x) = x \tag{2.9}$$

resulting in the linear processing characteristics of the neuron (2.4). Linear neurons are often used in the last, output layer to allow arbitrary range of the mapping realized by the network.

2.1.2 A single layer

A layer of the network is a collection of M neurons, g_i, $i = 1, \ldots, M$, each with the same activation function Ψ and with the same input vector $x \in \Re^N$. The layer may also be defined as a finite-dimensional mapping $G : \Re^N \rightarrow \Re^M$, whose i-th coordinate function G_i is realized by the i-th neuron g_i, that is

$$y_i = g_i(x) = \Psi(\sum_{j=1}^{N} w_{i,j} x_j \; + \theta_i) \tag{2.10}$$

where y_i is the i-th element of the output vector $y = [y_1, \ldots, y_M]^T$, and the parameters $w_{i,j} \in \Re$, $j = 1, \ldots, N$, $\theta_i \in \Re$ are the weights and threshold associated with i-th neuron. A convenient notation is obtained, if the argument x is augmented with the bias term, to form $\tilde{x} \in \Re^{N+1}$ as in (2.5). Let $W \in \Re^{M \times (N+1)}$ denote the matrix composed of weights $w_{i,j}$, and define a nonlinear function $\Psi : \Re^M \rightarrow \Re^M$, whose coordinate functions Ψ_i, $i = 1, \ldots M$ are neuron activation functions Ψ acting on the i-th coordinate of the argument

$$\Psi_i(x) = \Psi(x_i) \tag{2.11}$$

Then the mapping realized by the layer may be represented as a composition of linear transformation W with a simple nonlinear mapping Ψ acting on each coordinate of $W\tilde{X}$ separately.

$$G(x) = \Psi(W\tilde{x}) \tag{2.12}$$

2.1.3 Multilayered network

A feedforward network is a collection of L layers $G^{(l)} : \Re^{N_l} \rightarrow \Re^{M_l}$, $l = 1, \ldots, L$, such that $M_l = N_{l+1}$, so that all outputs of the l-th layer act as inputs to the $l+1$-th layer. The resulting mapping $F : \Re^N \rightarrow \Re^M$, $N = N_1$, $M = M_L$, is defined

$$F(x) = \underbrace{G^{(L)}(G^{(L-1)}(\ldots(G^{(1)}}_{L\ times}(x))\ldots)) \tag{2.13}$$

With parameters of the l-th layer denoted as $w_{i,j}^{(l)}$ and $\theta_i^{(l)}$, and activation functions $\Psi^{(l)}$, the i-th coordinate function F_i is defined

$$F_i(x) = \Psi^{(L)}(\theta_i^{(L} + \sum_{i_L=1}^{N_L} w_{i,i_L}^{(L)}(\ldots\Psi(\theta_{i_2}^{(1)} + \sum_{i_1=1}^{N_1} w_{i_2,i_1}x_{i_1})\ldots)) \tag{2.14}$$

A simpler notation is possible if instead of using bias terms, each layer is augmented with a fictitious neuron, with corresponding weights and threshold such that its output is nonzero and constant (that is independent from the outputs of the preceding layer). Then the bias term θ may again be incorporated in the neuron weights and the network is represented as a cascade of linear and decomposed nonlinear functions

$$F(x) = \underbrace{\Psi^{(L)}(W^{(L)}(\Psi^{(L-1)}(W^{(L-1)}(\ldots\Psi^{(1)}(W^{(1)}}_{L\ times}(\tilde{x}))\ldots)))) \tag{2.15}$$

with $W^{(l)}$ being the linear mapping defined by the weight matrix associated with neurons of the l-th layer.

Layers $G^{(i)}$ corresponding to $1 \leq i < L$ are customarily called hidden layers, as values of their outputs are not "visible" directly outside the network, and they affect the output values indirectly. Also the number of neurons in a hidden layer is not affected by the input-output structure of the network, that is dimensionality of input and output vectors. Layer $G^{(L)}$ is called the output layer and its size is determined by dimensionality of the required output signal. In the majority of applications activation functions of all layers, except the output one, are identical, that is $\Psi^{(1)} = \Psi^{(2)} = \ldots = \Psi^{(L-1)}$. The output layer is often a linear transformation, that is $\Psi^{(L)}(x) = x$.

2.2 Approximation properties

The main area of applications of feedforward neural networks is the approximation of arbitrary nonlinear functions. That is, given some finite dimensional function $f : \Re^N \to \Re^M$, a neural model $F : \Re^N \to \Re^M$ will be used such that f and F are close in an appropriately defined sense. Usually, the approximated function f is not given explicitly, but rather is available in the form of pairs $(x, f(x))$ describing its behavior. For example f may represent the dynamics of a system whose model is identified from input and output data using a neural network F. In another case function f may be a priori unknown, but it might be guaranteed that there exists a control policy, which is synthesized by a neural network. In both situations it is of paramount importance to ensure that approximation using neural networks is suitable for the given problem. Part of the success of neural networks as a modeling tool follows from the findings that indeed very large classes of finite-dimensional functions can be approximated arbitrarily closely by feedforward networks. This means that by choosing an appropriately large neural network structure, that is with sufficiently many hidden neurons, it is possible to find parameter values such that the resulting neural network approximates the given function in some specific sense.

This section contains a brief review of some of the available theoretical results concerned with the approximation by neural networks. All the results cited here are of existential nature only. The statement that there exists an appropriate L

layered network with activation functions $\Psi^{(1)}, \ldots, \Psi^{(L)}$, such that some property is satisfied, will be understood in the sense that there exist integers $M_1, \ldots, M_{L-1}$, the sizes of hidden layers, weights $w_{i,j}^{(l)}$, and thresholds $\theta_i^{(l)}$, $l = 1, \ldots, L$, $i = 1, \ldots, M_l$, $j = 1, \ldots, N_l$, such that the resulting function F defined by (2.14) satisfies the property in question. The problem how to find the appropriate weight vectors is not addressed here.

The approximation theorems reviewed below are concerned with various types of feedforward networks corresponding to different activation functions Ψ of hidden layers of the network. Generally, better and more specific approximation properties are obtained if more restrictive assumptions are posed for Ψ, whose properties should correspond to properties of the class of functions being approximated. In particular the most popular networks with smooth sigmoidal neurons cannot guarantee satisfactory approximation of discontinuous functions, which arise in nonlinear controller analysis.

2.2.1 Approximation of continuous functions

Results on uniform approximation of continuous functions by feedforward networks were proved independently in [18], [19], [20], using slightly different assumptions on functions Ψ. The architecture that corresponds to majority of applications is a network with linear output neurons, $\Psi^{(L)}(x) = x$, and with identical activation functions of hidden layer neurons, $\Psi^{(1)} = \Psi^{(2)} = \ldots = \Psi^{(L-1)} = \Psi_h$, which are sigmoidal, or squashing. For this network structure the following approximation theorem holds.

Theorem 2.1 *Let $\Psi_h : \Re \to \Re$ be a sigmoidal function, C a compact set $C \in \Re^N$, and $f : \Re^N \to \Re^M$ a function that is continuous on C. Then, for any number $\varepsilon > 0$ and any integer $L \geq 2$ there exists a feedforward network $F : \Re^N \to \Re^M$ of the form (2.14) with L layers, hidden activation functions Ψ_h and linear output activation* *functions such that it approximates the given function uniformly on C with error smaller than ε, that is*

$$\sup_{x \in C} \| F(x) - f(x) \| < \varepsilon$$

Proof: Theorem 2.4, [19].

The above theorem yields the argument that any continuous function can be approximated by a neural network of the postulated form, so that the upper bound of approximation error may be made arbitrarily small. The requirement, that this should be possible only on compact sets, is not very restrictive, since in most physical applications signals of interest are bounded. The condition that the hidden activation function Ψ_h be sigmoidal may be replaced by other requirements. For example in [21] it is shown that theorem 2.1 holds if Ψ_h is continuous, absolutely integrable on $\Re$, that is $\int \mid \Psi(x) \mid dx < \infty$, and such that $\int \Psi(x)dx \neq 0$. Note that theorem 2.1 applies to networks with typical activation functions (2.6), (2.7) and (2.8). The above results have been stated for linear output activation function $\Psi^{(L)}$ in order not to restrict the range of the approximated function. If the function Ψ_L is continuous and strictly increasing (as for example functions (2.7) and (2.8)), then the above result also holds provided that the range of each coordinate function f_i is contained in the range of $\Psi(L)$.

Feedforward networks are often applied as models of unknown nonlinear mappings, and often sensitivity of the approximated function is analyzed using the neural model. This kind of application requires that not only the values of the function can be approximated, but also its derivatives. Appropriate theorems relevant to this problem are available for sufficiently smooth functions and networks. Let partial derivative of a function $g : \Re^N \rightarrow \Re$ associated with multiindex $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ be denoted as

$$D^\alpha g(x) = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \ldots \partial x_N^{\alpha_N}}(x)$$

where $\mid \alpha \mid = \sum_{i=1}^{N} \alpha_i$, and α_i are nonnegative integers. Let f_i denote the i-th coordinate function of $f : \Re^N \rightarrow \Re^M$. Then the following theorem holds.

Theorem 2.2 *Let $\Psi_h : \Re \rightarrow \Re$ be a function with continuous derivatives $D^i\Psi_h$ of order $i \leq K$, such that its K-th derivative is absolutely integrable, that is $\int \mid D^K\Psi_h \mid dx < \infty$. Let $C \subset \Re^N$ be a compact set, and $f : \Re^N \rightarrow \Re^M$ a function whose coordinate functions, $f_i : \Re^N \rightarrow \Re$, all have partial derivatives of all orders continuous and quickly vanishing, that is $\lim_{x\rightarrow\infty} x^\beta D^\alpha f_i(x) = 0$ for all possible multiindices α and β. Then, for any number $\varepsilon > 0$ and any integer $L \geq 2$ there exists a feedforward network $F : \Re^N \rightarrow \Re^M$ of the form (2.14) with L layers, hidden activation functions Ψ_h and linear output activation functions $\Psi^{(1)} = \Psi^{(2)} = \ldots = \Psi^{(L-1)} = \Psi_h$, such that it approximates the given function and all its partial derivatives of order less than or equal to K uniformly on C with error smaller than ε, that is*

$$\max_{1\leq i\leq N} \max_{|\alpha|\leq K} \sup_{x\in C} \mid D^\alpha F_i(x) - D^\alpha f_i(x) \mid < \varepsilon$$

Proof: Corollary 3.5, [22].

It may be noted that the standard smooth sigmoidal activation functions (2.7) and (2.8) satisfy the requirements of this theorem, and the corresponding networks may be used to simultaneously approximate both nonlinear functions and their derivatives.

2.2.2 Approximation of discontinuous functions

Nonlinear mappings occurring in some control problems may sometimes fail to be continuous. Unfortunately, the theoretical results on approximation of such functions by neural networks are considerably weaker than those reviewed in section 2.2.1. For general measurable functions the technique based on Lusin's theorem is used to provide uniform approximation everywhere, but on a set of arbitrarily small measure. The general statement is as follows.

Theorem 2.3 *Let $\Psi_h : \Re \rightarrow \Re$ be a sigmoidal function, μ a finite measure on the sigma-field of Borel measurable subsets of $\Re^N$, and $f : \Re^N \rightarrow \Re^M$ a Borel measurable* *function. Then, for any numbers $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, and for any integer $L \geq 2$, there exists a feedforward network $F : \Re^N \to \Re^M$ of the form (2.14) with L layers, hidden activation functions Ψ_h and linear output activation functions $\Psi^{(1)} = \Psi^{(2)} = \ldots = \Psi^{(L-1)} = \Psi_h$, such that it approximates the given function uniformly with error smaller than ε_1, except on a set of measure smaller than ε_2, that is*

$$\mu(\{x \in \Re^N : \| F(x) - f(x) \| > \varepsilon_1\}) < \varepsilon_2$$

Proof: Theorem 2.4, [19].

In particular, measure μ may be the Lebesgue measure restricted to some compact set $C \subset \Re^N$, for which the theorem implies that uniform approximation may be obtained on some $\tilde{C} \subset C$ with $\mu(\tilde{C}) < \varepsilon_2$. Unfortunately, the nature of Lusin's theorem invoked in the proof does not allow infering of anything about the location of the small excluded set $C \setminus \tilde{C}$, which depends on the particular function f being approximated.

If uniform approximation is not required and may be replaced by approximation in $\mathcal{L}^p$ sense, then any measurable function may be approximated by a feedforward network.

Theorem 2.4 *Let $\Psi_h : \Re \to \Re$ be a sigmoidal function, μ a finite measure on the sigma-field of Borel measurable subsets of $\Re^N$, with compact support and $f : \Re^N \to \Re^M$ a Borel measurable function. Then, for any number $\varepsilon > 0$ and for any integers $L \geq 2$, $p \geq 1$, there exists a feedforward network $F : \Re^N \to \Re^M$ of the form (2.14) with L layers, hidden activation functions Ψ_h and linear output activation functions $\Psi^{(1)} = \Psi^{(2)} = \ldots = \Psi^{(L-1)} = \Psi_h$, such that it approximates the given function in $\mathcal{L}^p$ sense with error less than ε, that is*

$$\left(\int_{\Re^N} \| F(x) - f(x) \|^p \, d\mu\right)^{\frac{1}{p}} < \varepsilon$$

Proof: Corollary 2.2, [19].

2.2.3 Selection of set valued mappings

In the control system context, the problem of approximation of an a priori unknown mapping may be posed in a slightly different way. Instead of specifying a single value of the unknown function, which should be approximated by the output of the neural network, it is possible to specify a set of admissible values such that any mapping with its value in this set represents an acceptable approximation. In this formulation the graph of approximating neural network is required to be contained in some desired set.

Consider a set $\Omega \subset \Re^N \times \Re^M$, and denote its projection on the first N coordinates as

$$\pi_N(\Omega) = \{x \in \Re^N : \exists y \in \Re^M \quad (x, y) \in \Omega\} \tag{2.16}$$

With this notation the following property holds for feedforward networks with Heaviside or step hidden activation functions (2.6).

Theorem 2.5 *Let $\Psi_h : \Re \rightarrow \Re$ be the Heaviside function (2.6). Suppose that $\Omega \subset \Re^N \times \Re^M$ is an open set, and let compact set C be contained in its projection on the first N coordinates, $C \subset \pi_N(\Omega)$. Then for any $L \geq 3$ there exists a feedforward network $F : \Re^N \rightarrow \Re^M$ of the form (2.14) with L layers, hidden activation functions Ψ_h and linear output activation functions such that if $x \in C$, then $(x, F(x)) \in \Omega$. If, additionally, $C_0 \subset C$ is a compact set such that $(x, 0) \in \Omega$ if $x \in C_0$, then the neural network may be chosen such that $F(x) = 0$ for $x \in C_0$.*

Proof: Proposition 2.4 [16].

It is interesting to note that no family of continuous functions possesses this property, thus it cannot be satisfied for neural networks with continuous activation functions. In the developments of the subsequent chapters, this theorem will be used for construction of proper state feedback mapping. The set Ω will be formed as a collection of state-control pairs defining the desired control policy. For each point x

in the interesting subset of the state space, a corresponding set $\Phi(x)$ of appropriate control actions will be defined. Thus, a set-valued control policy will be defined on an open subset $B = \pi_N(\Omega)$ of the state space, and will be sought in such a form that the sets $\Phi(x)$ are open. It will also be required that the mapping Φ is lower semicontinuous; that is, if $x \in \pi_N(\Omega)$ and $u \in \Phi(x)$, then for any open neighborhood U of u there exists an open neighborhood V of x, such that for any $\tilde{x} \in V$ it follows that $\Phi(\tilde{x}) \cap U \neq \emptyset$. If the set-valued feedback policy Φ satisfies these requirements, it will follow that the set Ω formed as a collection of admissible state-control pairs

$$\Omega = \{(x, u) : \ u \in \Phi(x)\} \tag{2.17}$$

is open, and theorem 2.5 guarantees the existence of network F such that for all state points of interest $F(x) \in \Phi(x)$. Such a single-valued function F is called a selection of the set-valued mapping Φ, since it selects one of possible elements of $\Phi(x)$. Thus theorem 2.5 may be rephrased as a result on neural network selections of lower semicontinuous set valued mappings.

The approach of forming a set-valued feedback control policy will be central to a large portion of developments of chapters 3 and 4. The fact that a selection cannot generally be guaranteed to be continuous, will imply that it will not be possible to construct a desired feedback policy using neural networks with smooth hidden activation functions, and utilization of Heaviside networks will be necessary. However, as it will be discussed in the next section, for the network synthesis or training task to be practically feasible, the activation functions of all neurons should be at least differentiable. Therefore construction of the continuous selection of the set-valued control mapping will be desirable. Unfortunately attempts to perform such a construction were not successful. The only theoretical result on continuous selections, that the author is aware of, is Michael's theorem which requires that the images $\Phi(x)$ (here the sets of admissible control values) be convex. This requirement seemed impossible to satisfy. Instead a properly defined approximate selection of $\Phi(x)$ by a continuous function will be investigated.

2.3 Training process

To utilize the approximation results of the previous section it is necessary to find appropriate parameter vectors $w_{i,j}^{(l)}$, such that the corresponding mapping is close to the desired one. This is usually done in an iterative process of supervised learning where the parameters of the network are adapted in order to decrease the approximation error. Even though uniform approximation results discussed in section 2.2 are widely quoted as a justification for approximation of neural networks to function approximation, the actual training of networks is most often based on an $\mathcal{L}^2$ type of error criterion. A usual case corresponds to the training data consisting of a finite number of pairs $(x^i, \hat{y}^i)$, where $\hat{y}^i = f(x^i)$ are the desired responses of the network to inputs x^i. Then the quadratic error function is defined

$$J = \frac{1}{2}\sum_{i=1}^{K}(\hat{y}^i - y^i)^T(\hat{y}^i - y^i) \tag{2.18}$$

where $y^i = F(x^i)$ are the actual responses of the network. If the data points x^i are drawn uniformly from some compact set $C \subset \Re^N$ of interest, then this error function is an approximation of the squared $\mathcal{L}^2$ distance between functions f and F calculated over set C. With fixed training data, the error J can be expressed as a function of the network parameters

$$J(w) = \frac{1}{2}\sum_{i=1}^{K}(\hat{y}^i - F(x^i, w))^T(\hat{y}^i - F(x^i, w)) \tag{2.19}$$

where $w \in \Re^K$, $K = \sum_{i=1}^{L}(N_i + 1)M_i$, is the combined vector of weights and thresholds $w_{i,j}^{(l)}$, $\theta_i^{(l)}$ Then the objective of the training process is to find

$$\min_{w \in \Re^K} J(w) \tag{2.20}$$

with respect to the weight vector w.

If all the neuron activation functions of the network are differentiable, then the problem of minimization of the error function can be approached using any of standard gradient optimization techniques. Calculation of gradient $\nabla_w J$ of the quality criterion (2.19) requires sensitivities of the networks output with respect to the weights. This may be achieved through the so called backpropagation method. Essentially it is a straightforward application of the chain rule to calculate the gradient of a composition of many functions. Consider the neural network in notation (2.15) and introduce the intermediate signals

$$z^l = W^{(l)}(\Psi^{(l-1)}(W^{(l-1)}(\dots\Psi^{(1)}(W^{(1)}(x))\dots))) \tag{2.21}$$

$$x^l = \Psi^{(l)}(W^{(l)}(\Psi^{(l-1)}(\dots\Psi^{(1)}(W^{(1)}(x))\dots))) \tag{2.22}$$

so that the function realized by the network can be defined recursively as

$$x^l = \Psi^l(z^{(l)}), \quad z^{(l)} = W^{(l)}(x^{(l-1)}) \tag{2.23}$$

with $x^{(0)} = x$ and $x^{(L)} = y = F(x)$. Consider the sensitivity of the output vector with respect to some particular weight $w_{i,j}^{(l)}$, which can be written

$$\left(\frac{\partial y}{\partial w_{i,j}^{(l)}}\right)^T = \left(\frac{\partial z^{(l)}}{\partial w_{i,j}^{(l)}}\right)^T \left(\frac{\partial y}{\partial z^{(l)}}\right)^T \tag{2.24}$$

The first term on the right hand side is just a vector with the only nonzero j-th entry equal to $x_j^{(l-1)}$, and the second term may be expressed as

$$\left(\frac{\partial y}{\partial z^{(l)}}\right)^T = \left(\frac{\partial \Psi^{(l)}}{\partial z^{(l)}}\right)^T (W^{(l+1)})T \left(\frac{\partial \Psi^{(l+1)}}{\partial z^{(l+1)}}\right)^T \dots (W^L)^T \left(\frac{\partial \Psi^{(L)}}{\partial z^{(L)}}\right)^T \tag{2.25}$$

Thus calculation of the sensitivities is seen as a composition of a number of linear operations, whose coefficients depend on particular value of the input signal x. Observe that the equation (2.25) may be rewritten into a recursive formula

$$\left(\frac{\partial y}{\partial z^{(l)}}\right)^T = \left(\frac{\partial \Psi^{(l)}}{\partial z^{(l)}}\right)^T (W^{(l+1)})T \left(\frac{\partial y}{\partial z^{(l+1)}}\right)^T \tag{2.26}$$

with

$$\left(\frac{\partial y}{\partial z^{(L)}}\right)^T = \left(\frac{\partial \Psi^{(L)}}{\partial z^{(L)}}\right)^T \tag{2.27}$$

This may be visualized as the propagation of sensitivity signals through the neural network back from its output (the L-th layer) to its input - hence the name backpropagation.

With the gradient of the quality criterion already calculated it is possible to use any minimization approach to find the optimal weight vector. The most popular is the gradient descent method, known also as the backpropagation training algorithm, for the reason that the direction of weight change is given directly by the minus gradient, obtained in turn from the backpropagation method. Numerous specialized training algorithms have been developed utilizing special features of the error function, which result from the particular form of function F realized by the neural network. Examples of such methods may be found in [23], [24], [25]. On the other hand standard gradient-based optimization algorithms may be utilized, such as those reviewed in [26]. In this work the conjugate gradient method was used.

The training problem becomes considerably more difficult, if not untractable, for the networks with discontinuous activation functions. In particular, in the case of Heaviside neurons, the function F realized by the network, and consequently the quality criterion J, are piecewise constant, and no information may be inferred from the local behavior about the desired direction of change of the weight vector. No multilayer equivalent of the perceptron training algorithm has yet been developed and the only viable option is that of random search techniques, which however offer usually slow convergence, particularly in view of the large dimensionality of the independent variable w. This is a very unfortunate conclusion in view of previous remarks on superior approximation and selection properties of Heaviside networks, and on insufficiency of continuous networks to various control problems. Until an efficient training algorithm for such networks is devised, the possibility of any practical application seems rather unlikely.

The objective of this short overview of problems related to training, as well as of the whole chapter, was to provide background necessary to follow the developments of this dissertation. Thus only issues directly related to this work has been signaled, and several important topics were omitted altogether. Training algorithms for feedforward networks constitute a broad field of research, whose thorough review would require a dissertation on its own. For example only off-line training with fixed training data has been considered, while applications to adaptive control often require on-line weight adaptation with new data arriving in each iteration. Also the problem of network size selection was not given any attention, as well as the difficult issue of avoiding local minima of the error function. These problems are topics of ongoing research that is continuously reported in the neural network journals.

Chapter 3
Time optimal control

Optimal control offers a conceptually attractive approach to the problem of controller synthesis. Whenever the quality of the control process may be assessed in form of some numerical criterion, it is natural to seek the controller that will maximize (or, depending on the convention, minimize) that criterion. Reduction of the design process to optimization of a scalar quality criterion is very appealing from the engineering point of view, and allows to obtain, at least in principle, "the best" controller for the given control task. This is possible if there is indeed a way to measure the control quality with a single number, or a quality index. One of the most popular choices of such an index is the time of transfer to some prespecified set in the system state space. The resulting optimal control problem is usually referred to as minimum time control or time optimal control.

The majority of results available on optimal control are concerned with the open-loop control problem. That is, for a given initial condition, the task is to find a control signal, as a function of time, that will minimize the quality criterion and satisfy other performance specifications. For a large class of problems with integral quality criterion efficient computational methods are available to solve this problem if the exact system model is available. However, from the control engineer point of view, the open-loop approach has seldom any practical value. Because of the inevitable modeling inaccuracies, the obtained open-loop control signal will almost never satisfy the postulated performance constraints, let alone optimize the quality index. Also, the possibility of disturbances adds to inherent nonrobustness of the open-loop control policies. Therefore, in the majority of practical applications, it is usually imperative to use measurement feedback, that is to make the control value dependent on the measured system outputs. With this requirement, the optimal control problem becomes considerably more difficult. For nonlinear systems, no general method exists that would allow synthesis of an optimal feedback controller, and would be comparable, in terms of computational efficiency, to algorithms developed for the open-loop control case. It has been suggested that artificial neural networks be used as a tool for synthesis of optimal feedback [17]. With open-loop optimal trajectories readily available using standard optimization approaches, an artificial neural network would be used to extract the information about the optimal feedback contained in those trajectories. With the particular controller structure, considered here, in the form of state feedback $u(t) = \phi(x(t))$, the data used for the synthesis task would consists of a number of optimal state and control trajectories $x(t)$ and $u(t)$. This data would be used as a source of training data for the neural network to deduce the nature of association between $x(t)$ and $u(t)$. The a priori unknown mapping ϕ would be learned and approximated by a neural network, which would be used as an approximately optimal controller. Thus, a difficult task of closed-loop synthesis would be replaced with a considerably simpler task of open-loop optimization with respect to the control signal, and the synthesis would be completed through automatic training of a neural network. With the resulting output of the neural controller approximately equal to the desired optimal control, the state trajectory would also be expected to approximate the optimal one.

This idea is based on two assumptions, that require closer examination. First, it must be guaranteed that the optimal control can indeed be expressed as a function of the instantaneous state. Second, if this is the case, it must be possible to approximate the optimal feedback using a neural network controller, in a manner which will assure that the resulting closed-loop trajectories are a close approximation of open-loop optimal trajectories. The first assumption seems intuitive from the engineering point of view, since system state should represent all information needed to choose an appropriate control action. For the continuous-time systems, this intuition is misleading. It turns out that it is very difficult to assure existence of optimal feedback, even if the open-loop optimal control problem is well defined and possesses a solution for any initial condition. The theoretical reasons for the possible failure of optimal feedback to exist are reviewed in [27] and [28]. Roughly speaking, they are related to the problem of uniqueness of the optimal control signal. A rather unpleasant consequence of the above is that, in the situation when it is not possible to find any feedback mapping providing optimal control, any attempts to approximate the nonexistent optimal feedback with neural networks will necessarily be quite futile. Thus suboptimal control by neural networks may lack suitable theoretical justification.

Time-optimal control has been studied mostly for continuous time systems, and the majority of available results are for linear systems and for some very restricted classes of nonlinear systems [29]. Synthesis of time optimal feedback was analyzed only for two dimensional systems with scalar control, for which phase plane techniques are applicable [30]. The quality index, that is the minimal time required to reach the target, is not guaranteed to be smooth as a function of initial condition and the dynamic programming approach cannot be used to show existence of optimal feedback. Hence, the previously made remarks apply to approximation of time optimal control in continuous time by neural networks. It is proposed here that the problem be studied in a discrete-time rather than continuous-time setting. Since a majority of modern hardware implementations are digital controllers operating in sample and hold mode, it seems natural from the engineering point of view to consider discrete-time modeling. It turns out that a number of theoretical difficulties plaguing the continuous time problem disappear in discrete time. For example, the problem of existence and uniqueness of a solution does not arise for a set of difference equations. Also, control and state trajectories are numerical sequences, that are much easier to deal with than functions defined on the real line. In addition, the problem of optimal feedback becomes more tractable. This is perhaps best illustrated by a simple example. Consider a one dimensional continuous time system

$$\dot{x}(t) = u(x(t)) \tag{3.1}$$

and its discrete time counterpart

$$x_{k+1} = x_k + u_k \tag{3.2}$$

with control signal constrained in both cases to the interval $[-1, 1]$, and the target set being the origin. For such a simple system the continuous time optimal feedback exists, but is discontinuous

$$\phi(x(t)) = \begin{cases} 1 & \text{if } x(t) < 0 \\ 0 & \text{if } x(t) = 0 \\ -1 & \text{if } x(t) > 0 \end{cases} \tag{3.3}$$

while for the discrete time case the optimal feedback may be taken as continuous with

$$\phi(x_k) = \begin{cases} 1 & \text{if } x_k < -1 \\ -x & \text{if } -1 \leq x_k \leq 1 \\ -1 & \text{if } x_k > 1 \end{cases} \tag{3.4}$$

The reason for the difference is obvious. In the discrete-time setting the minimal time necessary to reach the origin is constant on interval $(-1, 1)$, thus allowing for continuous dependence of optimal control on the instantaneous state. This illustrates a general rule that more regularity is introduced into the time optimal control problem when the discrete time systems are considered.

Approximation of time optimal control by artificial neural networks will be studied here in a discrete time setting. First, existence of optimal feedback will be resolved after a proper relaxation of the problem by replacement of the target point by its small neighborhood. Then, the obtained feedback mapping will be approximated by discontinuous and continuous feedforward neural networks. In the former case, suboptimal control will be guaranteed for compact sets of initial conditions, while in the latter case an arbitrarily small subset of initial conditions will have to be excluded. The results are concerned only with conditions under which there exists a proper neural network realizing suboptimal control. A statement, that it is indeed possible to find such a network, will be understood only in the sense, that for some, a priori unknown, weight vector, whose dimensionality also cannot be specified, the resulting network may be used as a desired controller. The problem of finding the actual values of the network weights, or training the neural controller, is not addressed here, and neither is selection of the network size. Before these issues become a concern, first it has to be assured that the set of feasible neural networks is not empty, or attempts to train a suboptimal controller may have no chance of success. This existence issue is successfully resolved here for a fairly general class of dynamical systems.

3.1 Problem statement

The time optimal control problem will be considered for a class of finite dimensional time invariant dynamical systems, that admit a model of the following form:

$$x_{k+1} = f(x_k, u_k) \tag{3.5}$$

with $x_k \in \Re^n$ and $u_k \in U \subset \Re^m$, where x_k is the state of the system, and u_k the control signal applied at time instant k. Set U represents constraints placed on admissible values of the control signal. Both the state transition function $f : \Re^n \times \Re^m \to \Re^n$ and set U do not depend on time index k. In the majority of practical cases dimensionality of the control vector is lower than that of the state vector, $m < n$, however it is neither required nor used here. It is assumed that $u_k = 0$ is always an admissible control, $0 \in U$, and that the function f satisfies

$$f(0,0) = 0 \tag{3.6}$$

that is, the origin of the state space is an equilibrium of the system corresponding to the origin of the control space. This assumption is not restrictive, since only time optimal control to an equilibrium will be considered here, so that the state may remain in the target point once it is reached. If the target state considered is an equilibrium $x_{eq} \neq 0$ corresponding to some other control value $u_{eq} \neq 0$, $u_{eq} \in U$, so that $f(x_{eq}, u_{eq}) = x_{eq}$, then a simple coordinate transformation $\tilde{x}_k = x_k - x_{eq}$, $\tilde{u}_k = u_k - u_{eq}$ yields a system satisfying (3.6).

3.1.1 Special notations

By a slight abuse of notation, the same symbol f will be used to denote the state transition mapping $f : \Re^n \times U \rightarrow \Re^n$, as well as the associated set valued transformations. Suppose that $u \in U$ and $A \subset \Re^n$. Then the image of set A under control u is denoted

$$f(A, u) = \{y \in \Re^n : \; y = f(x, u), \; x \in A\} \tag{3.7}$$

Similarly, for $x \in \Re^n$ and $V \subset U$, the set reachable in one step from x under control values from set V is

$$f(x, V) = \{y \in \Re^n : \; y = f(x, u), \; u \in V\} \tag{3.8}$$

The set of states reachable from A under controls from V, the image of $A \times V$, is denoted

$$f(A, V) = \bigcup_{x \in A} f(x, V) = \bigcup_{u \in V} f(A, u) \tag{3.9}$$

Additional simplified notation is used for the set reachable from set A in one step using any admissible controls from U

$$f(A) = \bigcup_{u \in U} f(A, u) = f(A, U) \tag{3.10}$$

This multipurpose usage of the same symbol f should not lead to any misunderstandings, as the meaning will follow from the arguments used.

For a fixed control value, $u_k = u \in U$, the corresponding unforced dynamical system is denoted

$$x_{k+1} = f_u(x_k) = f(x_k, u) \tag{3.11}$$

Of particular interest will be behavior of the zero input dynamics

$$x_{k+1} = f_0(x_k) = f(x_k, 0) \tag{3.12}$$

which has the origin as its equilibrium. A sequence of $K > 0$ control values $u_i \in U$, $k \leq i \leq k+K-1$, will be denoted $\{u_i\}_{i=k}^{k+K-1}$. If this sequence is applied starting at moment k, with the initial condition x_k, the resulting state at moment $k+K$ may be expressed as

$$x_{k+K} = f^K(x_k, \{u_i\}_{i=k}^{k+K-1}) \tag{3.13}$$

with the K-step state transition function $f^K : \Re^n \times U^K \to \Re^n$ defined by

$$f^K(x_k, \{u_i\}_{i=k}^{k+K-1}) = \underbrace{f(\ldots(f(f}_{K\ times}(x_k, u_k), u_{k+1}), \ldots), u_{k+K-1}) \tag{3.14}$$

As before, for a given control sequence $\{u_i\}_{i=0}^{K-1} \in U^K$, the image of a set $A \subset \Re^n$ is denoted

$$f^K(A, \{u_i\}_{i=0}^{K-1}) = \{y \in \Re^n : \ y = f^K(x, \{u_i\}_{i=0}^{K-1}, \ x \in A\} \tag{3.15}$$

and the set reachable from A in K steps using any admissible controls from U is

$$f^K(A) = \bigcup_{\{u_i\}_{i=0}^{K-1} \in U^K} f^K(A, \{u_i\}_{i=1}^{K}) \tag{3.16}$$

For investigation of time optimal control, a notion of controllability will be needed. A point $x \in \Re^n$ is called controllable to a set $A \subset \Re^n$ in $K > 0$ steps if there exists an admissible control sequence $\{u_i\}_{i=0}^{K-1}$, which transfers x to A, that is

$$f^K(x, \{u_i\}_{i=0}^{K-1}) \in A \tag{3.17}$$

The set of all states controllable to A in K steps is denoted as

$$\mathcal{C}^K(A) = \{x \in \Re^n : \ f^K(x, \{u_i\}_{i=0}^{K-1}) \in A, \quad u_i \in U\} \tag{3.18}$$

Occasionally controllability will be needed with control values restricted to some small subset of U. In particular, for $\delta > 0$, the set controllable to A in K steps with controls satisfying $\| u_k \| < \delta$ is denoted as

$$\mathcal{C}_\delta^K(A) = \{x \in \Re^n : \ f^K(x, \ \{u_i\}_{i=1}^{K}) \in A, u_i \in U \cap \mathcal{B}^m(0, \delta)\} \tag{3.19}$$

A point $x \in \Re^n$ is called controllable to the set A, if there exists an integer $K > 0$ such that x is controllable to A in K steps. The set of points controllable to A is denoted

$$\mathcal{C}(A) = \bigcup_{i=1}^{\infty} \mathcal{C}^i(A) \tag{3.20}$$

and the set controllable to A using controls from $\mathcal{B}^m(0, \delta)$ is denoted

$$\mathcal{C}_\delta(A) = \bigcup_{i=1}^{\infty} \mathcal{C}_\delta^i(A) \tag{3.21}$$

3.1.2 Assumptions about the system

To facilitate the theoretical developments of this chapter, the following specific assumptions are made about the dynamical system in question.

Assumption 3.1 *The set of admissible control values U is compact, such that $clos(int(U)) = U$, and $0 \in int(U)$.*

Assumption 3.2 *State transition function $f : \Re^n \times U \rightarrow \Re^n$ is continuous.*

Assumption 3.3 *For each fixed control value $u \in U$, the state transition function $f_u : \Re^n \rightarrow \Re^n$ of the unforced dynamics (3.11) is one-to-one. Furthermore, the inverse dynamics function $f^{-1} : \Re^n \times U \rightarrow \Re^n$ is well defined by $(f^{-1}(f(x,u),u) = x$, and is continuous.*

Assumption 3.4 *There exist $\delta_{Lx} > 0$ and $\delta_{Lu} > 0$, such that $\mathcal{B}^m(0, \delta_{Lu}) \subset int(U)$, and that f is locally Lipschitz continuous on $\mathcal{B}^n(0, \delta_{Lx}) \times \mathcal{B}^m(0, \delta_{Lu})$, that is*

$$\| f(x_1, u_1) - f(x_2, u_2) \| \leq L_x \| x_1 - x_2 \| + L_u \| u_1 - u_2 \| \tag{3.22}$$

for all $x_1, x_2 \in \mathcal{B}^n(0, \delta_{Lx})$, $u_1, u_2 \in \mathcal{B}^m(0, \delta_{Lu})$. Without loss of generality it is assumed that $L_x > 1$ and $L_u > 1$.

Assumption 3.5 *There exists an integer $M > 0$, such that for every $\delta_u > 0$ it is possible to choose $\delta_x > 0$ satisfying $\mathcal{B}^n(0, \delta_x) \subset \mathcal{C}^M_{\delta_u}(\{0\})$, that is every state x such that $\| x \| < \delta_x$ is controllable to origin in M steps using controls satisfying $\| u \| < \delta_u$.*

Assumption 3.6 *For zero control signal the unforced dynamical system (3.12) is asymptotically stable on some open neighborhood V of the origin.*

Assumption 3.1 is the least restrictive, as in the majority of practical applications the admissible control values are bounded and contained in a closed set. An example most typical for technical systems is a set U being an m-dimensional rectangle

$$U = \times_{i=1}^{m}[a_i, b_i], \quad a_i < 0 < b_i \tag{3.23}$$

Again, should the bounds a_i and b_i exclude the origin of $\Re^m$ as an admissible control value, a simple shift of control variables will allow for condition $0 \in \mathrm{int}(U)$ to be satisfied.

Continuity of f with respect to both the state and the control signal is a natural assumption, without which very little could be achieved in terms of approximation of any control policy. The typical case is when the discrete time system (3.5) is a result of sampling of a continuous time system

$$\dot{\xi}(t) = f_c(\xi(t), v(t)) \tag{3.24}$$

with $x_k = \xi(k\Delta)$ and $v(t) = u_k$ for $t \in [k\Delta, (k+1)\Delta)$, where Δ is the sampling time. If f_c is continuous and such that (3.24) admits a unique solution on the interval $[k\Delta, (k+1)\Delta]$ for any constant control $v(t) = v$, then it is well known that $\xi((k+1)\Delta)$ depends continuously on initial condition $\xi(k\Delta)$ and on parameter v, and thus the function f is continuous. The requirement stated in assumption 3.3, that function f_u is one-to-one for any admissible control value u, may be seen as more restrictive. Observe however, that if solutions of (3.24) are unique, so are the solutions of the reversed time system

$$\dot{\zeta}(t) = -f_c(\zeta(t), \eta(t)) \tag{3.25}$$

obtained by substituting $\zeta(t) = \xi(-t)$, $\eta(t) = v(-t)$. Therefore, the initial condition $\xi(k\Delta)$ can also be expressed as a unique function of the final point $\xi(k+1)\Delta$ and on parameter v, and consequently inverse dynamics f^{-1} is well defined and continuous. Since practically all reasonable continuous time systems indeed admit unique solutions (see [31], pages 469-473), assumption 3.3 appears reasonable from the practical point of view, and is satisfied for control systems resulting from sampling of physical continuous time plants.

Assumptions 3.4 and 3.5 are slightly more restrictive, even though Lipschitz continuity is very often regarded in literature as standard. Assumption 3.5 is implied, for example, by controllability of the linearized system (if it exists). In such a case assumption 3.4 is also satisfied.

The most restrictive assumption 3.6 will be needed only in part of the results of this chapter. For a linearizable system this stability requirement may be satisfied by applying a locally stabilizing linear feedback. Assume that f is differentiable and that the linear approximation of the system

$$\tilde{x}_{k+1} = A\tilde{x}_k + B\tilde{u}_k \tag{3.26}$$

may be stabilized by means of a state feedback

$$\tilde{u}_k = K\tilde{x}_k \tag{3.27}$$

Then it is possible to apply the linear stabilizing feedback to the system (3.5) to obtain a locally asymptotically stable system

$$x_{k+1} = f(x_k, Kx_k + \hat{u}_k) \tag{3.28}$$

to which the suboptimal strategies of this chapter may be later applied. The problem with this approach to the controller is, however, that the set of admissible control values $\hat{u}_k$ will depend on the current value of the system state, as condition $Kx_k + \hat{u}_k \in U$ has to be satisfied for all k. A remedy to this complication might be to use a nonlinear additive feedback $\tilde{\phi}_s(x_k)$, such that it is equal to Kx_k in the vicinity of the origin, but vanishes outside some small neighborhood of the origin, so that $\| \tilde{\phi}_s(x_k) \| < \varepsilon$ for any x_k. Then, the resulting stabilized system

$$x_{k+1} = f(x_k, \phi_s(x_k) + \hat{u}_k) \tag{3.29}$$

is locally asymptotically stable, and if the control $\hat{u}_k$ is such that

$$\hat{u}_k \in \hat{U} = \{u \in U : \text{dist}(u, U^c) \geq \varepsilon\} \tag{3.30}$$

then the combined control signal satisfies

$$\hat{u}_k + \phi_s(x_k) \in U \tag{3.31}$$

If ε is sufficiently small, then the modified set of admissible control values $\hat{U}$ defined by (3.30) is for all practical purposes equivalent to the original set U, and the locally stable system (3.29) may be used in the developments of this chapter. It must be cautioned that after such a maneuver the controllability assumption 3.5 may no longer be satisfied for the stabilized system. However, that assumption will be not needed for systems that are already locally asymptotically stable. Likewise, invertibility assumption 3.3 may be affected by application of a locally stabilizing feedback. That requirement will be used sparingly, though, to weaken the assumptions on sets of initial conditions for which suboptimal feedback will be synthesized. For most of the developments it will be enough that continuity assumption 3.2 holds, which would not be affected as long as feedback ϕ_s were continuous.

3.1.3 Time optimal control problem

The optimal control problem considered in this chapter is the minimum time transfer of the system state to the origin. For the problem to be tractable, the initial condition must be controllable to the origin, $x \in \mathcal{C}(\{0\})$. Then the optimal transfer time is defined as

$$T(x) = \min\{k \in N : x \in \mathcal{C}^k(\{0\})\} \tag{3.32}$$

and the optimization task is to find a control sequence $\{\hat{u}_i\}_{i=0}^{T(x)-1}$ with $\hat{u}_i \in U$ such that

$$f^{T(x)}(x, \{u_i\}_{i=0}^{T(x)-1}) = 0 \tag{3.33}$$

The range of the quality index $T(x)$, treated as a function of the initial condition, is the set of natural numbers, hence function (3.32) cannot be continuous. Usually the solution of thus defined optimal control problem is not unique with respect to the independent variable $\{u_i\}_{i=0}^{T(x)-1}$), and control values satisfying (3.33) may form an uncountable subset of $U^{T(x)}$.

Assuming a certain set of initial conditions of interest, $C \subset \mathcal{C}(\{0\})$, the feedback synthesis task considered here is to find a state feedback mapping $\phi : \Re^n \to U$, such that for any initial condition $x_0 \in C$ the solution of the closed loop system

$$x_{k+1} = f(x_k, \phi(x_k)) \tag{3.34}$$

satisfies $x_k = 0$ for all $k \geq T(x_0)$. This exact optimization task may be very difficult to achieve. Neural network controllers considered later in this chapter will be able to represent only an approximation of the strictly time optimal control. Therefore, for the purpose of synthesis of feedback control strategies, a relaxed near time optimal control problem is introduced. For a small $\varepsilon > 0$, the target set is the open neighborhood of the origin, $\mathcal{B}^n(0, \varepsilon)$, rather than the origin itself. Then a feedback mapping ϕ is sought such that for any $x_0 \in C$ the solution of the closed loop system (3.34) satisfies $x_k \in \mathcal{B}^n(0, \varepsilon)$ for $k \geq T(x_0)$. If this can be achieved for an arbitrary ε then for all practical purposes time optimal transfer to the origin will be assured.

3.2 Controllability properties

This section is concerned with three lemmas regarding existence of certain sets controllable to each other. These results are by themselves rather trivial consequences of assumptions made in section 3.1.2. However, they will be crucial in construction of suboptimal feedback. The first lemma ascertains existence of a sufficiently small neighborhood of the origin controllable to itself using small control values.

Lemma 3.7 *Let system (3.5) satisfy assumptions 3.2, 3.4 and 3.5. Then for any* $\varepsilon > 0$ *and any* $\delta > 0$ *there exists an open neighborhood* A_0 *of the origin, such that* $A_0 \subset \mathcal{B}^n(0,\varepsilon)$, *and* $clos(A_0) \subset \mathcal{C}^1_\delta(A_0)$.

Proof: From the assumption 3.5 it follows that that for any $\delta_u > 0$ there exists a $\delta_x > 0$ such that $\mathcal{B}^n(0,\delta_x) \subset \mathcal{C}^M_{\delta_u}(\{0\})$. Let

$$\tilde{\varepsilon} = min(\varepsilon, \delta_{Lx}), \tag{3.35}$$

and choose a positive δ_u, such that

$$\delta_u < min(\delta, \delta_{Lu}, \frac{\tilde{\varepsilon}}{2L_u \sum_{i=0}^{M-1}(L_x)^i}) \tag{3.36}$$

where constants δ_{Lx}, δ_{Lu}, L_x, and L_u are as in assumption 3.4. Next choose a corresponding δ_x, small enough so that $\mathcal{B}^n(0,\delta_x) \subset \mathcal{C}^M_{\delta_u}(\{0\})$, and satisfying

$$\delta_x < \frac{\tilde{\varepsilon}}{2L_u((L_x)^{M-1}+1)} \tag{3.37}$$

Now construct M open sets B_i, $i = 1, \ldots, M$, such that

$$\text{clos}(B_{i-1}) \subset \mathcal{C}^1_{\delta_u}(B_i) \tag{3.38}$$

$$\text{clos}(B_i) \subset \mathcal{C}^{M-i+1}_{\delta_u}(\mathcal{B}^n(0, \frac{i}{M}\delta_x)) \tag{3.39}$$

The recursive construction is as follows. For $i = 1$, let $B_1 = \mathcal{B}^n(0,\delta_x)$, for which (3.38) is vacuous and (3.39) is satisfied by choice of δ_x. Suppose now that B_i has already been defined, and construct B_{i+1} in the following way. To each $x \in B_i$ assign a corresponding sequence of $M-i+1$ control values $\{u_{x,j}\}_{j=1}^{M-i+1}$, $\| u_{x,j} \| < \delta_u$ transferring x into a sufficiently small neighborhood of origin:

$$\| f^{M-i+1}(x, \{u_{x,j}\}_{j=1}^{M-i+1}) \| < \frac{i}{M}\delta_x \tag{3.40}$$

This is possible, since B_i satisfies (3.39). The choice of control sequence defines also a collection of points $y_x = f(x, u_{x,1})$ which satisfy

$$\| f^{M-i}(y_x, \{u_{x,j}\}_{j=2}^{M-i+1}) \| < \frac{i}{M}\delta_x \tag{3.41}$$

Thus, the set

$$\tilde{B} = \bigcup_{x \in B_i} \{y_x\} \tag{3.42}$$

is a subset of $\mathcal{C}_{\delta_u}^{M-i}(\mathcal{B}^n(0, \frac{i+1}{M}\delta_x))$. This set satisfies both (3.38) and (3.39). Unfortunately, it need not be open, and cannot be taken as B_{i+1}. To amend that, choose $\tilde{\delta}_i > 0$ such that

$$\tilde{\delta}_i < \frac{\delta_x}{2M(L_x)^{M-i}} \tag{3.43}$$

and define an open set B_{i+1} as

$$B_{i+1} = \bigcup_{x \in B_i} \mathcal{B}^n(y_x, \tilde{\delta}_i) \tag{3.44}$$

Take any $\tilde{x} \in \mathrm{clos}(\mathcal{B}^n(y_x, \tilde{\delta}_i))$ for some $x \in B_i$. Then control sequence $\{u_{x,j}\}_{j=2}^{M-i+1}$ transfers $\tilde{x}$ into $\mathcal{B}^n(0, \frac{i+1}{M}\delta_x)$. To see that, first observe that

$$B_1 \subset \mathcal{B}^n(0, \delta_x) \subset \mathcal{B}^n(0, \delta_{Lx}) \tag{3.45}$$

hence f is Lipschitz continuous on $B_1 \times \mathcal{B}^m(0, \delta_u)$. Next, recursive application of Lipschitz continuity results in

$$B_i \subset \mathcal{B}^n(0, \delta_x(L_x^{i-1} + \frac{i-1}{ML_x^{M-i+1}}) + \delta_u L_u \sum_{j=0}^{i-1} L_x) \subset \mathcal{B}^n(0, \tilde{\varepsilon}) \tag{3.46}$$

where the first inclusion uses definition of $\tilde{\delta}_i$, and the second inclusion follows from (3.36), (3.37) and from assumption that $L_x > 1$ and $L_u > 1$. Thus, f is Lipschitz continuous on $B_i \times \mathcal{B}^m(0, \delta_u)$. Now apply the sequence of $M-i$ controls $\{u_{x,j}\}_{j=2}^{M-i+1}$ to $y_x = f(x, u_{x,1})$ and to $\tilde{x}$. From Lipschitz continuity it follows that

$$\| f^{M-i}(\tilde{x}, \{u_{x,j}\}_{j=2}^{M-i+1}) - f^{M-i}(y_x, \{u_{x,j}\}_{j=2}^{M-i+1} \|) < L_x^{M-1}\tilde{\delta}_i < \frac{\delta_x}{M} \tag{3.47}$$

which, together with (3.41), implies that

$$\| f^{M-i}(\tilde{x}, \{u_{x,j}\}_{i=2}^{M-i+1}) \| < \frac{i+1}{M}\delta_x \tag{3.48}$$

Thus B_{i+1} satisfies (3.39), and the recursive construction is complete. With the family of open sets B_i already defined, the desired set A_0 is obtained as

$$A_0 = \bigcup_{i=1}^{M} B_i \tag{3.49}$$

By (3.38), and because $B_M \subset \mathcal{C}^1_{\delta_u}(\mathcal{B}^n(0,\delta_x)) = \mathcal{C}^1_{\delta_u}(B_1)$, it follows that $A_0 \subset \mathcal{C}^1_{\delta}(A_0)$. Finally, from (3.46) and (3.35) it follows that

$$A_0 \subset \mathcal{B}^n(0,\varepsilon) \tag{3.50}$$

as required. This concludes the proof.

A similar, and actually a little stronger, result may be shown, if local asymptotic stability of the zero input system (3.12) is assumed. It is then not necessary to invoke the controllability assumption 3.5 or the Lipschitz continuity assumption 3.4.

Lemma 3.8 *Let system (3.5) satisfy assumptions 3.1, 3.2 and 3.6. Then for any* $\varepsilon > 0$ *it is possible to find open sets* A_0 *and* V_0*, and a number* $\delta > 0$*, such that* $clos(V_0) \subset A_0 \subset \mathcal{B}^n(0,\varepsilon)$ *and if* $x \in A_0$*, and* $\| u \| < \delta$*, then* $f(x,u) \in V_0$*.*

Proof: The proof can be found in the proof of lemma 3.1 in [16], where the result is stated without the requirement that $A_0 \subset \mathcal{B}^n(0,\varepsilon)$. But this is a straightforward consequence of asymptotic stability of (3.12) and continuity of f.

The next lemma will play the crucial role in construction of suboptimal feedback in the following sections. It establishes existence of certain increasing families of sets that will be later used to cover the compact set of initial conditions of interest.

Lemma 3.9 *Suppose system (3.5) satisfies assumptions 3.1 through 3.5, or assumptions 3.1 through 3.2 and 3.6. Then for any number* $\varepsilon > 0$ *there exists a number* $\delta > 0$ *and sequences of open sets* $\{A_i\}_{i=0}^{\infty}$*, compact sets* $\{C_i\}_{i=1}^{\infty}$*, and open sets* $\{B_i\}_{i=1}^{\infty}$ *such that*

$$clos(A_0) \subset \mathcal{C}^1_{\delta}(A_0) \tag{3.51}$$

$$A_0 \subset \mathcal{B}^n(0, \varepsilon) \tag{3.52}$$

$$A_0 \subset A_1 \tag{3.53}$$

$$\mathcal{C}^i(\{0\}) \subset A_i \subset C_i \subset B_i \subset \mathcal{C}^1(A_{i-1}) \tag{3.54}$$

Proof: For $i = 0$ the desired set A_0 satisfying (3.51) and (3.52) exists by virtue of lemma 3.7 or lemma 3.8. Suppose now that an open A_i has been defined for some $i \geq 0$, such that $\mathcal{C}^i(\{0\}) \subset A_i$, and construct appropriate sets A_{i+1}, B_{i+1}, and C_{i+1}. First, observe that by compactness of U and continuity of inverse state transition function f^{-1}, set $\mathcal{C}^{i+1}(\{0\})$ is compact. Define

$$B_{i+1} = \mathcal{C}^1(A_i) \tag{3.55}$$

which satisfies $\mathcal{C}^{i+1}(\{0\}) \subset B_{i+1}$. It is also an open set - to see that, take any control value $u \in U$ and consider a set

$$B_{i+1}^u = \{x : f(x, u) \in A_i\} \tag{3.56}$$

which is open by openness of A_i and by continuity of f. Then, B_{i+1} can be expressed as

$$B_{i+1} = \bigcup_{u \in U} B_{i+1}^u \tag{3.57}$$

and it is open itself. Now it is always possible to find an open set A_{i+1} and a compact set C_{i+1} so that (3.54) is satisfied. To see that (3.53) can be satisfied as well, use the property $\mathrm{clos}(A_0) \subset \mathcal{C}^1(A_0)$. Thus, in the first step of the recursive construction it is possible to take $A_0 = A_1$ and satisfy (3.54) for $i = 1$. This concludes the proof.

3.3 Suboptimal feedback by discontinuous networks

The simplicity of the time optimal control problem allows a reduction to study of sequences of sets controllable to the origin, or in the case of suboptimal control, to some small neighborhood of the origin. Therefore, the lemmas discussed in the previous section provide a sufficient basis for investigation of the existence of suboptimal feedback policies. In this section it will be shown that they may be realized by multilayered feedforward neural networks with Heaviside hidden neurons.

3.3.1 Existence of measurable suboptimal feedback

The first step towards neural suboptimal feedback is to establish that suboptimal control may be expressed in the form of state feedback. As remarked earlier, in the continuous time case it is very difficult to obtain an affirmative answer even to such a general question. With the previously stated assumptions, it is shown here that for the discrete time a measurable suboptimal feedback indeed exists, if the set of initial conditions of interest is such that for any element of this set the open loop time optimal control problem has a well defined solution.

Proposition 3.10 *Let system (3.5) satisfy assumptions 3.1 through 3.5. Then for any set $C \in \mathcal{C}(\{0\})$, and for any $\varepsilon > 0$ there exists a measurable feedback function $\phi : \Re^n \to U$, such that for any initial condition $x_0 \in C$, the solution of the closed-loop system (3.34) satisfies $\| x_k \| < \varepsilon$ for all $k \geq T(x_0)$. Function ϕ takes countably many values, and if C is compact it takes finitely many values.*

Proof: Construct sequences of open sets A_i and B_i, and of compact sets C_i as in lemma 3.9. Note that

$$C \subset \mathcal{C}(\{0\}) = \bigcup_{i=1}^{\infty} \mathcal{C}^i(\{0\}) \subset \bigcup_{i=1}^{\infty} A_i \subset \bigcup_{i=1}^{\infty} C_i \tag{3.58}$$

so that set C is covered by open sets A_i, and consequently by compact sets C_i. The required feedback mapping will be constructed on the latter. For every $i > 0$, a covering of the compact set C_i is defined as follows. With each point $x \in C_i$ associate a control value $u_{i,x} \in \text{int}(U)$ such that $f(x, u_{i,x}) \in A_{i-1}$ - this is possible because of (3.54). Since f is continuous and A_{i-1} is open, there exists an open neighborhood of x, $V_{i,x} \subset B_i$, such that $f(\tilde{x}, u_{i,x}) \in A_{i-1}$ for any $\tilde{x} \in V_{i,x}$. Sets $V_{i,x}$ form an open covering of the compact set C_i, so it is possible to find a finite subcover $\{V_{i,j}\}_{j=1}^{N_i}$, with corresponding points $x_{i,j}$ and control values $u_{i,j}$. For each $V_{i,j}$, $i > 1$, $1 \leq j \leq N_i$, define its subset $W_{i,j}$ as

$$W_{i,j} = (V_{i,j} \setminus \bigcup_{k=1}^{j-1} V_{i,k}) \cap (C_i \setminus \bigcup_{k=1}^{i-1} C_k) \tag{3.59}$$

Sets $W_{i,j}$ are mutually disjoint and because

$$\bigcup_{i=1}^{\infty} \bigcup_{j=1}^{N_i} W_{i,j} = \bigcup_{i=1}^{\infty} \bigcup_{j=1}^{N_i} V_{i,j} \tag{3.60}$$

they cover set of admissible initial conditions C. It is possible now to define a desired feedback function

$$\phi(x) = \sum_{i=1}^{\infty} \sum_{j=1}^{N_i} u_{i,j} \chi_{W_{i,j}}(x) \tag{3.61}$$

where χ_W is the characteristic function of set W. Sets $V_{i,j}$ and C_i are open and closed, respectively, hence Borel measurable. Therefore sets $W_{i,j}$ are also Borel measurable and consequently the feedback function ϕ is Borel measurable. It is also observed that ϕ takes countably many values $u_{i,j}$. To see that ϕ provides near-optimal control, first consider a point $x \in W_{i,j} \subset C_i$ for some $i > 0$, $1 \leq j \leq N_i$. Then $\phi(x) = u_{i,j}$, and from the construction of sets $V_{i,j}$ it follows that $f(x, \phi(x)) \in A_{i-1}$. Observe also that A_0 is an invariant set of the closed loop system - this follows from the inclusion $A_0 \subset A_1 \subset C_1$. Finally consider a solution of the closed loop system (3.34) for an arbitrary initial condition $x_0 \in C$. Suppose that, at time step $k \geq 0$, $x_k \in C_i$ for some $i \leq T(x_k)$. Then $x_k \in W_{l,j}$ for some $l \leq i$. Consequently,

$$x_{k+1} = f(x_k, \phi(x_k)) \in A_{l-1} \tag{3.62}$$

This in turn implies that for any $k \leq T(x_0)$

$$x_k \in A_l, \quad l \leq T(x_0) - k \tag{3.63}$$

and for $k \geq T(x_0)$

$$x_k \in A_0 \tag{3.64}$$

Thus $u_k = \phi(x_k)$ is a desired control law. If the set of admissible initial conditions C is compact, then it is possible to choose a finite subcovering from open cover A_i

$$C \subset \bigcup_{i=1}^{K} A_i \subset \bigcup_{i=1}^{K} C_i \tag{3.65}$$

for some integer $K > 0$. Then the feedback mapping may be defined as

$$\phi(x) = \sum_{i=1}^{K} \sum_{j=1}^{N_i} u_{i,j} \chi_{W_{i,j}}(x) \tag{3.66}$$

and it takes only finitely many values. This concludes the proof.

The above result guarantees that the system state can be driven to, and maintained in, a sufficiently small neighborhood of the origin. If, additionally, it is required that any trajectory originating in C converges to the origin, this can be guaranteed if the system is already locally asymptotically stable. This is in a sense trivial, since if the system is already stable in some neighborhood of the origin, no special control action is needed once this neighborhood is reached. What is needed to show is that the measurable suboptimal feedback may be made zero in this neighborhood. As in lemma 3.8 assumptions 3.4 and 3.5 are no longer needed.

Proposition 3.11 *Let system (3.5) satisfy assumptions 3.1 through 3.2 and 3.6. Then for any set $C \in \mathcal{C}(\{0\})$, and for any $\varepsilon > 0$ there exists a measurable feedback function $\phi : \Re^n \to U$, such that $\phi(x) = 0$ on some compact set $C_0 \subset \mathcal{B}^n(0, \varepsilon)$, and that for any initial condition $x_0 \in C$ solution of the closed loop system (3.34) satisfies $\| x_k \| < \varepsilon$ for all $k \geq T(x_0)$ and x_k converges to the origin.*

Proof: The proof is a very simple modification of the proof of proposition 3.10. Due to lemma 3.8, set A_0 may be chosen small enough, so that it is contained in the domain of asymptotic stability V specified in assumption 3.6. Additionally, an open set $V_0 \subset \mathrm{clos}(V_0) \subset A_0$ is constructed such that if $\tilde{x} \in A_0$ and $\| \tilde{u} \| < \delta$ then $f(\tilde{x}, \tilde{u} \in V_0$. Define a compact set $C_0 = \mathrm{clos}(V_0)$. Then, the only modification required in the proof of proposition 3.10 is that, while constructing the open subcovering $V_{1,j}$

of compact set C_1, it is possible to choose $V_{1,1} = A_0$ and the corresponding control value $u_{1,1} = 0$. Then, the feedback function defined by (3.66) satisfies $\phi(x) = 0$ for $x \in C_0 \subset W_{1,1}$, and any closed loop trajectory that entered A_0 converges to the origin. This concludes the proof.

Since the approximate target set $\mathcal{B}^n(0, \varepsilon)$ can be chosen arbitrarily small, the above results provide, for any practical purpose, existence of time-optimal stabilizing feedback. That ϕ can be constructed to have a finite range may be important in the case of systems whose actuators allow only finitely many output signal levels. On the other hand, the feedback constructed here is necessarily discontinuous and hence cannot be uniformly approximated by continuous neural networks. What can be shown, however, is that discontinuous neural networks may realize the suboptimal feedback defined here.

3.3.2 Set valued suboptimal feedback

Existence of suboptimal neural feedback will be studied through application of theorem 2.5. In order to use that result, the developments of the previous section need to be strengthened. A set valued, rather than single valued feedback mapping will be investigated

$$\Phi : B \to \mathcal{P}(U) \tag{3.67}$$

on some open set $B \subset \Re^n$, where $\mathcal{P}(U) = \{V : \; V \subset U\}$ is the collection of all subsets of U. The task is to find such Φ, so that if at moment k any element $u \in \Phi(x_k)$ is applied as a control signal, then the system behaves as desired. In addition, it will be required that Φ is lower semicontinuous, that is for any $x \in B$, $u \in \Phi(x)$ and $\rho > 0$ it is possible to choose a $\delta > 0$, so that if $\tilde{x} \in \mathcal{B}^n(x, \delta)$, then $\Phi(\tilde{x}) \cap \mathcal{B}^m(u, \rho) \neq \emptyset$.

Proposition 3.12 *Let system (3.5) satisfy assumptions 3.1 through 3.5. Then for any set $C \in \mathcal{C}(\{0\})$, and for any $\varepsilon > 0$, there exists an open set $B \subset \Re^n$, such that*

$C \subset B$, and a set valued feedback mapping $\Phi : B \to \mathcal{P}(U)$, such that if $u_k \in \Phi(x_k)$ for all k, then for any $x_0 \in C$ the solution of the system (3.5) satisfies $\| x_k \| < \varepsilon$ for all $k \geq T(x_0)$. The feedback mapping may be chosen such that images $\Phi(x)$ are open, and the mapping itself is lower-semicontinuous.

Proof: As before, construct a covering of set C satisfying (3.51 - 3.54) and (3.58), and for each C_i find a finite covering by open sets $V_{i,j}$. With each $V_{i,j}$ associate an open neighborhood of $u_{i,j}$, $U_{i,j} \subset U$, such that if $x \in V_{i,j}$ and $u \in U_{i,j}$, then $f(x,u) \in A_{i-1}$ - this is possible because of continuity of function f. Next, define a corresponding collection of open sets $W_{i,j}$ by

$$W_{i,j} = V_{i,j} \setminus \bigcup_{k=1}^{i-1} C_k \tag{3.68}$$

and denote

$$B = \bigcup_{i=1}^{\infty} \bigcup_{j=1}^{N_i} W_{i,j} \tag{3.69}$$

Set C is covered by open set B, and each $x \in C$ belongs to finitely many $W_{i,j}$. The desired set valued feedback mapping is then defined on B as

$$\Phi(x) = \bigcup_{i=1}^{\infty} \bigcup_{j=1}^{N_i} \chi_{W_{i,j}}(x) U_{i,j} \tag{3.70}$$

This mapping has open images and in addition is lower semi-continuous. To see this, take any $x \in B$ and any $u \in \Phi(x)$. Then $x \in W_{i,j}$, $u \in U_{i,j}$ for some $i > 0$, $1 \leq j \leq N_i$, and $u \in \Phi(\tilde{x})$ for any $\tilde{x} \in W_{i,j}$. Since $W_{i,j}$ is open this shows lower semicontinuity. Suppose now that $x \in C_i$. Then $x \notin W_{l,j}$ for any $l > i$. Therefore, if the control value satisfies $u \in \Phi(x)$, then $u \in U_{l,j}$ for some $l \leq i$ such that $x \in W_{l,j}$, $1 \leq j \leq N_l$. Then

$$f(x,u) \in A_{l-1} \tag{3.71}$$

Again, since $A_0 \subset A_1 \subset C_1$, set A_0 is invariant under any feedback satisfying $u \in \Phi(x)$. Finally, consider any solution of the closed loop system with the sevalued feedback Φ with an arbitrary initial condition $x_0 \in C$. From the above it follows that for any $k \leq T(x_0)$

$$x_k \in A_l, \quad l \leq T(x_0) - k \tag{3.72}$$

and for all $k \geq T(x_0)$ it is seen that

$$x_k \in A_0 \tag{3.73}$$

so that Φ is a desired feedback mapping. This concludes the proof.

For systems that are already locally asymptotically stable, it is again possible to show that mapping Φ can be chosen so that $0 \in \Phi(x)$ on a small neighborhood of the origin. The following result is a straightforward consequence of proofs of propositions 3.12 and 3.11.

Proposition 3.13 *Let system (3.5) satisfy assumptions 3.1, 3.2 and 3.6. Then for any set $C \in \mathcal{C}(\{0\})$, and for any $\varepsilon > 0$, there exists an open set $B \subset \Re^n$, such that $C \subset B$, a compact set $C_0 \subset \mathcal{B}^n(0, \varepsilon)$, and a set valued feedback mapping $\Phi : B \to \mathcal{P}(U)$, such that if $u_k \in \Phi(x_k)$ for all k, and $u_k = 0$ for $x_k \in C_0$, then for any $x_0 \in C$ the solution of the system (3.5) satisfies $\| x_k \| < \varepsilon$ for all $k \geq T(x_0)$, and the trajectory converges to the origin. The feedback mapping may be chosen such that images $\Phi(x)$ are open, the mapping itself is lower-semicontinuous, and $0 \in \Phi(x)$ for $x \in C_0$.*

Proof: The only modification to the proof of proposition 3.12 is the choice of proper A_0 contained in the domain of attraction of the origin and of the corresponding compact C_0, according to lemma 3.8. Then, while constructing an open covering of C_1, it is necessary to choose $V_{1,1} = A_0$ and $U_{1,1} = \mathcal{B}^m(0, \delta)$. The rest of the proof is an exact repetition of the proof of proposition 3.12.

3.3.3 Neural suboptimal feedback

The developments of the previous section lead to the existence of artificial neural networks providing suboptimal feedback. The class of networks for which this is shown are feedforward networks with at least 2 hidden layers of Heaviside neurons and the output layer with linear neurons. Theorem 2.5 together with proposition 3.12 immediately imply the following.

Corollary 3.14 *Let system (3.5) satisfy assumptions 3.1 through 3.5. Then for any compact set $C \in \mathcal{C}(\{0\})$, $\varepsilon > 0$, and for any integer $L \geq 2$, it is possible to construct a feedforward network $\phi : \Re^n \rightarrow \Re^m$, with L hidden layers, Heaviside hidden units and linear output units, such that $\phi(x) \in U$ on C, and for any initial condition $x_0 \in C$ the solution of system (3.34) satisfies $\| x_k \| < \varepsilon$ for all $k \geq T(x_0)$.*

Proof: With open set B and set valued feedback Φ, constructed as in proposition 3.12, define an open set

$$\Omega = \{(x, u) : x \in B, u \in \Phi(x)\} \subset \Re^n \times \Re^m \tag{3.74}$$

Then the compact set of admissible initial conditions C is a subset of its projection on the first n coordinates. Thus, by theorem 2.5 there exists a neural network ϕ of the required form, such that $\phi(x) \in \Phi(x)$ for any $x \in C$. But then, by proposition 3.12, this network performs the desired control task.

Similarly, for systems already locally asymptotically stable, the following can be obtained from proposition 3.13 and theorem 2.5.

Corollary 3.15 *Let system (3.5) satisfy assumptions 3.1 through 3.3 and 3.6. Then for any compact set $C \in \mathcal{C}(\{0\})$, $\varepsilon > 0$, and for any integer $L \geq 2$, it is possible to construct a feedforward network $\phi : \Re^n \rightarrow \Re^m$, with L hidden layers, Heaviside hidden units and linear output units, such that $\phi(x) \in U$ on C, and for any initial condition $x_0 \in C$ the solution of system (3.34) satisfies $\| x_k \| < \varepsilon$ for all $k \geq T(x_0)$ and the trajectory converges to the origin.*

Proof: Use proposition 3.13 to obtain Φ, B and C_0 as shown there. Then set Ω defined by (3.74) satisfies $(x, 0) \in \Omega$ for $x \in C_0$. Thus, theorem 2.5 may be used to find a proper network ϕ such that $\phi(x) \in \Phi(x)$ for any $x \in C$ and $\phi(x) = 0$ for $x \in C_0$. But then, by proposition 3.13, this network performs the desired control task.

3.4 Approximation by continuous networks

It has been demonstrated that neural networks with discontinuous neurons can be used to provide nearly time optimal feedback. Unfortunately, the majority of efficient training algorithms are based on the gradient of the error function, and require that the neuron activation functions be at least once differentiable. In practice, the most widely used neural networks have neurons with smooth sigmoidal activation functions. The theory reviewed in chapter 2 guarantees uniform approximation of only continuous functions by such networks. In order to use those results, existence of continuous near-optimal feedback must first be established. This will be possible, if a set of arbitrarily small measure is excluded from the set of admissible initial conditions C. The technique to construct such a set is dealt with in the following lemma.

Lemma 3.16 *Let system (3.5) satisfy the assumptions 3.1 through 3.3. Suppose that there exist collections of open sets* $\{E_i\}_{i=0}^{L}$*, compact sets* $\{F_i\}_{i=1}^{L}$*, and open sets* $\{G_i\}_{i=1}^{L}$*, for some* $L > 0$*, which satisfy*

$$E_i \subset F_i \subset G_i \subset \mathcal{C}^1(E_{i-1}) \tag{3.75}$$

Let $\hat{\varepsilon} > 0$ *be an arbitrary number. Then there exist collections of open sets* $\{\tilde{E}_i\}_{i=1}^{L}$*,* $\tilde{E}_0 = E_0$*, compact sets* $\{\tilde{F}_i\}_{i=1}^{L}$*,* $\tilde{F}_i \subset F_i$*, and open sets* $\{\tilde{G}_i\}_{i=1}^{L}$*, satisfying*

$$\tilde{E}_i \subset \tilde{F}_i \subset \tilde{G}_i \subset \mathcal{C}^1(\tilde{E}_{i-1}) \tag{3.76}$$

$$\mu(\bigcup_{i=1}^{L} \tilde{F}_i) > \mu(\bigcup_{i=1}^{L} F_i) - \hat{\varepsilon} \tag{3.77}$$

$$\tilde{F}_i \cap \tilde{F}_1 = \emptyset \quad \text{for } i > 1 \tag{3.78}$$

such that it is possible to find $\tilde{\delta}$ and a function $\psi : \Re^n \to int(U)$, continuous on $\tilde{F}_1$, and $\tilde{\delta} > 0$, such that if $x \in \tilde{F}_1$ and $\| u - \psi(x) \| < \tilde{\delta}$ then $f(x,u) \in E_0$, and if $x \notin \tilde{F}_1$ then $\psi(x) = 0$.

Proof: For each $1 \leq i \leq L$ increasing sequences of open sets $\tilde{E}_{i,p} \subset E_i$, compact sets $\tilde{F}_{i,p} \subset F_i$, and open sets $\tilde{G}_{i,p} \subset G_i$ will be constructed so that they satisfy

$$\lim_{p\to\infty} \mu(\bigcup_{k=1}^{i} \tilde{F}_{k,p}) = \mu(\bigcup_{k=1}^{i} F_k) \tag{3.79}$$

$$G_i = \bigcup_{p=1}^{\infty} \tilde{G}_{i,p} \tag{3.80}$$

$$\bigcup_{k=1}^{i} E_k \subset \text{clos}(\bigcup_{p=1}^{\infty} \bigcup_{k=1}^{i} \tilde{E}_{k,p}) \tag{3.81}$$

$$\tilde{E}_{i,p} \subset \tilde{F}_{i,p} \subset \tilde{G}_{i,p} \subset \mathcal{C}^1(\tilde{E}_{i-1,p}) \tag{3.82}$$

with $\tilde{E}_{0,p} = E_0$. The first step of the recursive construction is for $i = 1$. As in previous developments, a collection of open sets $\{V_j\}_{j=1}^{N}$, $N > 0$, covering the compact set F_1, is found together with associated open neighborhoods $U_j \subset U$ of control values u_j, such that if $x \in V_j$ and $u \in U_j$, then $f(x,u) \in E_0$. Since generally it is not true that the Lebesgue measure of an open set is equal to the measure of its closure, sets V_j are taken as open balls of the form $V_j = \mathcal{B}^n(x_j, \rho_j) \subset G_1$, $x_j \in F_1$, so that

$$\mu(V_j) = \mu(\text{clos}(V_j)) \tag{3.83}$$

Define a collection of mutually disjoint subsets of F_1

$$W_j = (V_j \cap F_1) \setminus \bigcup_{k=1}^{j-1} V_k \tag{3.84}$$

so that

$$F_1 = \bigcup_{j=1}^{N} W_j \tag{3.85}$$

Then take any sequence of positive numbers $\{\varepsilon_p\}_{p=1}^{\infty}$ such that $\lim_{p\to\infty} \varepsilon_p = 0$. For each $1 \leq j \leq N$ construct a decreasing sequence of open sets $\{\hat{V}_{j,p}\}_{p=0}^{\infty}$, and an increasing sequence of compact sets $\{\bar{V}_{j,p}\}_{p=0}^{\infty}$, defined by

$$\hat{V}_{j,p} = \{x \in \Re^n : \text{dist}(x, V_j) < \varepsilon_p\} \tag{3.86}$$

$$\bar{V}_{j,p} = \{x \in V_j : \operatorname{dist}(x, (V_j)^c) \geq \varepsilon_p\} \tag{3.87}$$

It is clear that

$$\bigcap_{p=1}^{\infty} \hat{V}_{j,p} = \operatorname{clos}(V_j) \tag{3.88}$$

$$\bigcup_{p=1}^{\infty} \bar{V}_{j,p} = V_j \tag{3.89}$$

Therefore by continuity of Lebesgue measure and by (3.83) it follows that for each $1 \leq j \leq N$

$$\lim_{p \to \infty} \mu(\hat{V}_{j,p}) = \mu(V_j) \tag{3.90}$$

$$\lim_{p \to \infty} \mu(\bar{V}_{j,p}) = \mu(V_j) \tag{3.91}$$

Now define recursively N sequences of compact subsets of V_j

$$\tilde{V}_{j,p} = \bar{V}_{j,p} \setminus (\bigcup_{k=1}^{j-1} \hat{V}_{j,p}) \tag{3.92}$$

For each fixed p, sets $\tilde{V}_{j,p}$ are mutually disjoint and they satisfy

$$\lim_{p \to \infty} \mu(\tilde{V}_{j,p}) = \mu(V_j \setminus (\bigcup_{k=1}^{j-1} V_k) \tag{3.93}$$

and consequently

$$\lim_{p \to \infty} \mu(\bigcup_{j=1}^{N} \tilde{V}_{j,p}) = \mu(\bigcup_{j=1}^{N} V_j) \tag{3.94}$$

Next, for each fixed p, a collection of mutually disjoint compact subsets $\tilde{W}_{j,p}$ of F_1 is defined as

$$\tilde{W}_{j,p} = F_1 \cap \tilde{V}_{j,p} \tag{3.95}$$

Finally, construct an increasing sequence of compact sets $\tilde{F}_{1,p} \subset F_1$

$$\tilde{F}_{1,p} = \bigcup_{j=1}^{N} \tilde{W}_{j,p} \tag{3.96}$$

for which it follows that

$$\lim_{p \to \infty} \mu(\tilde{F}_{1,p}) = \mu(F_1) \tag{3.97}$$

The first step of the proof is completed with setting $\tilde{E}_{1,p} = \operatorname{int}(\tilde{F}_{1,p}) \cap E_1$ and $\tilde{G}_{1,p} = G_1$, which clearly satisfy (3.80) and (3.81). Assume now that the desired construction has been performed up to some $i < L$. The desired sequences $\{\tilde{E}_{i+1,p}\}_{p=1}^{\infty}$,

$\{\tilde{F}_{i+1,p}\}_{p=1}^{\infty}$, $\{\tilde{G}_{i+1,p}\}_{p=1}^{\infty}$ will be now defined by

$$\tilde{G}_{i+1,p} = G_{i+1} \cap \mathcal{C}^1(\tilde{E}_{i,p}) \tag{3.98}$$

$$\tilde{F}_{i+1,p} = F_{i+1} \cap \{x \in \tilde{G}_{i+1,p} : \ \mathrm{dist}(x, (\tilde{G}_{i+1,p})^c) \geq \varepsilon_p, \ \mathrm{dist}(x, (\tilde{F}_{1,p})^c) \geq \varepsilon_p\} \tag{3.99}$$

$$\tilde{E}_{i+1,p} = E_{i+1} \cap int(\tilde{F}_{i+1,p}) \tag{3.100}$$

It is seen that all $\tilde{F}_{i,p}$ are disjoint from respective $\tilde{F}_{1,p}$ Also it follows that with inductive assumption (3.81) sets $\tilde{G}_{i+1,p}$ satisfy

$$\bigcup_{k=1}^{i+1} G_k = \bigcup_{p=1}^{\infty} \bigcup_{k=1}^{i+1} \tilde{G}_{k,p} \tag{3.101}$$

which, in turn implies

$$\bigcup_{k=1}^{i+1} F_k \subset \mathrm{clos}(\bigcup_{p=1}^{\infty} \bigcup_{k=1}^{i+1} \tilde{F}_{k,p}) \tag{3.102}$$

$$E_{i+1} \subset \mathrm{clos}(\bigcup_{p=1}^{\infty} \tilde{E}_{i+1,p}) \tag{3.103}$$

In particular, this means that (3.79), (3.80), and (3.81) are satisfied for $i+1$. This completes the recursive construction. Now it is possible to choose p^* sufficiently large, so that for each $1 \leq i \leq L$

$$\mu(\bigcup_{i=1}^{L} \tilde{F}_{i,p^*}) > \mu(\bigcup_{i=1}^{L} F_i) - \frac{\hat{\varepsilon}}{L} \tag{3.104}$$

Then let $\tilde{E}_i = \tilde{E}_{i,p^*}$, $\tilde{F}_i = \tilde{F}_{i,p^*}$, $\tilde{G}_i = \tilde{G}_{i,p^*}$. These sets satisfy properties (3.76), (3.77) and (3.78). To find a desired feedback function ψ set $\tilde{W}_j = \tilde{W}_{j,p^*}$, and define

$$\psi(x) = \sum_{j=1}^{N} u_j \chi_{\tilde{W}_j}(x) \tag{3.105}$$

$$\tilde{\delta} = \min_{1 \leq j \leq N} dist(u_j, U_j^c) \tag{3.106}$$

where U_j are open neighborhoods of control values u_j associated with covering sets V_j. Thus the defined feedback clearly satisfies the postulated requirements. This concludes the proof.

With the following lemma established, it is possible to discuss an approximation of the suboptimal policy discussed in section 3.3 by a continuous function.

Proposition 3.17 *Let system (3.5) satisfy assumptions 3.1 through 3.5. Then, for any compact set C, such that $C \in \mathcal{C}(\{0\})$, and for any $\varepsilon > 0$, $\tilde{\varepsilon} > 0$, there exists a compact set $\tilde{C} \subset C$ with measure $\mu(\tilde{C}) > \mu(C) - \tilde{\varepsilon}$, and a feedback function $\phi : \Re^n \to U$, continuous on $\tilde{C}$, such that for any initial condition $x_0 \in \tilde{C}$ the solution of the closed loop system (3.34) satisfies $\| x_k \| < \varepsilon$ for some $k \leq T(x_0)$. Furthermore, it is possible to choose $\varepsilon_u > 0$ such that any control law satisfying $\| u_k - \phi(x_k) \| < \varepsilon_u$ achieves the same result.*

Proof: As before, using lem3.9, collections of sets $\{A_i\}_{i=0}^K$, $\{C_i\}_{i=1}^K$, $\{B_i\}_{i=1}^K$, are constructed, which satisfy (3.53 - 3.54) and such that $C \subset \bigcup_{i=1}^K C_i$. Then the proof is based on a recursive construction of collections of open sets $\tilde{A}_{i,k} \subset A_i$, compact sets $\tilde{C}_{i,k} \subset C_i$ and open sets $\tilde{B}_{i,k} \subset B_i$, for $1 \leq k \leq K$, satisfying

$$\tilde{A}_{i,k} \subset \tilde{C}_{i,k} \subset \tilde{B}_{i,k} \subset \mathcal{C}^1(\tilde{A}_{i,k}) \tag{3.107}$$

$$\mu(\bigcup_{i=1}^K \tilde{C}_{i,k}) > \mu(\bigcup_{i=1}^K C_i) - \frac{k}{K}\varepsilon \tag{3.108}$$

$$\tilde{C}_{i,k} \cap \bigcup_{j=1}^k \tilde{C}_{j,k} = \emptyset \quad \text{for } i > k \tag{3.109}$$

Also feedback functions $\phi_k : \Re^n \to \text{int}(U)$, continuous on $\bigcup_{j=1}^k \tilde{C}_{j,k}$, are constructed, together with constants $\delta_k > 0$, such that if $x \in \tilde{C}_{i,k}$ for some $i \leq k$, and $u \in \mathcal{B}^m(\phi_k(x), \delta_k)$, then $f(x,u) \in \tilde{A}_{i,k}$, and if $x \notin \bigcup_{j=1}^k \tilde{C}_{j,k}$, then $\phi_k(x) = 0$. To perform the first step of the construction, for $k = 1$, substitute $E_i = A_i$, $F_i = C_i$, $G_i = B_i$, $L = K$, $\hat{\varepsilon} = \frac{\tilde{\varepsilon}}{K}$, and apply lemma 3.16. The obtained sets $\tilde{E}_i$, $\tilde{F}_i$, $\tilde{G}_i$ are the desired sets $\tilde{A}_{i,1} = \tilde{E}_i$, $\tilde{C}_{i,1} = \tilde{F}_i$, $\tilde{B}_{i,1} = \tilde{G}_i$ for all $1 \leq i \leq K$, Feedback function ϕ_1, continuous on $\tilde{C}_{1,1}$ is taken as $\phi_1(x) = \psi(x)$, with $\delta_1 = \tilde{\delta}$, where $\psi(x)$ and $\tilde{\delta}$ are obtained from lemma 3.16. This completes the construction for $k = 1$. Suppose now that the construction has already been performed for some $1 \leq k < K$, and construct the desired sets for $k+1$. For $1 \leq i \leq k$, let $\tilde{A}_{i,k+1} = \tilde{A}_{i,k}$, $\tilde{C}_{i,k+1} = \tilde{C}_{i,k}$, $\tilde{B}_{i,k+1} = \tilde{B}_{i,k}$. Then for $k \leq i \leq K$ substitute $E_{i-k} = \tilde{A}_{i,k}$, $F_{i-k} = \tilde{C}_{i,k}$, $G_{i-k} = \tilde{B}_{i,k}$. Also let $L = K - k$ and $\hat{\varepsilon} = \frac{\tilde{\varepsilon}}{K}$ and apply lemma 3.16. With thus obtained sets $\tilde{E}_i$, $\tilde{F}_i$, $\tilde{G}_i$ let $\tilde{A}_{k+i,k+1} = \tilde{E}_i$, $\tilde{C}_{k+i,k+1} = \tilde{F}_i$, $\tilde{B}_{k+i,k+1} = \tilde{G}_i$, for $k < i \leq K$. Since the feedback function ψ obtained from lemma 3.16 has $\tilde{C}_{k+1,k+1}$ as its support, the desired function ϕ_{k+1} may be defined simply as

$$\phi_{k+1}(x) = \phi_k(x) + \psi(x) \tag{3.110}$$

which is clearly continuous on $\bigcup_{i=1}^{k+1} \tilde{C}_{i,k+1}$ outside which it vanishes. The step $k+1$ of the recursion is completed by setting

$$\delta_{k=1} = min(\delta_k, \tilde{\delta}) \tag{3.111}$$

With sets $\tilde{C}_{i,j}$ and functions ϕ_j and numbers δ_j already constructed, the final result is obtained by setting

$$\tilde{C} = \bigcup_{i=1}^{K} \tilde{C}_{i,K} \tag{3.112}$$

$$\phi(x) = \phi_K(x) \tag{3.113}$$

$$\varepsilon_u = \delta_K \tag{3.114}$$

That feedback ϕ results in near-time-optimal transfer of the system state to set $A_0 \subset \mathcal{B}^n(0, \varepsilon)$ can be demonstrated exactly as in proof of proposition 3.10. This concludes the proof.

Unlike the results of section 3.3, proposition 3.17 does not guarantee that the system state will remain in the desired target set after it was reached at time $k \leq T(x_0)$. This is because the residual set $C \setminus \tilde{C}$ may intersect $\mathcal{B}^m(0, \varepsilon)$. So even if the state is transferred into the target set, it may leave it after finitely many steps. This may be inconsequential in applications where it is not required that the state remains in the target set. If stabilization is indeed required together with time optimal control, then local stability assumption 3.6 will be needed again. Then the following extension of proposition 3.17 is easily obtained.

Proposition 3.18 *Let system (3.5) satisfy assumptions 3.1 through 3.3 and 3.6. Then, for any compact set C, such that $C \in \mathcal{C}(\{0\})$, and for any $\varepsilon > 0$, $\tilde{\varepsilon} > 0$, there exists a compact set $\tilde{C} \subset C$ with measure $\mu(\tilde{C}) > \mu(C) - \tilde{\varepsilon}$, and a feedback function*

$\phi : \Re^n \to U$, continuous on $\tilde{C}$, such that for any initial condition $x_0 \in \tilde{C}$ the solution of system (3.34) satisfies $\| x_k \| < \varepsilon$ for all $k \geq T(x_0)$. Furthermore, it is possible to choose $\varepsilon_u > 0$ such that any control law satisfying $\| u_k - \phi(x_k) \| < \varepsilon_u$ achieves the same result.

Proof: The proof is a modified version of the proof of proposition 3.17. The difference is in the construction of sets A_0, $\tilde{A}_1$, $\tilde{C}_1$, and of mapping ϕ_1. Using lemma 3.8, the set A_0 and its subset V_0 are constructed such that $f(x,u) \in V_0$ for $x \in A_0$ and $\| u \| < \varepsilon_u$. Then in the first step of recursion, when applying lemma 3.16 with $E_0 = A_0$, $E_1 = A_1$, $F_1 = C_1$ the first of compact sets $\tilde{W}_{j,p}$ for each p is taken as $\tilde{W}_{1,p} = \text{clos}(E_0)$, and the associated control value is $u_1 = 0$. This results in inclusion $A_0 \subset \text{int}(\tilde{W}_{1,p}) \subset \tilde{E}_{1,p} \subset \tilde{F}_{1,p}$. Consequently, the obtained feedback function ϕ_1 satisfies $\phi_1(x) = 0$ on $clos(A_0)$, and $f(x,u) \in V_0$ if $x \in A_0$ and $\| u - \phi_1(x) \| < \varepsilon_x$. The rest of the proof is a repetition of the proof of proposition 3.17.

As an immediate consequence of the above results, it is possible to establish the existence of the approximation of suboptimal feedback by continuous neural networks, using theorem 2.1. If subsequent stabilization is not required after the target has been reached, the following is easily obtained.

Corollary 3.19 *Let system (3.5) satisfy assumptions 3.1 through 3.5. Then, for any compact set C, such that $C \in \mathcal{C}^K(\{0\})$ for some integer K, and for any $\varepsilon > 0$, $\tilde{\varepsilon} > 0$, and integer $L \geq 1$, there exists a compact set $\tilde{C} \subset C$ with measure $\mu(\tilde{C}) > \mu(C) - \tilde{\varepsilon}$, and a feedforward network $\phi : \Re^n \to \Re^m$ with L hidden layers, sigmoidal hidden units and linear output units, such that $\phi(x) \in U$ on $\tilde{C}$, and for any initial condition $x_0 \in \tilde{C}$ the solution of system (3.34) satisfies $\| x_k \| < \varepsilon$ for $k = T(x_0)$.*

Proof: The corollary follows immediately from proposition 3.17 and from theorem 2.1 applied to set $\tilde{C}$, function ϕ and constant ε_u.

If it is desired that the state does not leave the required neighborhood of the origin, the following result follows.

Corollary 3.20 *Let system (3.5) satisfy assumptions 3.1 through 3.3 and 3.6. Then, for any compact set* C*, such that* $C \in \mathcal{C}^K(\{0\})$ *for some integer* K*, and for any* $\varepsilon > 0$*,* $\hat{\varepsilon} > 0$*, and integer* $L \geq 1$*, there exists a compact set* $\tilde{C} \subset C$ *with measure* $\mu(\tilde{C}) > \mu(C) - \hat{\varepsilon}$*, and a feedforward network* $\phi : \Re^n \rightarrow \Re^m$ *with* L *hidden layers, sigmoidal hidden units and linear output units, such that* $\phi(x) \in U$ *on* $\tilde{C}$*, and for any initial condition* $x_0 \in \tilde{C}$ *the solution of system (3.34) satisfies* $\| x_k \| < \varepsilon$ *for all* $k > T(x_0)$.

Proof: Again this is an immediate consequence of proposition 3.18 and of theorem 2.1 applied to set $\tilde{C}$, function ϕ and constant ε_u.

3.5 Comments

Perhaps the most important, from the point of view of control theory, developments of this chapter are propositions 3.10 and 3.11. They demonstrate that under fairly general assumptions there exists a state feedback mapping which provides an arbitrarily close approximation of the combined task of time optimal control with subsequent stabilization around the target. Then it is shown, that such a feedback mapping can be realized in a simple form of a multilayer perceptron, which for the case of two hidden layers means simply a function that is piecewise constant on polyhedra. These existence results are reassuring from the point of view of a control engineer, whose intuition indicates that, by definition, the system state should represent all the information needed to decide about suitable control action. This is in stark contrast with the continuous time case, where this intuition seems misleading, and the optimal feedback is not always a well defined concept. It seems that the problems with the continuous time case stem in part from insistence that the control be expressed as a function of instantaneous state, which creates problems with definition of resulting system of differential equations, and from too rich a class of admissible control signals considered. On the other hand the control signal in discrete time case corresponds to a simple notion of a numerical sequence, and solution of the associated system of difference equations never fails to exist. Consequently it was possible to obtain all the results using only simple mathematical tools related to the fundamental topological notion of continuity. This suggests that optimal control analysis and synthesis may be much simpler and accessible to a control engineer, if it is performed in discrete, rather than in continuous, time.

The mathematical techniques utilized here were motivated by the work [16] on stabilization of discrete time systems by Heaviside networks. In particular the approximation theorem 2.5 proved therein was used to show existence of suboptimal neural feedback in propositions 3.14 and 3.15. Although these results provide some justification for application of neural networks to synthesis of near-optimal feedback, it has to be cautioned that their value for practical implementations of neural controllers is rather limited. This is because of the already mentioned unavailability of efficient training algorithms for discontinuous multilayered networks. Thus the problem of actual controller synthesis may still be practically untractable, even if existence of appropriate networks is theoretically guaranteed.

Since an overwhelming majority of practical implementations of neural network controllers involves networks with continuous neurons, theoretical results for this architecture are the most important ones from the point of view of engineering applications. Unfortunately, the theory developed here with this respect is somewhat limited. The difficulty lies in synthesizing a continuous selection ϕ of the set valued mapping Φ obtained in propositions 3.12 and 3.13. The only result allowing for such possibility, that the author is aware of, is the Michael's theorem on selections. The condition posed there is that the values of Φ are convex. In the context of this chapter this would mean the requirement that for each point x the set of admissible suboptimal controls is convex. This condition may be satisfied, apart from some very specific cases, only for linear systems, for which use of nonlinear neural control does not make much sense. In the nonlinear case, despite many efforts, the author was unable to isolate a class of systems for which a continuous suboptimal feedback would be guaranteed to exist. This is probably due to the simplicity of mathematical tools used and to efforts to keep the class of the systems considered reasonably general. Using more advanced methods and postulating more properties of the controlled system, it would undoubtedly be possible to show existence of continuous suboptimal feedback. However the question whether the obtained class of systems would correspond to any real world applications remains open. This is certainly an avenue of research to pursue in the future.

Propositions 3.19 and 3.20 provide a limited theoretical justification for use of sigmoidal networks as time-optimal controllers. The main shortcoming is that no information is given on the location of the excluded set of initial conditions $C \setminus \tilde{C}$. The following interpretation of these results may be given. Suppose that the time optimal control problem corresponds to rejection of a disturbance that displaces the system state into a random location in the state space. Suppose furthermore that the disturbance can be modeled as a random variable with the probability distribution that is absolutely continuous with respect to the Lebesgue measure on $\Re^n$ and has a compact support. Then, the results obtained may be rephrased that the probability of the state being transferred in a close-to-optimal manner may be made not less than $1 - \varepsilon$, with ε arbitrarily small. However, this interpretation is valid only for a model of a disturbance as a single random event. Should a sequence of independent disturbances be considered, the guaranteed probability of success will inevitably decrease. Still, for applications when such a disturbance appears rather seldom this seems a valid argument. Usually reliability of the controller hardware is specified in terms of probability of failure in a given time and if this probability is sufficiently low the control system is deemed acceptable. If the probability of control algorithm failing can be made smaller than that for controller hardware, then it may be argued that the algorithm is suitable for the given task. In this sense, the results of section 3.4 justify application of sigmoidal neural networks for suboptimal control. It must be also added, that the obtained continuous controller does not have to fail for some, or even all, initial conditions in the excluded set. Therefore, the actual probability of failure may be smaller, or even zero. In other words, the theory developed here cannot guarantee, but does not exclude the possibility of, continuous neural controllers performing suboptimally for all initial conditions of interest.

The controller structure discussed in this section is that of the state feedback. It may be argued that in practical cases the state measurements are seldom available, and output feedback is the only feasible choice. Unfortunately, construction of time-optimal output feedback is considerably more difficult. The key notion here is that of observability, or ability to reconstruct information about the state from measurements of the system input and output. Even if this is possible in a finite number of steps, there remains an issue how to provide appropriate control action before the state is reconstructed by the controller. For example in case of disturbance rejection, the time needed to bring the state to a vicinity of the equilibrium will depend on the controller actions between the moment when the disturbance occurred and the moment when the state information was reconstructed. There is certainly a need to address those problems and to develop a theory concerning suboptimal neural control for systems with output measurements. As is almost always the case, practice is ahead of theory here, and a successful application of suboptimal output feedback will be discussed in chapter 5.

Chapter 4
Optimal control with summable quality index

The class of optimal control problems considered in this chapter corresponds to the optimized quality criterion being the cost integrated or, in the discrete-time case, summed along the trajectory, with the cost function representing the penalty for deviation of the trajectory from the equilibrium. The most popular example is the quadratic optimal regulator problem, in which the cost function corresponds to a squared norm of deviation from the equilibrium. In the linear case, the problem has been extensively studied and in particular, synthesis of optimal control is readily available both in continuous and discrete time cases [32]. For continuous time nonlinear systems, existence of the optimal feedback can be established through dynamic programming approach, subject however to quite stringent smoothness requirements with respect to the system and integrated quality criterion [33]. The actual synthesis however, requires knowledge of the analytical form of the model and involves solution of complicated systems of partial differential equations. As in the previous chapter it is suggested that artificial neural networks be used to synthesize an approximately optimal feedback controller, based on data contained in a number of open-loop optimal trajectories. This approach allows to replace the difficult task of optimal synthesis with a number of easier tasks of open loop optimizations, for which efficient numerical techniques exist.

The optimal regulation problem will be studied in discrete-time setting. As in the previous chapter, this introduces more regularity into the problem and allows to avoid difficulties with existence and uniqueness of solutions of the associated sets of equations. Also the mathematical tools that will be used, are considerably simpler than it would be necessary in continuous-time case. Since only fundamental continuity properties will be utilized, it is possible not to require almost any smoothness of the system and the cost function - only local Lipschitz continuity of the system model in a small neighborhood of the equilibrium will be used in a portion of this chapter.

The theory developed in this chapter provides foundations for the approximation of optimal feedback control by artificial neural network. First the existence of measurable optimal feedback is established for a suitably defined relaxed optimization problem. Then approximation of such a feedback by artificial neural networks is studied. The obtained results are purely existential, that is they establish that the set of neural network controllers providing suboptimal control is not empty. As remarked earlier, this is of paramount importance for the proposed approach to have sense. Once the existence problem is solved, synthesis of the controller, or its training, may be undertaken. The issue of practical implementation of the synthesis process is not, however, addressed here.

4.1 Problem statement

The nonlinear finite-dimensional dynamical system is assumed to admit a model of the form (3.5). Accordingly, all notations introduced in section 3.1.2 will be used here as needed. Also assumptions 3.1 through 3.6 will be referred to.

The quality criterion associated with the optimization problem is in the form of a sum of costs corresponding to consecutive points of the controlled trajectory. Consider any initial condition $x_0 \in \Re^n$ and any finite sequence of admissible control values, $\{u_i\}_{i=0}^{k-1}$, $u_i \in U$, together with the corresponding state trajectory

$$x_k = f^k(x_0, \{u_i\}_{i=0}^{k-1}) \tag{4.1}$$

The cost associated with each individual element of the trajectory is assumed to take the form of a nonnegative valued function $h : \Re^n \to \Re_+ \cup \{0\}$. The cost associated with the entire trajectory (4.1) can be expressed as a function of the initial condition and the control signal, $q_u^k : \Re^n \times U^k \rightarrow \Re_+ \cup \{0\}$ of the form

$$q_u^k(x_0, \{u_i\}_{i=0}^{k-1}) = \sum_{i=1}^{k} h(f^k(x_0, \{u_i\}_{i=0}^{k-1}) \tag{4.2}$$

or alternatively, as a function of the elements of the trajectory, $q_s^k : (\Re^n)^k \rightarrow \Re_+ \cup \{0\}$

$$q_s^k(\{x_i\}_{i=1}^{k}) = \sum_{i=1}^{k} h(x_k) \tag{4.3}$$

For both forms of the above cost function the same symbol q^k will be used, as it will usually be obvious whether the independent variable is the control signal or the elements of the trajectory.

Consider the target set $A \subset \Re^n$, controllable to itself in one step, $A \subset \mathcal{C}^1(A)$, so that once A is reached it is possible for the state to remain in A. Suppose that the initial condition x_0 is controllable to set A in k steps, $x \in \mathcal{C}^k(A)$. Then the optimal cost corresponding to control horizon k, and target set A may be expressed as a function of the initial condition, $J_A^k : \mathcal{C}^1(A) \rightarrow \Re_+ \cup \{0\}$, defined as follows

$$J_A^k(x_0) = \inf\{q^{k-1}(x_0, \{u_i\}_{i=0}^{k-2}) : \ u_i \in U, \ x_k = f^k(x, \{u_i\}_{i=0}^{k-1}) \in A\} \tag{4.4}$$

The infinum is taken over all admissible control sequences transferring x_0 into A in k steps. Note that costs of neither the initial condition x_0, nor the final point x_k are included in the minimized cost. Since x_0 is given and cannot be changed by the subsequent controls, its inclusion in the quality criterion will not change the solution. On the other hand, it is assumed that all final points in set A are treated equivalently, and as long as A is reached at moment k, there is no need to differentiate between final points in A by inclusion of x_k in cost q^k. Therefore the last term in the minimized sum (4.2) is $h(x_{k-1})$, rather than $h(x_k)$. To assure continuity of function q^{k-1} with respect to control values, the cost of each trajectory element x_i, $i < k$, is added to the summed cost, even if $x_i \in A$. That may seem unreasonable at first glance, and in conflict with the preceding argument for exclusion of cost of the final point $h(x_k)$ from the total cost. However, while considering trajectories minimizing (4.2), the solutions of interests will satisfy $x_i \notin A$ for $i < k$, thus eliminating this minor inconvenience. The alternative formulation of the quality criterion would be to multiply individual costs $h(x_i)$ by the characteristic function of the complement of A, so that only elements outside the target set would contribute to the cost of the trajectory. An advantage of such an approach would be that the optimal costs (4.5) corresponding to different control horizons k would form a nonincreasing sequence, $J_A^{k+1} \leq J_A^k$, provided that $A \subset \mathcal{C}^1(A)$. However, this approach would result in larger problems due to possible discontinuities of summed cost (4.2) with respect to control values.

The infinite horizon optimal cost corresponding to target set A, is a function of initial condition, $J_A : \mathcal{C}(A) \to \Re_+ \cup \{0\}$, defined by

$$J_A(x_0) = \inf\{J_A^k(x_0) : \; x_0 \in \mathcal{C}^k(A)\} \tag{4.5}$$

where the infinum is taken here over all finite control horizons k, for which the initial condition is controllable to A. Note that (4.5) is well defined for all points controllable to A.

The class of optimal control problems considered in this chapter corresponds to the target set being the origin of the state space, $A = \{0\}$. The earlier assumption that $f(0,0) = 0$ assures that the state can be maintained at the origin, once it is reached. The set of initial conditions C is assumed to be compact, and controllable to the origin, $C \subset \mathcal{C}(\{0\})$, so that the transfer to the origin is a feasible task. As discussed before, this actually implies that $C \subset \mathcal{C}^K(\{0\})$ for some finite K, so that every point in C is controllable to the origin in at most K steps. The optimal cost functions for finite and infinite horizons, $J^k : \mathcal{C}^k(\{0\}) \to \Re_+ \cup \{0\}$ and $J : \mathcal{C}(\{0\}) \to \Re_+ \cup \{0\}$ are defined as

$$J^k(x) = J_{\{0\}}^k(x) \tag{4.6}$$

$$J(x) = J_{\{0\}}(x) \tag{4.7}$$

The so far formulated class of optimal control problems includes also time optimal control if the cost function is $h(x) = 1$. The problems considered here, however, are associated with functions $h : \Re^n \to \Re_+ \cup \{0\}$ satisfying the following assumptions.

- $h(x) = 0$ if and only if $x = 0$
- h is continuous
- for any real number $\alpha > 0$, the set $Q_\alpha = \{x \in \Re^n : \ h(x) < \alpha\}$ is bounded

A typical example of a cost function satisfying these assumptions is a quadratic function $h(x) = x^T Q x$, with Q a positive definite matrix. This kind of quality criterion is commonly used in linear control theory. Also any norm on $\Re^n$ may serve as h.

Observe that with h continuous, the summed cost of a given trajectory (4.2) is also a continuous function of the argument $\{u_i\}_{i=0}^{k-1}$. Since for a given x_0 the set of admissible control signals transferring x_0 to the origin is a compact subset of U^k, it follows that the minimal value $J^k(x_0)$ of (4.2) is actually achieved for some control signal from U^k. This is not necessarily true for the infinite horizon cost J, and may happen that the infinum (4.5) is not achieved for any finite k. However for any x it is possible to find a finite control horizon k such that the achievable minimal cost (4.4) is as close to infinite horizon cost (4.5) as desired.

For the purpose of feedback synthesis, a relaxed optimal control problem will be defined for some positive real numbers ε^s and ε^j. A suboptimal control sequence will be sought, such that both the minimum cost and the target set are achieved approximately, that is

$$q^{k-1}(x_0, \{u_i\}_{i=0}^{k-2}) < J(x_0) + \varepsilon^j \tag{4.8}$$

$$x_k = f^k(x_0, \{u_i\}_{i=0}^{k-1}) \in \mathcal{B}^n(0, \varepsilon^s) \tag{4.9}$$

for some finite $k > 0$. The following analysis will be concerned with state feedback controllers of the form $\phi : \Re^n \to U$, such that the resulting control signal

$$u_i = \phi(x_i) \tag{4.10}$$

satisfies (4.8) and (4.9). An additional requirement will also be considered, that the state remains in the specified neighborhood of the origin after it has been reached, that is

$$x_i \in \mathcal{B}^n(0, \varepsilon^s) \quad \text{for} \quad i > k \tag{4.11}$$

Thus the suboptimal control will be combined with approximate stabilization around the origin. Existence of controllers satisfying the above specifications will be studied for arbitrarily small ε^j and ε^s.

4.2 Properties of approximate cost function

To study suboptimal control by artificial neural networks, the approximation results discussed in chapter 2 will be used. It will be desirable, that small deviations from the initial condition and the corresponding optimal control signal result in the cost still close to the optimal one. The ideal situation would be, if the optimal cost were a continuous function of the initial condition. Unfortunately, with the target set being a single point, this need not be the case. Thus, attempts to approximate the optimal cost (4.7) directly, as specified by condition (4.8), may lead to some technical problems. It is more convenient to introduce an approximate cost J_{A_0}, corresponding to some neighborhood A_0 of the original target, and then to seek feedback strategies approximating J_{A_0}. If the original and the relaxed optimal costs are close to each other, this approach will allow satisfaction of suboptimality conditions (4.8) and (4.9).

Consider a bounded target set containing a neighborhood of the origin. It will be shown that the sequence of optimal costs associated with increasing control horizons will actually achieve minimum for some finite horizon. The following lemma is concerned with this property.

Lemma 4.1 *Suppose that system (3.5) satisfies assumptions 3.1 and 3.2. Let C be a compact set of initial conditions controllable to the origin in at most K steps, $C \subset \mathcal{C}^K(\{0\})$, and let A_0 be a subset of C, with the origin contained in its interior, $0 \in int(A_0) \subset A_0 \subset C$. Then there exists an integer K_J, such that for any $x \in C$,* *the optimal cost associated with infinite horizon is equal to the optimal cost for some finite horizon $l < K_J$, that is*

$$J_{A_0}(x) = J^l_{A_0}(x) \tag{4.12}$$

or, equivalently, for any $i \geq 0$

$$J_{A_0}(x) \leq J^{K_J+i}_{A_0}(x) \tag{4.13}$$

Proof: First observe that the optimal cost to reach the origin, associated with a finite horizon k, $J^k(x)$, is bounded on C for every $k > K$. For every horizon $j > 0$, denote by H_j the maximal possible value of function h on the set reachable from C in j steps

$$H_j = \max\{h(x) : \ x \in f^j(C, U)\} \tag{4.14}$$

Then, since any state $x \in C$ is controllable to the origin in K steps, it follows that the optimal cost associated with any horizon $j > K$ is bounded by

$$J(x) \leq J^k(x) \leq H = \sum_{k=1}^{K-1} H_j \tag{4.15}$$

Consequently, also the optimal costs to reach A_0 are bounded on the set C by the number H

$$J_{A_0}(x) \leq J^{A_0}_k(x) \leq J^k(x) \leq H \tag{4.16}$$

Now consider the lower bound of the continuous function h on the set $C \cap A_0^c$

$$\gamma = \inf\{h(x) : \ x \in C \cap A_0^c\} \tag{4.17}$$

This value is positive because of assumed continuity and positive definiteness of h, and because the interior of A_0 contains the origin. Next, choose an integer K_J such that

$$K_J > \max\{\frac{H}{\gamma}, K\} \tag{4.18}$$

Take an arbitrary $\varepsilon > 0$, and $x \in C$, and consider optimal cost $J^k_{A_0}(x)$ to reach A_0 for some finite horizon $k > K_J$. There exists a control sequence $\{u_i\}_{i=0}^{k-1}$, transferring the state to A_0,

$$f^k(x, \{u_i\}_{i=0}^{k-1}) \in A_0 \tag{4.19}$$

and such that the associated cost is close to the optimal value

$$q^{k-1}(x, \{u_i\}_{i=0}^{k-2}) < J_{A_0}^k(x) + \varepsilon \tag{4.20}$$

Suppose that of k points of the corresponding trajectory, $x_j = f^j(x, \{u_i\}_{i=0}^{j-1})$, $j = 1, \dots, k$, exactly $l \leq k$ lie outside A_0. But because of (4.17) these elements of the trajectory satisfy $h(x_j) \geq \gamma$, and it follows that

$$q^{k-1}(x, \{u_i\}_{i=0}^{k-2}) \geq l\gamma \tag{4.21}$$

In view of (4.16) and (4.20), this implies that $l\gamma \leq H + \varepsilon$. Since ε is an arbitrarily small number, it follows from (4.18) that $l \leq \frac{H}{\gamma} \leq K_j$. Therefore, there exists an integer $l^\star < K_J$ such that

$$f^{l^\star}(x, \{u_i\}_{i=0}^{l^\star-1}) \in A_0 \tag{4.22}$$

which in turn means that the optimal cost associated with horizon $l^\star$ satisfies

$$J_{A_0}^{l^\star}(x) \leq q^{l^\star-1}(x, \{u_i\}_{i=0}^{l^\star-2}) < J_{A_0}^k(x) + \varepsilon \tag{4.23}$$

Because this was shown for an arbitrary $\varepsilon > 0$ it follows that

$$J_{A_0}^{l^\star}(x) \leq J_{A_0}^k(x) \tag{4.24}$$

This concludes the proof.

The significance of the above lemma is that it will be possible to limit the considered optimal and suboptimal transitions to some finite number of steps. This will make it possible to cover the set of initial conditions with finitely many subsets, similarly to the previous developments concerning stabilization and time optimal control. The next lemma is concerned with approximation of the infinite horizon 'true' optimal cost $J(x)$ by the finite horizon 'relaxed' optimal cost $J_{A_0}^k(x)$.

Lemma 4.2 *Suppose that system (3.5) satisfies assumptions 3.1 and 3.2. Let A_0 be an open and bounded neighborhood of the origin, $k > 0$ some finite control horizon, and C a compact set of initial conditions controllable to the origin in at most k steps,*

$C \subset \mathcal{C}^k(\{0\})$. Then there exists an integer K_J, such that for any $x \in C$, and for any $\varepsilon > 0$ it is possible to choose a $\delta > 0$ and a control horizon $l < K_J$, for which the cost J is approximated by $J^l_{A_0}$ on $\mathcal{B}^n(x, \delta)$, in the sense that

$$\text{if } \ \tilde{x} \in \mathcal{B}^n(x, \delta) \cap C \ \text{ then } \ J(\tilde{x}) > J^l_{A_0}(x) - \varepsilon \tag{4.25}$$

Proof: To show that the assertion of the lemma holds, suppose, to the contrary, that it is false. That is, assume that there exist some $x \in C$ and $\varepsilon > 0$, such that for any open neighborhood of x, $\mathcal{B}^n(x, \delta)$ such that $C \cap \mathcal{B}^n(x, \delta) \neq \{x\}$, and for any optimization horizon l, there exists at least one point $\tilde{x} \in C \cap \mathcal{B}^n(x, \delta)$ satisfying

$$J(\tilde{x}) \leq J^l_{A_0}(\tilde{x}) - \varepsilon \tag{4.26}$$

This assumption will be shown contradictory. First, consider another open bounded neighborhood of the origin $\tilde{A}$, whose closure $\hat{A}$ is contained in A_0

$$0 \in \tilde{A} \subset \hat{A} = \text{clos}(\tilde{A}) \subset A_0 \tag{4.27}$$

Making use of lemma 4.1 twice, choose a sufficiently large control horizon K_J, such that if $\tilde{x} \in C$ and $j > K_J$, then

$$J_{A_0}(\tilde{x}) \leq J^j_{A_0}(\tilde{x}) \tag{4.28}$$

$$J_{\tilde{A}}(\tilde{x}) \leq J^j_{\tilde{A}}(\tilde{x}) \tag{4.29}$$

For the chosen $x \in C \subset \mathcal{C}^k(\{0\})$, it is possible to find $l < K_J$, such that the associated optimal cost satisfies

$$J_{A_0}(x) = J^l_{A_0}(x) \tag{4.30}$$

If the assertion of the lemma is false, then it is possible to find a sequence of points $\{\tilde{x}_i\}_{i=1}^{\infty}$ convergent to x, $\lim_{i \to \infty} \tilde{x}_i = x$, such that

$$J(\tilde{x}_i) \leq J^l_{A_0}(x) - \varepsilon \tag{4.31}$$

For every element $\tilde{x}_i$ of the sequence it is possible to find a finite horizon j_i such that

$$J^{j_i}(\tilde{x}_i) \leq J(\tilde{x}_i) + \frac{\varepsilon}{3} \tag{4.32}$$

Then consider optimal transfer of points $\tilde{x}_i$ to the open set $\tilde{A}$. Since the origin is an element of $\tilde{A}$, the optimal cost $J^{j_i}_{\tilde{A}}$ associated with target set $\tilde{A}$ and horizon j_i satisfies

$$J^{j_i}_{\tilde{A}}(\tilde{x}_i) \leq J^{j_i}(\tilde{x}_i) \tag{4.33}$$

By lemma 4.1 and by choice of K_J, it is possible to find for each $\tilde{x}_i$ a control horizon $l_i < K_J$ such that

$$J_{\tilde{A}}(\tilde{x}_i) = J^{l_i}_{\tilde{A}}(\tilde{x}_i) < J^{j_i}_{\tilde{A}}(\tilde{x}_i) < J(\tilde{x}_i) + \frac{\varepsilon}{3} \leq J^{l}_{A_0}(x) - \frac{2}{3}\varepsilon \tag{4.34}$$

where the second inequality is a consequence of (4.32), and the last one follows from (4.31). Choose a limit point $l^\star \leq K_J$ of the bounded sequence of integers $\{l_i\}_{i=1}^{\infty}$, and pick a subsequence of points $\tilde{x}_i$, for which the associated control horizon l_i satisfies $l_i = l^\star$. After necessary renumbering, denote the new sequence convergent to x as $\{\hat{x}_i\}_{i=1}^{\infty}$. For each element of this sequence it is now possible to construct a finite control sequence of length $l^\star$, $\{\hat{u}_{i,j}\}_{j=0}^{l^\star-1}$, such that the state is transferred to $\tilde{A}$

$$f^{l^\star}(\hat{x}_i, \{\hat{u}_{i,j}\}_{j=0}^{l^\star-1}) \in \tilde{A} \tag{4.35}$$

and the incurred cost is close to the optimal one

$$q^{l^\star-1}(\hat{x}_i, \{\hat{u}_{i,j}\}_{j=0}^{l^\star-2}) < J^{l^\star}_{\tilde{A}}(\hat{x}_i) + \frac{\varepsilon}{3} \leq J^{l}_{A_0}(x) - \frac{\varepsilon}{3} \tag{4.36}$$

with the second inequality following from (4.34). Since the control values are bounded, it is possible to choose yet another subsequence $\{\bar{x}_i\}_{i=0}^{\infty}$ convergent to x, such that for each $0 \leq j < l^\star$ the corresponding sequence of controls $\{\bar{u}_{i,j}\}_{i=1}^{\infty}$ converges to some $u^\star_j \subset U$. From (4.35), and by continuity of the state transition function f, it follows that

$$f^{l^\star}(x, \{u^\star_j\}_{j=0}^{l^\star-1}) \in \operatorname{clos}(\tilde{A}) \subset A_0 \tag{4.37}$$

Then, (4.36) together with continuity of cost function h imply that control sequence $\{u^\star_j\}_{j=0}^{l^\star-1}$ transfers x to A_0 with the incurred cost satisfying

$$q^{l^\star-1}(x, \{u^\star_j\}_{j=0}^{l^\star-2}) \leq J^{l}_{A_0}(x) - \frac{\varepsilon}{3} \tag{4.38}$$

Consequently the optimal cost associated with horizon $l^\star$ for the chosen point x satisfies

$$J_{A_0}^{l^\star}(x) < J_{A_0}^{l}(x) \tag{4.39}$$

However, this contradicts (4.30), which implies that l is the horizon allowing for the minimal cost of transfer to A_0. Hence, the assertion of the lemma cannot be false, which completes the proof.

The above lemma will be crucial for construction of optimal trajectories, as it allows to replace the task of minimization of infinite horizon cost $J(\tilde{x})$ by minimization of finite horizon cost $J_{A_0}^{l}(x)$ in an open neighborhood x. The obtained neighborhoods will then be utilized to cover the compact set of initial conditions and the associated control values will be used to construct a suboptimal feedback mapping.

4.3 Existence of suboptimal feedback

Developments of the previous section will allow consideration of only finite length approximations of optimal trajectories on open sets. Then by covering the compact set of initial conditions C with finitely many open sets, it will be possible to partition C into finitely many subsets on which the desired suboptimal control strategy will be constant. To this end, first a locally suboptimal feedback will be considered for an open target set. The following lemma is concerned with existence of such a mapping defined for a small open set of initial conditions.

Lemma 4.3 *Suppose that system (3.5) satisfies assumptions 3.1 and 3.2. Let C be a compact set of initial conditions controllable to the origin in finitely many steps, $C \subset \mathcal{C}^k(\{0\})$, and let A_0 be an open neighborhood of the origin, controllable to itself in one step, $0 \in A_0 \subset \mathcal{C}^1(A_0)$. Then there exists an integer $K_J > 0$, such that for any $\varepsilon > 0$ and for any $x \in C$ there exists an integer $l_x < K_J$, a collection of $l_x + 1$ mutually disjoint open sets $V_{x,i} \subset \Re^n$, and a corresponding collection of open sets of control values $U_{x,i} \subset U$, satisfying $x \in V_{x,0}$ and $V_{x,l_x} \subset A_0$, and such that if $\tilde{x} \in V_{x,i}$*

and $\tilde{u} \in U_{x,i}$, *then*

$$f(\tilde{x}, \tilde{u}) \in V_{x,i+1} \tag{4.40}$$

if $\tilde{x} \in V_{x,l_x}$ *and* $\tilde{u} \in U_{x,l_x}$ *then*

$$f(\tilde{x}, u_{x,l_x}) \in A_0 \tag{4.41}$$

and that the cost of transfer of any $\tilde{x} \in V_{x,i}$, $i = 0, \ldots, l_x - 1$, *to* A_0 *using arbitrary controls* $\tilde{u}_j \in U_{x,j}$, $j = i, \ldots, l_x - 1$ *satisfies*

$$q^{l_x-i-1}(\tilde{x}, \{\tilde{u}_j\}_{j=i}^{l_x-2}) < J(\tilde{x}) + \varepsilon \tag{4.42}$$

Proof: The proof is a straightforward consequence of continuity of functions f and h and of lemma 4.2. The first step is to use that result to choose the control horizon K_J, which will limit the length of all considered control sequences. Then, pick an arbitrary $x \in C$ and, again by virtue of lemma 4.2 choose a control horizon $l_x < K_j$, which is the optimal horizon for transfer of x to A_0, so that

$$J_{A_0}^{l_x}(x) = J_{A_0}(x) \tag{4.43}$$

Next, an associated control sequence $\{u_{x,i}\}_{i=0}^{l_x-1}$ of length l_x is constructed, such that it transfers x to A_0

$$f^{l_x}(x, \{u_{x,i}\}_{i=0}^{l_x-1}) \in A_0 \tag{4.44}$$

with the incurred cost sufficiently close to the optimal cost $J_{A_0}^{l_x}$

$$q^{l_x-1}(x, \{u_{x,i}\}_{i=0}^{l_x-2}) < J_{A_0}^{l_x}(x) + \frac{\varepsilon}{3} \tag{4.45}$$

Because of the assumption that $U = \text{clos}(\text{int}(U))$, the control values may always be chosen so that $u_{x,i} \in \text{int}(U)$. For each element of the corresponding state trajectory, $x_i = f^i(x, \{u_{x,j}\}_{j=0}^{i-1})$, it follows from (4.45) and from additivity of the cost function q, that the cost associated with the remaining trajectory portion satisfies

$$q^{l_x-i-1}(x_i, \{u_{x,j}\}_i^{l_x-2}) < J_{A_0}^{l_x-i} + \frac{\varepsilon}{3} \tag{4.46}$$

It is also easily seen that $l_x - i$ is the optimal control horizon for transfer of the point x_i to the target set A_0

$$J_{A_0}^{l_x-i}(x_i) = J_{A_0}(x_i) \tag{4.47}$$

Now, the two required collections of open sets $V_{x,i}$ and $U_{x,i}$ will be defined for $i = 0, \ldots, l_x$ in the recursive manner. For $i = l_x$ a neighborhood $V_{x,l_x} \subset A_0$ of the point $x_{l_x} \in A_0$ and an open set U_{x,l_x} are chosen so that (4.41) is satisfied. This is possible to achieve by openness of A_0, and by assumption that $A_0 \subset \mathcal{C}^1(A_0)$. If $V_{x,i+1}$ has been already defined for some $i < l_x$, then a neighborhood $V_{x,i}$ of x_i and a neighborhood $U_{x,i}$ of $u_{x,i}$ can be chosen so that (4.40) is satisfied. In addition, the choice of $V_{x,i}$ can always be such that if $\tilde{x} \in V_{x,i}$, then

$$h(\tilde{x}) < h(x_i) + \frac{\varepsilon}{3l_x} \tag{4.48}$$

$$J(\tilde{x}) > J_{A_0}^{l_x-i}(\tilde{x}) - \frac{\varepsilon}{3} \tag{4.49}$$

These are possible to satisfy because of continuity of the cost function h and by application of lemma 4.2 in view of (4.47). It may be also assumed that sets $V_{x,i}$ are mutually disjoint, as $x_i \neq x_j$ for $i < j$. Should it not be the case, the whole trajectory and the control sequence could be reduced, by eliminating points $i+1, \ldots, j$, thus resulting in a lower transfer cost for control horizon $l_x^\star = l_x - (j-i) < l_x$, which would contradict the choice of l_x as the optimal control horizon for x. To see that the defined sets $V_{x,i}$ and $U_{x,i}$ satisfy property (4.42), consider some initial condition $\tilde{x}_0 \in V_{x,k}$, for some $0 \le k < l_x$, and control sequence $\{\tilde{u}_j\}_{j=0}^{l_x-k-1}$ such that $\tilde{u}_j \in U_{x,k+j}$. Consider the resulting trajectory, which has been demonstrated to satisfy

$$\tilde{x}_j = f^j(\tilde{x}_0, \{\tilde{u}_i\}_{i=0}^{j-1}) \in V_{x,k+j} \tag{4.50}$$

Then by (4.48) it follows that the incurred cost of the trajectory satisfies

$$q^{l_x-k-1}(\tilde{x}, \{\tilde{u}_j\}_{j=0}^{l_x-k-2}) = \sum_{j=1}^{l_x-k-1} h(\tilde{x}_j) < \sum_{j=1}^{l_x-k-1} h(x_{k+j}) + \frac{\varepsilon}{3} \tag{4.51}$$

The sum on the right hand side of (4.51) is equal to $q^{l_x-k-1}(x_k, \{u_j\}_{j=0} l_x - k - 2$. Therefore it is possible to combine the above inequality with (4.46) to obtain

$$q^{l_x-k-1}(\tilde{x}, \{\tilde{u}_j\}_{j=0}^{l_x-k-2}) < J_{A_0}^{l_x-k}(x) + \frac{2\varepsilon}{3} \tag{4.52}$$

which in view of (4.49) leads to the conclusion that

$$q^{l_x-k-1}(\tilde{x}, \{\tilde{u}_j\}_{j=0}^{l_x-k-2}) < J(\tilde{x}) + \varepsilon \tag{4.53}$$

This concludes the proof.

The above lemma yields for every $x \in C$ a local suboptimal feedback mapping $\phi_x : V_x \to U$, defined on the open set $V_x = \bigcup_{i=0}^{l_x} V_{x,i}$ by

$$\phi_x(\xi) = \sum_{i=0}^{l_x} u_{x,i}\chi_{V_{x,i}}(\xi) \tag{4.54}$$

so that $\phi_x(\xi) = u_{x,i}$ for all $\xi \in V_{x,i}$. For any initial condition in V_x the solution of the closed loop system with feedback ϕ_x reaches A_0 in finitely many steps with almost optimal cost. It is not assured, however, that the trajectory will remain in A_0, as it may leave V_{x,l_x} outside which the feedback ϕ_x is not defined. To eliminate this shortcoming, the local controllability assumption will be used, which will allow to construct a suitable target set A_0 invariant under zero control. With this possibility utilized, it will be possible to establish the first result of this section, which guarantees existence of a measurable feedback mapping that provides suboptimal control.

Proposition 4.4 *Suppose that system (3.5) satisfies assumptions 3.1, 3.2, 3.4 and 3.5. Let C be a compact set of initial conditions controllable to the origin in finitely many steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$ and $\varepsilon_q > 0$ be arbitrary numbers. Then there exists a measurable state feedback function $\phi : \Re^n \to U$, and an integer K_J, such that for any initial condition $x_0 \in C$ the solution of the closed-loop system reaches the target sets $\mathcal{B}^n(0, \varepsilon_s)$ in finitely many, $k_x < K_J$, steps incurring cost*

$$q^{k_x-1}(\{x_i\}_{i=1}^{k_x-1}) < J(x_0) + \varepsilon_q \tag{4.55}$$

and the trajectory satisfies $x_i \in \mathcal{B}^n(0, \varepsilon_s)$ for all $i \geq k_x$. In addition the feedback mapping ϕ may be chosen to take finitely many values on C.

Proof: The first step consists of invoking lemma 3.7 to construct a sufficiently small open set $A_0 \subset \mathcal{B}^n(0, \varepsilon_s)$, which is controllable to itself, that is $A_0 \subset \operatorname{clos}(A_0) \subset$

$\mathcal{C}^1(A_0)$. It will also be required that any state in A_0 be transferable to the origin with cost less than $\frac{\varepsilon}{2K_J}$. This is always possible to achieve by choosing A_0 small enough, because of assumptions 3.4 and 3.5, and from continuity of h (the detailed proof would follow the lines of proof of lemma 3.7). Then, suboptimal transfer to A_0 rather than to $\mathcal{B}^n(0,\varepsilon_q)$ will be considered. Next, find the limiting control horizon K_J corresponding to the target set A_0 using lemma 4.1. Then set $\varepsilon = \frac{\varepsilon_q}{2K_J}$ and invoking lemma 4.3 for every $x \in C$, construct open sets $V_{x,j} \subset \Re^n$ and $U_{x,j} \subset U$, $j = 0 \leq j \leq l_x < K_j$, such that conditions (4.40-4.42) are satisfied. Also, from each $U_{x,j}$ pick an arbitrary control value $u_{x,j}$. Note that sets $V_{x,0}$ form an open covering of C, and also of $\text{clos}(A_0)$. Therefore it is possible to choose a finite subcover $V_{i,0}$, $i = 1, \ldots, N$, so that

$$C \subset \bigcup_{i=1}^{N} V_{i,0} \tag{4.56}$$

To each of open sets $V_{i,0}$ there corresponds a point $x_i \in V_{i,0}$, a finite horizon l_i, a control sequence $\{u_{i,j}\}_{j=0}^{l_i-1}$, and open sets $V_{i,j}$, $j = 1, \ldots, l_i - 1$, satisfing the properties (4.40-4.42). It is possible to construct the covering $V_{i,0}$, in such a manner that $\text{clos}(A_0)$ is covered by sets corresponding to points $x_i \in \text{clos}(A_0)$

$$\text{clos}(A_0) \subset \bigcup_{i:\ x_i \in \text{clos}(A_0)} V_{i,0} \tag{4.57}$$

Since all control horizons l_i are bounded by K_J, it is possible to arrange all sets $V_{i,j}$ into K_J finite collections, corresponding to different control horizons necessary to reach the target set A_0 in a close to optimal manner. That is, the k-th collection consists of all sets $V_{i,j}$ for which it holds that $l_i - j = k$. Assume the numbering of sets $V_{i,0}$ such that the corresponding control horizons l_i form a nonincreasing sequence, that is $l_{i+1} \leq l_i$. Denote by M_k the number of sets $V_{i,0}$ such that $l_i \geq k$, so that $0 \leq M_{K_J} \leq \ldots \leq M_1 = N$, that is the sets for which $l_i = k$ correspond to indices $i = M_{k+1} + 1, \ldots, M_k$. This ordering induces also similar classification among sets $V_{i,j}$, $j > 0$. For each j there are M_{j+1} sets $V_{i,j}$ associated with the constructed covering, and those for which the length of transfer to A_0 is $l_i - j = k$ correspond to indices $i = M_{k+j+1} + 1, \ldots, M_{k+j}$. Now define unions of sets corresponding to the same length of transfer to A_0

$$A_k = \bigcup_{i=1}^{M_k} V_{i,l_i-k} = \bigcup_{j=0}^{K_J-1} \bigcup_{i=M_{k+j+1}+1}^{M_{k+j}} V_{i,j} \tag{4.58}$$

Thus constructed open sets A_k cover the compact set C

$$C \subset \bigcup_{k=1}^{K_J} A_k \tag{4.59}$$

Recall that for all $x \in \mathrm{clos}(A_0)$ the corresponding optimal horizon satisfies $l_x = 1$. Therefore it follows from (4.57) that $\mathrm{clos}(A_0)$ is covered with sets V_{i,l_i-1} and consequently

$$\mathrm{clos}(A_0) \subset A_1 \tag{4.60}$$

For each A_k, $0 < k \leq K_J$, a collection of mutually disjoint sets $W_{k,i} \subset A_k$, $i = 1, \ldots, M_k$ is defined as

$$W_{k,i} = (V_{i,l_i-k} \setminus (\bigcup_{j=1}^{i-1} V_{k,j})) \setminus (\bigcup_{j=1}^{k-1} A_j) \tag{4.61}$$

Thus defined, sets W_{k_1,j_1} and W_{k_2,j_2} are also disjoint if $k_1 \neq k_2$. The desired suboptimal feedback mapping $\phi : \Re^n \rightarrow U$ is defined as

$$\phi(x) = \sum_{k=1}^{K_j} \sum_{i=1}^{M_k} u_{i,l_i-k} \chi_{W_{k,i}}(x) \tag{4.62}$$

Observe, that if $x \in A_0$, then necessarily $x \in W_{1,i} \subset V_{i,l_i-1}$, and $\phi(x) = u_{i,l_i-1}$ for some $1 \leq i \leq M_1 = N$, and it follows that $f(x, \phi(x)) \in V_{i,l_i} \subset A_0$. Thus A_0 is invariant under feedback ϕ. Consider therefore an arbitrary $x_0 \in C \setminus A_0$. To see that feedback ϕ provides transfer of x to A_0 in finitely many steps, suppose that $\xi_0 \in W_{k_0,i_0} \subset V_{i_0,l_{i_0}-k}$ and $\phi(\xi_0) = u_{i_0,l_{i_0}-k_0}$, for some $1 \leq k \leq K_j$, $1 \leq j_0 \leq N_{k_0}$. From properties of sets $V_{i,j}$, established in lemma 4.3, it follows that the next element of the trajectory satisfies

$$\xi_1 = f(\xi_0, \phi(\xi_0)) \in V_{i,l_i-k_0+1} \subset A_{k_0-1} \subset \bigcup_{j=0}^{k_0-1} A_j \tag{4.63}$$

If $k_0 = 1$, then the state x_1 has reached A_0, and, as concluded earlier, the entire trajectory ξ_i will remain therein for $i > 1$. Suppose therefore that $k_0 > 1$. Then from the construction of sets $W_{k,j}$ it is seen that $\xi_1 \in W_{k_1,i_1}$, where

$$k_1 = \min\{k : \xi_1 \in A_k\} < k_0 \tag{4.64}$$

for some $1 \leq i_1 \leq M_{k_1}$. By recursive application of the above argument it is seen that the solution of the closed loop system enters $A_0 \subset B$ after some $\tilde{k} \leq k_0$ steps. Once in A_0, the state satisfies $x_i \in A_0 \subset B$, for $i > \tilde{k}$.

The remaining part of the proof is to show that the incurred cost of the transfer to A_0 is a close approximation of the optimal cost corresponding to transfer to the origin. To show this, consider a closed loop trajectory originating outside A_0, $\xi_i \in W_{k_i,j}$, $i = 0, \ldots, l-1$, $\xi_l \in A_0$. As has been demonstrated, indices k_i form a strictly decreasing sequence. It will be shown recursively that for each i the remaining portion of the trajectory incurs the cost which is close to the optimal one, namely

$$q^{l-i-1}(\{\xi_j\}_{j=i+1}^{l-1}) < J(\xi_i) + \frac{(l-i)\varepsilon}{K_J} \tag{4.65}$$

For $i = l-1$ this is trivially true, as the remaining portion of the trajectory consist of a single point $\xi_l \in A_0$ and the incurred cost is zero. Suppose therefore that (4.65) has been shown true up to some $i+1$, and consider transfer of point $\xi_i \in W_{k_i,j} \subset V_{j,l_j-k_i}$ to A_0. as sets V_{j,l_j-k_i} were constructed using lemma 4.3, it follows that the sequence of controls $\{u_{j,p}\}_{p=l_j-k_i}^{l_j-1}$ associated with those open sets satisfies

$$J_{A_0}(\xi_i) \leq q^{k_i-1}(\xi_i, \{u_{j,p}\}_{p=l_j-k_i}^{l_j-2}) < J(\xi_i) + \frac{\varepsilon}{2K_J} \tag{4.66}$$

Similarly, the sequence of controls $\{u_{j,p}\}_{p=l_j-k_i+1}^{l_j-1}$ applied to point $\xi_{i+1} = f(\xi_i, u_{j,l_j-k_i})$ results in a transfer cost which satisfies

$$J_{A_0}(\xi_{i+1}) \leq q^{k_i-2}(\xi_{i+1}, \{u_{j,p}\}_{p=l_j-k_i+1}^{l_j-1}) < J(\xi_{i+1}) + \frac{\varepsilon}{2K_J} \tag{4.67}$$

The above inequalities allow to estimate the portion of cost associated with point ξ_{i+1}, $h(\xi_{i+1}) = q^{k_i-1}(\xi_i, \{u_{j,p}\}_{p=l_j-k_i}^{l_j-2}) - q^{k_i-2}(\xi_{i+1}, \{u_{j,p}\}_{p=l_j-k_i+1}^{l_j-1})$

$$h(\xi_{i+1}) < J(\xi_i) - J_{A_0}(\xi_{i+1}) + \frac{\varepsilon}{2K_J} \tag{4.68}$$

From the inductive assumption that (4.65) holds for $i+1$, it follows that

$$q^{l-i-1}(\{\xi_j\}_{j=i+1}^{l-1}) < J(\xi_{i+1} - J_{A_0}(\xi_{i+1}) + J(\xi_i) + \frac{(l-i-1)\varepsilon}{K_J} + \frac{\varepsilon}{2K_J} \tag{4.69}$$

From the choice of the target set A_0 it follows that the optimal costs of transfer of ξ_{i+1} to the origin, $J(\xi_{i+1})$ and to A_0, $J_{A_0}(\xi_{i+1})$, differ by less than $\frac{\varepsilon}{2K_J}$. Therefore it is seen that

$$q^{l-i-1}(\{\xi_j\}_{j=i+1}^{l-1}) < J(\xi_i) + \frac{(l-i)\varepsilon}{K_j} \tag{4.70}$$

Hence (4.65) indeed holds for all elements of the trajectory. This condition was obtained for the target set A_0, rather then $\mathcal{B}^n(0,\varepsilon_s)$, as required in the statement of the proposition. But if the trajectory enters $\mathcal{B}^n(0,\varepsilon_q)$ earlier than A_0, say for some $l^\star < l$, the associated cost will be obviously smaller, and will also satisfy

$$q^{l^\star-i-1}(\{\xi_j\}_{j=i+1}^{l^\star-1}) < J(\xi_i) + \varepsilon_q \tag{4.71}$$

This concludes the proof.

The importance of the above proposition is that it establishes existence of suboptimal feedback mapping for an arbitrarily small target set and optimization accuracy, thus ensuring for all practical purposes possibility of optimal feedback control. Since the set of suboptimal feedback mappings is not empty, it will make sense to search for neural network approximations of such controllers.

4.4 Suboptimal control by discontinuous networks

As in the case of time-optimal control, it will be shown that the suboptimal feedback, demonstrated to exist in the previous section, may be realized in a form of a multilayered network with discontinuous neuron activation functions. To this end, theorem 2.5 will be used. To utilize that result, a slightly stronger version of the proposition 4.4 will be needed, that is existence of an appropriate set valued feedback mapping must be shown. The first step to achieve this will be an improved version of lemma 4.3.

Lemma 4.5 *Suppose that system (3.5) satisfies assumptions 3.1 and 3.2. Let C be a compact set of initial conditions controllable to the origin in finitely many steps,*

$C \subset \mathcal{C}^k(\{0\})$, and let A_0 be an open neighborhood of the origin, controllable to itself in one step, $0 \in A_0 \subset \mathcal{C}^1(A_0)$. Then, there it is possible to choose an integer $K_J > 0$, such that for any $\varepsilon > 0$, and for any $x \in C$, there exist two finite collections of open sets $L_{x,i}$ and $V_{x,i}$, and a corresponding collection of open sets of control values $U_{x,i} \subset U$, $i = 0, \ldots, l_x$, such that if $\tilde{x} \in V_{x,i}$ and $\tilde{u} \in U_{x,i}$, then

$$f(\tilde{x}, u_{x,i}) \in L_{x,i+1} \tag{4.72}$$

and if $\tilde{x} \in V_{x,l_x}$ and $\tilde{u} \in U_{x,l_x}$ then

$$f(\tilde{x}, u_{x,l_x}) \in A_0 \tag{4.73}$$

Additionally, it holds that $x \in L_{x,0}$, $V_{x,l_x} \subset A_0$, $L_{x,i} \subset clos(L_{x,i}) \subset V_{x.i}$, sets $V_{x,i}$ are mutually disjoint, and the cost of transfer of any $\tilde{x} \in V_{x,i}$ to A_0 using controls $\tilde{u}_j \in U_{x,j}$, $j = i, \ldots, l_x - 1$ satisfies

$$q^{l_x-i-1}(\tilde{x}, \{\tilde{u}_j\}_{j=i}^{l_x-2}) < J(\tilde{x}) + \varepsilon \tag{4.74}$$

Proof: The proof is basically a repetition of the proof of lemma 4.3, with the modification that two sets $L_{x,i}$ and $V_{x,i}$ are constructed at each step. Steps (4.43) to (4.47) are repeated unchanged, resulting in an appropriate control horizon l_x and a control sequence $\{u_{x,i}\}_{i=0}^{l_x-1}$. Then the recursive construction of the required sets is performed. For $i = l_x$ a neighborhood $V_{x,l_x} \subset A_0$ of the point $x_{l_x} \in A_0$ and an open set U_{x,l_x} are chosen so that (4.73) is satisfied, and then another open neighborhood of x_{l_x}, $L_{x,l_x} \subset clos(L_{x,l_x}) \subset V_{x,l_x}$ can always be chosen. Similarly, with sets $V_{x,i+1}$ and $L_{x,i+1}$ already defined, it is possible to find a neighborhood $V_{x,i}$ of x_i and a neighborhood $U_{x,i}$ of $u_{x,i}$ can be chosen so that (4.72) is satisfied, and another open neighborhood of x_i, $L_{x,i} \subset clos(L_{x,i}) \subset V_{x,i}$ can also be constructed. In addition, $V_{x,i}$ can always be constructed small enough, so that if $\tilde{x} \in V_{x,i}$, then (4.48) and (4.49) are satisfied, and that sets $V_{x,i}$ are mutually disjoint, for the reasons explained in the proof of lemma 4.3. It is easily seen that if any control sequence $\{u_{x,j}\}_{j=i}^{l_x-1}$ is applied to an initial condition $\tilde{x}_0 \in V_{x,i}$ for some $0 \leq i < l_x$ then the resulting trajectory satisfies $\tilde{x}_j = f^j(\tilde{x}_0, \{u_{x,p}\}_{p=i}^{j-1}) \in L_{x,i+j} \subset V_{x,i+j}$, and in particular $\tilde{x}_{l_x-i} \subset V_{x,l_x} \subset A_0$.

The argument, that (4.74) can be satisfied as well, is identical to steps (4.50) to (4.53) in proof of lemma 4.3. This concludes the proof.

With the above lemma established, it is now possible to show existence of an appropriate set valued feedback mapping, which in turn will allow suboptimal feedback by a neural network.

Proposition 4.6 *Suppose that system (3.5) satisfies assumptions 3.1, 3.2, 3.4 and 3.5. Let C be a compact set of initial conditions controllable to the origin in finitely many steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$ and $\varepsilon_q > 0$ be arbitrary numbers. Then there exists an integer $K_J > 0$, and an open set $B \subset \Re^n$, such that $C \subset B$, and a set valued feedback mapping $\Phi : B \to \mathcal{P}(U)$, such that if control value applied at each moment k satisfies $u_k \in \Phi(x_k)$, with an arbitrary initial condition $x_0 \in C$, then the solution of the closed loop system reaches the target sets $\mathcal{B}^n(0, \varepsilon_s)$ in finitely many, $k_x < K_J$, steps incurring cost*

$$q^{k_x-1}(\{x_i\}_{i=1}^{k_x-1}) < J(x_0) + \varepsilon_q \tag{4.75}$$

and the trajectory satisfies $x_i \in \mathcal{B}^n(0, \varepsilon_s)$ for all $i \geq k_x$. The feedback mapping may be chosen such that images $\Phi(x)$ are open, and the mapping itself is lower-semicontinuous.

Proof: The proof is similar to that of proposition 4.4. First lemma 3.7 is invoked to obtain an open set $A_0 \subset \mathcal{B}^n(0, \varepsilon_s)$, controllable to itself in one step, $\text{clos}(A_0) \subset \mathcal{C}^1(A_0)$. Such constructed A_0 will be the considered target set for the suboptimal transfer. Next, the control horizon K_J corresponding to the target set A_0 is found through lemma 4.1. Then invoking lemma 4.5 with $\varepsilon = \varepsilon_q$ for every $x \in C$, construct open sets $V_{x,j} \subset \Re^n$, $L_{x,j} \subset \Re^n$ and $U_{x,j} \subset U$, $0 \leq j \leq l_x < K_j$, such that conditions (4.40-4.42) are satisfied.

Since sets $L_{x,0}$ are defined for all $x \in C$, they form an open covering of C. Therefore it is possible to choose a finite subcovering $L_{i,0}$, $i = 1, \ldots, N$, so that

$$C \subset \bigcup_{i=1}^{N} L_{i,0} \tag{4.76}$$

To each of open sets $L_{i,0}$ there corresponds its element x_i, finite control horizon l_i, the control sequence $\{u_{i,j}\}_{j=1}^{l_i-1}$, the remaining sets $L_{i,j}$, $j = 1, \ldots, l_i - 1$, the open sets $V_{i,j}$, and open sets of control values $U_{i,j} \subset U$, such that $L_{i,j} \subset \text{clos}(L_{i,j}) \subset V_{i,j}$ and satisfying the properties (4.72-4.74). The covering may be chosen such that the compact set $\text{clos}(A_0)$ is covered by sets $V_{i,0}$, whose corresponding elements x_i satisfy $x_i \in \text{clos}(A_0)$. Again, it is possible to arrange all sets $L_{i,0}$ in the order of nonincreasing horizons l_i, so that $l_i \geq l_{i+1}$, and to find integers $M_{K_J} \leq \ldots M_1 = N$ such that for all indices i such that $i \leq M_k$ it holds that $l_i \geq k$. For each $k \geq 1$ define unions of all sets $V_{i,j}$ and $L_{i,j}$ that correspond to suboptimal transfer to A_0 in k steps $1 \leq k \leq K_J$

$$A_k = \bigcup_{i=1}^{M_k} L_{i,l_i-k} \tag{4.77}$$

$$B_k = \bigcup_{i=1}^{M_k} V_{i,l_i-k} \tag{4.78}$$

It follows that

$$A_k \subset \text{clos}(A_k) \subset B_k \tag{4.79}$$

The set on which the feedback mapping will be defined is

$$B = \bigcup_{k=1}^{K_J} B_k \tag{4.80}$$

The open sets A_k, and consequently also sets B_0, cover the compact set C

$$C \subset \bigcup_{k=1}^{K_J} A_k \subset B \tag{4.81}$$

To define the desired feedback mapping, for each B_k, $1 < k \leq K_J$, a collection of open subsets $W_{k,i}$, $i = 1, \ldots, M_k$ is defined as

$$W_{k,i} = (V_{i,l_i-k} \setminus (\bigcup_{i=1}^{k-1} \text{clos}(A_i)) \tag{4.82}$$

It is seen that the open sets $W_{k,j}$ cover the compact set C

$$C \subset B = \bigcup_{k=1}^{K_J} B_k = \bigcup_{k=1}^{K_J} \bigcup_{j=1}^{M_k} W_{k,j} \tag{4.83}$$

The desired set valued feedback mapping $\Phi : B \rightarrow \mathcal{P}(U)$ is defined as

$$\Phi(x) = \bigcup_{k=1}^{K_J} \bigcup_{i=1}^{M_k} \chi_{W_{k,i}}(x) U_{i,l_i-k} \tag{4.84}$$

with the above expression understood that only those of the sets U_{i,l_i-k}, for which $x \in W_{k,i}$ get summed. For any $x \in C$ the value $\Phi(x)$ is a union of open sets, hence an open set. Continuity of mapping $\Phi(x)$ follows from the openness of sets $W_{k,j}$. It will be shown that any feedback strategy which satisfies $u_i \in \Phi(x_i)$ provides suboptimal control in the required sense. First, consider some initial condition $\tilde{x} \in A_0$ and control value $\tilde{u} \in \Phi(\tilde{x})$. Then, by construction of open sets $W_{k,j}$ it follows that the only ones that may contain $\tilde{x}$ are those corresponding to $k = 1$. Suppose therefore that $\tilde{u} \in U_{i,l_i-1}$, for some $1 \leq i \leq M_1$. This means that $\tilde{x} \in W_{1,i} \subset V_{i,l_i-1}$. Then from property (4.73) it follows that $f(\tilde{x}, \tilde{u}) \in A_0$. Consequently, if a trajectory reaches set A_0, it remains therein. To see that any trajectory originating in C does reach A_0, consider some initial condition outside A_0, $\tilde{x} \in A_{k_0} \setminus A_0$, $k_0 \geq 1$. Then all $W_{k,i}$ that contain $\tilde{x}$ satisfy $k \leq k_0$. Suppose therefore that $\tilde{u} \in U_{i,l_i-k^\star}$, $\tilde{x} \in W_{i,l_i-k^\star}$ with some $k^\star \leq k_0$. Then from property (4.72) it follows that

$$f(\tilde{x}, \tilde{u}) \in L_{i,l_i-k^\star+1} \subset A_{k^\star-1} \tag{4.85}$$

Therefore, at each step the index of set A_k containing the element of the trajectory decreases. From recursive application of this argument it follows that for any $x_0 \in A_k$ the resulting trajectory satisfies $x_i \in A_0$ for all $i \geq \tilde{k}$ with $\tilde{k} \leq k$.

The argument that the suboptimality property (4.75) can also be satisfied, is similar to that in the proof of proposition 4.4. The difference is that for a given trajectory, its element x_i may now be a member of more than one of sets $W_{k,j}$ and the corresponding control value transferring x_i to x_{i+1} can be a member of many control sets $\tilde{U}_{k,j}$. The method of the proof, however, is identical. Consider a closed loop trajectory originating outside A_0. It has been shown that $x_i \in W_{k_i,j}$, with indices k_i forming a strictly decreasing sequence until $x_l \in A_0$ for some $l \leq k$ and the corresponding control values satisfy $u_i \in \tilde{U}_{k_i,j}$. For these sequences arguments (4.65) to (4.69) can be repeated virtually verbatim, thus showing that (4.75) actually holds. This concludes the proof.

From the above proposition it is immediately possible to conclude existence of suboptimal control in the form of a discontinuous neural network.

Corollary 4.7 *Suppose that system (3.5) satisfies assumptions 3.1, 3.2, 3.4 and 3.5. Let C be a compact set of initial conditions controllable to the origin in finitely many steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$ and $\varepsilon_q > 0$ be arbitrary numbers. Then there exists an integer $K_J > 0$, and a feedback mapping $\phi : \Re^n \to \Re^m$ realized by a neural network with $L > 2$ layers and Heavyside hidden neurons, such that the solution of the closed loop system reaches the target sets $\mathcal{B}^n(0, \varepsilon_s)$ in finitely many, $k_x < K_J$, steps incurring cost*

$$q^{k_x-1}(\{x_i\}_{i=1}^{k_x-1}) < J(x_0) + \varepsilon_q \tag{4.86}$$

and the trajectory satisfies $x_i \in \mathcal{B}^n(0, \varepsilon_s)$ for all $i \geq k_x$.

Proof: Use proposition 4.6 to obtain open set B, and the set valued mapping Φ. Then construct the open set

$$\Omega = \{(x, u) : x \in B, u \in \Phi(x)\} \subset \Re^n \times \Re^m \tag{4.87}$$

Then it follows that

$$C \subset \pi_n(\Omega) \tag{4.88}$$

and from theorem 2.5 it follows that there exists a desired neural network mapping such that for each $x \in C$, $(x, \phi(x)) \in \Omega$, or $\phi(x) \in \Phi(x)$. By proposition 4.6 this is a required suboptimal feedback.

The controller obtained above guarantees that any trajectory originating in C reaches, and remains in, a neighborhood of the origin, which may be made arbitrarily small. To additionally obtain convergence of any trajectory to the origin, it is needed that the zero input system is already asymptotically stable on some small neighborhood of the origin. As before, this will eliminate the need for controllability and Lipschitz continuity assumptions 3.4 and 3.5. Then the proposition (4.6) may be strengthened to obtain the following result.

Proposition 4.8 *Suppose that system (3.5) satisfies assumptions 3.1, 3.2 and 3.6. Let C be a compact set of initial conditions controllable to the origin in a finite number of steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$ and $\varepsilon_q > 0$ be arbitrary numbers. Then there exists an integer $K_J > 0$, an open set $B \subset \Re^n$, and a compact set C_0, such that $C \subset B$, a set valued feedback mapping $\Phi : B \rightarrow \mathcal{P}(U)$, such that if the control value applied at each moment k satisfies $u_k \in \Phi(x_k)$ and $u_k = 0$ if $x_k = C_0$, with an arbitrary initial condition $x_0 \in C$, then the solution of the closed loop system reaches the target sets $\mathcal{B}^n(0, \varepsilon_s)$ in finitely many, $k_x < K_J$, steps incurring cost*

$$q^{k_x-1}(\{x_i\}_{i=1}^{k_x-1}) < J(x_0) + \varepsilon_q \tag{4.89}$$

and the trajectory remains in $\mathcal{B}^n(0, \varepsilon_s)$ converging to the origin, $\lim_{i\to\infty} x_i = 0$. The feedback mapping may be chosen such that images $\Phi(x)$ are open, $0 \in \Phi(x)$ for $x \in C_0$, and the mapping itself is lower-semicontinuous.

Proof: The proof of proposition 4.6 needs to be modified only with respect to construction of set A_0. It is necessary to use lemma 3.8 to obtain an arbitrarily small open set A_0 and a compact set $C_0 \subset A_0$, together with a positive δ, such that if the control law satisfies $\| u_k \| < \delta$ for $x_k \in A_0$, then any $f(x_k, u_k) \in C_0 \subset A_0$, and that C_0 is contained in the domain of attraction of the origin. Then with the control set $U_{0,1} = \mathcal{B}^m(0, \delta)$, corresponding to the open set $W_{0,1} = B_{0,1} = A_{0,1}$, mapping Φ satisfies $0 \in \Phi(x)$ for $x \in C_0$. It is then easily seen that if a control law satisfies $u_k = 0 \in \Phi(x_k)$ for $x_k \in C_0$, then any trajectory originating in A_0 converges to the origin. The rest of the proof follows directly from the proof of proposition 4.6.

Theorem 2.5 may now be used to show existence of a suboptimal and strictly stabilizing neural feedback.

Corollary 4.9 *Suppose that system (3.5) satisfies assumptions 3.1, 3.2, 3.6. Let C be a compact set of initial conditions controllable to the origin in finitely many steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$ and $\varepsilon_q > 0$ be arbitrary numbers. Then there exists an integer $K_J > 0$, and a feedback mapping $\phi : \Re^n \to \Re^m$ realized by a neural network with $L \geq 2$ hidden layers and Heavyside hidden neurons, such that the solution of the closed loop system reaches the target sets $\mathcal{B}^n(0, \varepsilon_s)$ in finitely many, $k_x < K_J$, steps incurring cost*

$$q^{k_x - 1}(\{x_i\}_{i=1}^{k_x - 1}) < J(x_0) + \varepsilon_q \tag{4.90}$$

and the trajectory converges to the origin.

Proof: Use proposition 4.8 to obtain open set B, compact set C_0 and the set valued mapping Φ. Then construct the open set

$$\Omega = \{(x, u) : x \in B, u \in \Phi(x)\} \subset \Re^n \times \Re^m \tag{4.91}$$

Then it follows that

$$C_0 \subset C \subset \pi_n(\Omega) \tag{4.92}$$

and $(x, 0) \in \Omega$ for each $x \in C_0$. From theorem 2.5 it follows that there exists a desired neural network mapping such that for each $x \in C$, $(x, \phi(x)) \in \Omega$, or $\phi(x) \in \Phi(x)$, and if $x \in C_0$ then $\phi(x) = 0$. By proposition 4.8 this is the required suboptimal and stabilizing feedback.

4.5 Approximation by continuous networks

As mentioned before, it is desirable from the practical point of view to implement neural controllers in the form of continuous networks with continuous activation functions. Unfortunately, it is difficult to guarantee existence of continuous suboptimal feedback on an arbitrary set of initial conditions C. Using techniques similar to those of chapter 3, such a feedback will be constructed on a compact set of initial conditions $\tilde{C}$ differing from C by a set of arbitrarily small measure. Then theorem 2.1 will be used to approximate the obtained feedback using continuous networks.

The following lemma will be needed in construction of a suitable subset of C. It is a modified version of lemma 3.16. An additional requirement in comparison with results of the the previous section is that assumption 3.3 holds, that is the state transition function f is invertible for any fixed control value.

Lemma 4.10 *Suppose that system (3.5) satisfies assumptions 3.1, 3.2 and 3.3. Let* $\{K_i\}_{i=1}^{M}$ *be a nonincreasing finite sequence of integers,* $K_i \geq K_{i+1}$*, and for each* $i = 1, \ldots, M$ *suppose there exist collections of subsets of* $\Re^n$, $\{E_{i,j}\}_{j=0}^{K_i}$ *open,* $\{F_{i,j}\}_{j=1}^{K_i}$ *compact,* $\{G_{i,j}\}_{j=1}^{K_i}$ *open, and a corresponding collection of* $\{Q_{i,j}\}_{j=1}^{K_i}$*, open subsets of* U*, such that* $E_{i,j} \subset F_{i,j} \subset G_{i,j}$*, and if* $\tilde{x} \in G_{i,j}$ *and* $\tilde{u} \in Q_{i,j}$ *then* $f(\tilde{x}, \tilde{u}) \in E_{i,j-1}$*. Let* $\hat{\varepsilon} > 0$ *be an arbitrary number. Then for each* $i = 1, \ldots, M$ *it is possible to construct collections of open sets* $\tilde{E}_{i,j} \subset E_{i,j}$ *with* $\tilde{E}_{i,0} = E_{i,0}$*, compact sets* $\tilde{F}_{i,j} \subset F_{i,j}$*, open sets* $\tilde{G}_{i,j} \subset G_{i,j}$*, with* $\tilde{F}_{i,j} \subset \tilde{G}_{i,j}$*, and open sets of control values* $\tilde{Q}_{i,j} \subset Q_{i,j}$*, such that the following properties are satisfied.*

$$\tilde{F}_{i_1,1} \cap \tilde{F}_{i_2,1} = \emptyset \quad \text{if } i_1 \neq i_2 \tag{4.93}$$

$$\tilde{F}_{i_1,j} \cap \tilde{F}_{i_2,1} = \emptyset \quad \text{for } j > 1 \tag{4.94}$$

$$\mu\Big(\bigcup_{i=1}^{M} \bigcup_{j=1}^{K_i} \tilde{F}_{i,j}\Big) > \mu\Big(\bigcup_{i=1}^{M} \bigcup_{j=1}^{K_i} F_{i,j}\Big) - \hat{\varepsilon} \tag{4.95}$$

$$\tilde{E}_{i,j} \subset \bigcup_{k=1}^{j} \bigcup_{l=1}^{M_k} \tilde{F}_{l,k} \tag{4.96}$$

and if $\tilde{x} \in \tilde{G}_{i,j}$ *and* $\tilde{u} \in \tilde{Q}_{i,j}$ *then* $f(\tilde{x}, \tilde{u}) \in \tilde{E}_{i,j-1}$*.*

Proof: As before, let M_k denote the number of sets $E_{i,0}$ such that $K_i \geq k$, so that if $i \leq M_k$ then $K_i \geq k$. The proof will consist of construction, for each

$1 \leq i \leq M$ and for each $1 \leq j \leq K_i$ of appropriate increasing sequences of open sets $\tilde{E}^p_{i,j} \subset E_{i,j}$, compact sets $\tilde{F}^p_{i,j} \subset F_{i,j}$, and open sets $\tilde{G}^p_{i,j} \subset G_{i,j}$, with $\tilde{F}^p_{i,j} \subset \tilde{G}^p_{i,j}$. The following properties will be required of those sets for each $j = 1, \ldots, K_1$

$$\lim_{p\to\infty} \mu(\bigcup_{k=1}^{j}\bigcup_{i=1}^{M_j} \tilde{F}^p_{i,k}) = \mu(\bigcup_{k=1}^{j}\bigcup_{i=1}^{M_j} F_{i,k}) \tag{4.97}$$

$$G_{i,j} \subset \operatorname{clos}(\bigcup_{p=1}^{\infty} \tilde{G}^p_{i,j}) \tag{4.98}$$

$$E_{i,j} \subset \operatorname{clos}(\bigcup_{p=1}^{\infty} \tilde{E}^p_{i,j}) \tag{4.99}$$

$$\tilde{E}^p_{i,j} \subset \bigcup_{k=1}^{j}\bigcup_{l=1}^{M_k} \tilde{F}_{l,k} \tag{4.100}$$

In addition open sets of control values $\tilde{Q}^p_{i,j} \subset Q_{i,j}$ will be defined, such that and if $\tilde{x} \in \tilde{G}^p_{i,j}$ and $\tilde{u} \in \tilde{Q}^p_{i,j}$ then $f(\tilde{x}, \tilde{u}) \in \tilde{E}^p_{i,j-1}$.

This construction will be recursive. The first step is for $j = 1$, that is sets $\tilde{E}^p_{i,1}$, $\tilde{F}^p_{i,1}$, $\tilde{G}^p_{i,1}$ will be constructed for $i = 1, \ldots, M_1$. Take any strictly decreasing sequence of positive numbers $\{\varepsilon_p\}_{p=1}^{\infty}$ such that $\lim_{p\to\infty} \varepsilon_p = 0$. For each $1 \leq i \leq M_1$ construct a decreasing sequence of open sets $\{\hat{F}^p_{i,1}\}_{p=0}^{\infty}$, defined by

$$\hat{F}^p_{i,1} = \{x \in \Re^n : \operatorname{dist}(x, F_{i,1}) < \varepsilon_p\} \tag{4.101}$$

for which it follows that

$$F_{i,1} = \bigcap_{p=1}^{\infty} \hat{F}^p_{i,1} \tag{4.102}$$

and from the regularity of Lebesgue measure it is seen that

$$\lim_{p\to\infty} \mu(\hat{F}^p_{i,1}) = \mu(F_{i,1}) \tag{4.103}$$

Then define for each p mutually disjoint compact subsets of $F_{i,1}$

$$\tilde{F}^p_{i,1} = F_{i,1} \setminus \bigcup_{k=1}^{i-1} \hat{F}^p_{k,1} \tag{4.104}$$

From (4.103) and additivity of Lebesgue measure it follows that

$$\lim_{p\to\infty} \mu(\tilde{F}^p_{i,1}) = \mu(F_{i,1} \setminus \bigcup_{k=1}^{i-1} F_{k,1}) \tag{4.105}$$

Thus (4.97) is satisfied for $j = 1$. For each p define a compact set $\tilde{F}_1^p$ and an open set $\hat{F}_1^p$

$$\tilde{F}_1^p = \bigcup_{i=1}^{M_1} \tilde{F}_{i,1}^p \tag{4.106}$$

$$\hat{F}_1^p = \bigcup_{i=1}^{M_1} \hat{F}_{i,1} \tag{4.107}$$

so that

$$\bigcup_{i=1}^{M_1} F_{i,1} = \bigcap_{p=1}^{\infty} \hat{F}_1^p \tag{4.108}$$

$$\bigcup_{i=1}^{M_1} F_{i,1} = \operatorname{clos}(\bigcup_{p=1}^{\infty} \tilde{F}_i^p) \tag{4.109}$$

Next define $\tilde{G}_{i,1}^p = G_{i,1}$, $\tilde{Q}_{i,1} = Q_{i,1}$, and $\tilde{E}_{i,1}^p = E_{i,1} \cap \operatorname{int}(\tilde{F}_1^p)$. Conditions (4.98) and (4.100) are satisfied trivially, and (4.99) follows from definition of $\tilde{E}_{i,j}^p$. This completes the first step of the recursive construction.

Assume now that the desired construction has been performed up to some j. For each $i = 1, \dots M_j$, the sequences $\{\tilde{E}_{i,j+1}^p\}_{p=1}^{\infty}$, $\{\tilde{F}_{i,j+1}^p\}_{p=1}^{\infty}$, $\{\tilde{G}_{i,i+1}^p\}_{p=1}^{\infty}$ and $\{\tilde{Q}_{i,i+1}^p\}_{p=1}^{\infty}$ will be now defined as follows. The crucial part is to find appropriate $\tilde{G}_{i,j+1}^p$ and $\tilde{Q}_{i,j+1}^p$ such that if $\tilde{x} \in \tilde{G}_{i,j+1}^p$ and $\tilde{u} \in \tilde{Q}_{i,j+1}^p$ then $f(\tilde{x}, \tilde{u}) \in \tilde{E}_{i,j}^p$ To achieve this, pick an arbitrary $u \in Q_{i,j+1}$, and consider the image of $G_{i,j+1}$ under u,

$$\bar{E}_{i,j} = f(G_{i,j+1}, u) \subset E_{i,j} \tag{4.110}$$

Due to one-to-one assumption 3.3 this set is open. Construct an open subset $H_{i,j}^p$ of $\tilde{E}_{i,j}^p$ defined as

$$H_{i,j}^p = \{x \in \tilde{E}_{i,j}^p : \ \operatorname{dist}(x, (\tilde{E}_{i,j}^p)^c) > \frac{\varepsilon_p}{2}\} \tag{4.111}$$

and consider its intersection with $\bar{E}_{i,j}^p$

$$\hat{E}_{i,j}^p = \bar{E}_{i,j}^p \cap H_{i,j}^p \subset \operatorname{clos}(H_{i,j}^p) \subset \tilde{E}_{i,j}^p \tag{4.112}$$

Using again assumption 3.3, define the inverse image of this set under control u

$$\tilde{G}_{i,j+1}^p = f^{-1}(\hat{E}_{i,j}^p, u) \subset G_{i,j+1} \tag{4.113}$$

which is open. It is now possible to find an open neighborhood $\tilde{Q}^p_{i,j+1}$ of u, such that if $\tilde{x} \in \tilde{G}^p_{i,j+1}$ and $\tilde{u} \in \tilde{Q}^p_{i,j+1}$ then $f(\tilde{x}, \tilde{u}) \in \tilde{E}^p_{i,j}$ From the assumed continuity of f^{-1} it follows that (4.98) is satisfied. Compact sets $\tilde{F}^p_{i,j+1}$ are defined as

$$\tilde{F}^p_{i,j+1} = F_{i,j+1} \cap \{x \in \tilde{G}^p_{i,j+1} : \ \mathrm{dist}(x, (\tilde{G}^p_{i,j+1})^c) \geq \varepsilon_p\} \setminus \hat{F}^p_1 \tag{4.114}$$

and open sets $\tilde{E}^p_{i,j+1}$ are defined as

$$\tilde{E}^p_{i,j+1} = E_{i,j+1} \cap \mathrm{int}(\bigcup_{k=1}^{j+1} \bigcup_{l=1}^{M_k} \tilde{F}^p_{l,k}) \tag{4.115}$$

Condition (4.100) is satisfied by definition, and (4.99) follows from (4.114) and property (4.98) already established. It follows also that

$$\bigcup_{k=1}^{j+1} \bigcup_{i=1}^{M_j} F_{i,k} \subset \mathrm{clos}(\bigcup_{p=1}^{\infty} (\bigcup_{k=1}^{j+1} \bigcup_{i=1}^{M_j} \tilde{F}^k_{i,j})) \tag{4.116}$$

which implies (4.97). This completes the recursive construction. Now it is possible to choose p^* sufficiently large, so that

$$\mu(\bigcup_{j=1}^{M} \bigcup_{i=1}^{M_j} \tilde{F}^{p^*}_{j,i}) > \mu(\bigcup_{j=1}^{M} \bigcup_{i=1}^{M_j} F_{j,i}) - \frac{\hat{\varepsilon}}{L} \tag{4.117}$$

Then let $\tilde{E}_i = \tilde{E}_{i,p^*}$, $\tilde{F}_i = \tilde{F}_{i,p^*}$, $\tilde{G}_i = \tilde{G}_{i,p^*}$ and $\tilde{Q}^p_{i,j}$. These sets satisfy the required properties and the proof is concluded.

The technique used in the above proof was very similar to that used in lemma 3.16. The difference was that construction of set $\tilde{G}$ required that they are transferred to corresponding $\tilde{E}$ not just by any controls, but by controls from a specified set $\tilde{Q}$. In the following developments those control sets will be the ones allowing suboptimal transfer to the target set. The next proposition, which establishes existence of continuous suboptimal feedback on a suitably modified set of initial conditions, is a close relative of proposition 3.17.

Proposition 4.11 *Suppose that system (3.5) satisfies assumptions 3.1 through 3.5. Let C be a compact set of initial conditions controllable to the origin in finitely many* *steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$, $\varepsilon_q > 0$ and $\varepsilon_a > 0$ be arbitrary numbers. Then there exists an integer $K_J > 0$, a compact set $\tilde{C}$, such that $\mu(\tilde{C} \cap C) > \mu(C) - \varepsilon_a$, and a feedback mapping $\phi : \Re^n \to U$ continuous on $\tilde{C}$, such for an arbitrary initial condition $x_0 \in C'$ the solution of the closed loop system reaches the target sets $\mathcal{B}^n(0, \varepsilon_s)$ in finitely many, $k_x < K_J$, steps incurring cost*

$$q^{k_x-1}(\{x_i\}_{i=1}^{k_x-1}) < J(x_0) + \varepsilon_q \tag{4.118}$$

There also exists a $\delta > 0$ such that for any control strategy satisfying $\| u_k - \phi(x_k) \| < \delta$ the above properties hold.

Proof: The proof is similar to that of proposition 4.6. The set A_0 and families of corresponding sets $V_{i,j}$, $L_{i,J}$ and $U_{i,j}$ are constructed as shown there. Also sets A_k and B_k are defined as there. Define also compact sets $C_{i,j} = \mathrm{clos}(V_{i,j})$. As shown in the proof of proposition 4.6 every feedback mapping satisfying $\phi(x) \in U_{i,l_i-j}$ for $x \in V_{i,l_i-j} \setminus \bigcup_{k=1}^{j-1} \mathrm{clos}(A_j)$ is a desired suboptimal feedback. Such a continuous mapping ϕ together with set $\tilde{C}$ will be constructed here.

The proof is based on a recursive construction of collections of open sets $\tilde{V}_{i,j}^k \subset V_{i,j}$, compact sets $\tilde{C}_{i,j}^k \subset C_{i,j}$ and open sets $\tilde{L}_{i,j}^k \subset L_{i,j}$, and of corresponding open sets of control values $\tilde{U}_{i,j}^k$ for $1 \le k \le l_i$, satisfying

$$\tilde{L}_{i,j}^k \subset \bigcup_{j=1}^{k} \bigcup_{l=1}^{M_j} \tilde{C}_{i,j}^k \tag{4.119}$$

$$\tilde{C}_{i,j}^k \subset \tilde{V}_{i,j}^k \subset V_{i,j} \tag{4.120}$$

$$\mu(\bigcup_{j=1}^{K_J} \bigcup_{i=1}^{M_j} \tilde{C}_{i,j}^k) > \mu(\bigcup_{j=1}^{K_J} \bigcup_{i=1}^{M_j} C_{i,j}) - \frac{k}{K}\varepsilon \tag{4.121}$$

$$\tilde{C}_{i,j}^k \cap \bigcup_{j=1}^{k} \bigcup_{l=1}^{M_j} \tilde{C}_{l,j}^k = \emptyset \quad \text{for } i > k \tag{4.122}$$

$$\tilde{C}_{i_1,j_1}^k \cap \tilde{C}_{i_2,j_2}^k = \emptyset \quad \text{for } i_1 \neq i_2, \text{ or } j_1 \neq j_2, \text{ and } \; j_1 \le k, \; j_2 \le k \tag{4.123}$$

and if $\tilde{x} \in \tilde{V}_{i,j}^k$ and $\tilde{u} \in \tilde{U}_{i,j}^k$ then $f(\tilde{x}, \tilde{u}) \in \tilde{E}_{i,j-1}$

To perform the first step of the construction for $k = 1$, substitute $E_{i,j} = L_{i,j}$, $F_{i,j} = C_{i,j}$, $G_{i,j} = V_{i,j}$, $Q_{i,j} = U_{i,j}$, for $j = 1 \ldots, K_J$, $i = 1, \ldots l_i$. Also set $M = M_1$, $K_i = l_i$, $\hat{\varepsilon} = \frac{\bar{\varepsilon}}{K_J}$, and apply lemma 4.10. The obtained sets $\tilde{E}_{i,j}$, $\tilde{F}_i$, $\tilde{G}_i$ are the desired sets $\tilde{L}^1_{i,j} = \tilde{E}_{i,j}$, $\tilde{C}^1_{i,j} = \tilde{F}_{i,j}$, $\tilde{V}^1_{i,j} = \tilde{G}_{i,j}$, $\tilde{U}^1_{i,j} = \tilde{Q}_{i,j}$, for all $1 \leq j \leq K_J$, $1 \leq i \leq l_i$. This completes the construction for $k = 1$.

Suppose now that the construction has already been performed for some $1 \leq k < K_J$, and construct the desired sets for $k+1$. For $1 \leq j \leq k$, let $\tilde{L}^{k+1}_{i,j} = \tilde{L}^k_{i,j}$, $\tilde{C}^{k+1}_{i,j} = \tilde{C}^k_{i,j}$, $\tilde{V}^{k+1}_{i,j} = \tilde{V}^k_{i,j}$, $\tilde{U}^{k+1}_{i,j} = \tilde{U}^k_{i,j}$. Then for $k \leq j \leq K$ substitute $E_{i,j-k} = \tilde{L}^k_{i,j}$, $F_{i,j-k} = \tilde{C}^k_{i,j}$, $G_{i,j-k} = \tilde{V}^k_{i,j}$, $Q_{i,j-k} = \tilde{U}^k_{i,j}$, for $i = 1, \ldots, \leq M_k$ so that $l_i \geq k$. Also let $M = M_k$, $K_i = l_i - k$ and $\hat{\varepsilon} = \frac{\bar{\varepsilon}}{K}$ and apply lemma 4.10. With thus obtained sets $\tilde{E}_{i,j}$, $\tilde{F}_{i,j}$, $\tilde{G}_{i,j}$, $\tilde{Q}_{i,j}$, let $\tilde{L}^{k+1}_{i,k+j} = \tilde{E}_{i,j}$, $\tilde{C}^{k+1}_{i,k+j} = \tilde{F}_{i,j}$, $\tilde{V}^{k+1}_{i,k+j} = \tilde{G}_{i,j}$, $\tilde{U}^{k+1}_{i,k+j} = \tilde{Q}_{i,j}$, for $0 < j \leq K_j - k$. By lemma 4.10 these sets satisfy all the desired properties.

Note that in each step of the recursion only sets corresponding to $j > k$ are modified, so that all the previously defined compact sets $\tilde{C}$ remain mutually disjoint, as desired. With K_J steps of the recursion completed, define $\tilde{L}_{i,j} = \tilde{L}^{K_J}_{i,j}$, $\tilde{C}_{i,j} = \tilde{C}^{K_J}_{i,j}$, $\tilde{V}_{i,j} = \tilde{V}^{K_J}_{i,j}$, $\tilde{U}_{i,j} = \tilde{U}^{K_J}_{i,j}$. From the construction it follows that if $\tilde{x} \in \tilde{V}_{i,j}$ and $\tilde{u} \in \tilde{U}_{i,j}$, then $f(\tilde{x}, \tilde{u}) \in \tilde{L}_{i,j-1}$. Define

$$\tilde{C} = \bigcup_{j=1}^{K_J} \bigcup_{i=1}^{M_j} \tilde{C}_{i,j} \tag{4.124}$$

It follows that

$$\mu(\tilde{C}) > \mu(\bigcup_{j=1}^{K_J} \bigcup_{i=1}^{M_j} C_{i,j}) + \varepsilon_a \tag{4.125}$$

and consequently

$$\mu(\tilde{C} \cap C) > \mu(C) + \varepsilon_a \tag{4.126}$$

Since $\tilde{C}$ consists of mutually disjoint compact components $\tilde{C}_{i,j}$ the desired feedback may be taken as constant on $\tilde{C}_{i,j}$. That is, for each i, j, $1 \leq j \leq K_J$ and $1 \subset\leq M_j$, pick control values $\tilde{u}_{i,j} \in \tilde{U}_{i,j}$ and define

$$\phi(x) = \sum_{j=1}^{K_J} \sum_{i=1}^{M_j} \tilde{u}_{i,j} \chi_{\tilde{C}_{i,j}}(x) \tag{4.127}$$

Find also, for each $\tilde{u}_{i,j}$ a radius $\delta_{i,j}$ such that $\mathcal{B}^m(\tilde{u}_{i,j}, \delta_{i,j}) \subset \tilde{U}_{i,j}$, so that if $\tilde{x} \in \tilde{V}_{i,j}$ and $\| \tilde{u} - \phi(\tilde{x}) \| < \delta_{i,j}$ then $\tilde{u} \in \tilde{U}_{i,j}$. Then define

$$\delta = \min\{\delta_{i,j} : \ j = 1, \dots, K_j, \ i = 1, \dots, M_j\} \tag{4.128}$$

Then using similar arguments as in the proofs of propositions 4.4 and 4.6, it is easily seen that if the control signal at any moment k satisfies $\| u_k - \phi(x_k) \| < \delta$, then any trajectory starting in $x_0 \in \tilde{C}_{i_0,j_0}$ satisfies $\xi_k \in \tilde{C}_{i_k,j_k}$, where indices j_k form a strictly decreasing sequence, and finally $x_{\tilde{k}} \in A_0$ for some $\tilde{k} \leq k_0$. The argument that this transfer incurs a suboptimal cost satisfying (4.118) is identical to that used in the proof of proposition 4.4. This concludes the proof.

The above result has the same shortcoming as proposition 3.17. Although every trajectory can be transferred to A_0 in a close-to-optimal time, it is not guaranteed that the trajectory will remain therein. The reason is that in construction of sets $\tilde{C}_{i,1}$ it is not guaranteed that they cover A_0. Hence the final point of the suboptimal transient may lie in $C \setminus \tilde{C}$. Again, it will be possible to avoid it, if the system has been already locally stabilized.

Proposition 4.12 *Suppose that system (3.5) satisfies assumptions 3.1, 3.2, 3.3 and 3.6. Let C be a compact set of initial conditions controllable to the origin in finitely many steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$, $\varepsilon_q > 0$ and $\varepsilon_a > 0$ be arbitrary numbers. Then there exists an integer $K_J > 0$, a compact set $\tilde{C} \subset C$, such that $C \tilde{\cap} C > \mu(C) - \varepsilon_a$, and a feedback mapping $\phi : \Re^n \to U$ continuous on C', such for an arbitrary initial condition $x_0 \in \tilde{C}$ the solution of the closed loop system satisfies $\mathcal{B}^n(0, \varepsilon_s)$ in finitely many, $k_x < K_J$, steps incurring cost*

$$q^{k_x - 1}(\{x_i\}_{i=1}^{k_x - 1}) < J(x_0) + \varepsilon_q \tag{4.129}$$

There also exists a $\delta > 0$ such that for any control strategy satisfying $\| u_k - \phi(x_k) \| < \delta$ the above properties hold.

Proof: Only a slight modification is required to the proof of proposition 4.11. Lemma 3.6 is used to construct A_0 and V_0 as discussed there. Then the collection of compact sets $C_{i,1}$ constructed in the beginning of the proof is augmented with $\mathrm{clos}(A_0)$, and similarly collections $L_{i,1}$ and $U_{i,1}$ are augmented with A_0 and $\mathcal{B}^m(0,\delta)$ respectively. Then, in the first recursive step, while applying lemma 4.10 to obtain proper disjoint sets $\tilde{C}_{i,j}$, put the set $\mathrm{clos}(A_0)$ as the first, rather than last, so that $\mathrm{clos}(A_0) \subset \bigcup_{i=1}^{M_1} \tilde{C}_{i,1}$. Thus it is seen that $A_0 \subset C$ and the following construction of feedback function ϕ assures that if $\tilde{x} \in A_0$ then $f(\tilde{x}, \phi(\tilde{x})) \in \mathrm{clos}(V_0) \subset A_0$. The rest of the argument is identical to the proof of proposition 4.11.

The above results states that not only the suboptimal feedback may be continuous on $\tilde{C}$, but also any suitably close uniform approximation of the suboptimal strategy will provide suboptimal control. Therefore, using theorem 2.1 the two following corollaries are immediately obtained.

Corollary 4.13 *Suppose that system (3.5) satisfies assumptions 3.1 through 3.5. Let C be a compact set of initial conditions controllable to the origin in finitely many steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$, and $\varepsilon_q > 0$ be arbitrary numbers. Then there exists an integer $K_J > 0$, a compact set $\tilde{C}$, such that $\mu(\tilde{C} \cap C) > \mu(C) - \varepsilon_a$, such that for any integer $L \geq 1$ there exists a feedback mapping in the form of feedforward network $\phi : \Re^n \to \Re^m$ with L hidden layers, continuous hidden units and linear output units, such that for an arbitrary initial condition $x_0 \in \tilde{C}$ the solution of the closed loop system reaches the target sets $\mathcal{B}^n(0, \varepsilon_s)$ in a finite number of steps, $k_x < K_J$, incurring cost*

$$q^{k_x-1}(\{x_i\}_{i=1}^{k_x-1}) < J(x_0) + \varepsilon_q \tag{4.130}$$

If the system is already locally stable around the origin, the following result is obtained.

Corollary 4.14 *Suppose that system (3.5) satisfies assumptions 3.1, 3.2, 3.3 and 3.6. Let C be a compact set of initial conditions controllable to the origin in finitely many steps, $C \subset \mathcal{C}^k(\{0\})$, and let $\varepsilon_s > 0$, and $\varepsilon_q > 0$ be arbitrary numbers. Then there exists an integer $K_J > 0$, a compact set $\tilde{C}$, such that $\mu(\tilde{C} \cap C) > \mu(C) - \varepsilon_a$, such that for any integer $L \geq 1$ there exists a feedback mapping in the form of feedforward network $\phi : \Re^n \to \Re^m$ with L hidden layers, continuous hidden units and linear output units, such that for an arbitrary initial condition $x_0 \in \tilde{C}$ the solution of the closed loop system satisfies $x_k \in \mathcal{B}^n(0, \varepsilon_s)$ for all $k > k_x$, with $k_x < K_J$, and the cost of transfer to $\mathcal{B}^n(0, \varepsilon_s)$ satisfies*

$$q^{k_x - 1}(\{x_i\}_{i=1}^{k_x - 1}) < J(x_0) + \varepsilon_q \tag{4.131}$$

Proof: Both corollaries follow immediately from theorem (2.1) applied to set $\tilde{C}$, function ϕ and constant δ obtained from propositions 4.11 and 4.12, respectively.

4.6 Comments

The success of the approximation approach presented here lies in proper definition of the relaxed optimal control problem. By specifying an open neighborhood of the origin rather than the origin itself as the target set, it was possible to limit the considered control horizon to a finite number of steps. This in turn yields expression of the deviation between the actual and optimal value of the cost as a continuous function of a finite dimensional control signal. Generally the mathematical techniques used here closely mirrored those of chapter 3, and were concerned with basic properties of continuous functions.

Most of the comments made to chapter 3 apply also to the above developments. The significance of the obtained results is twofold. First, existence of a measurable state feedback approximating the optimal control, has been demonstrated under very mild assumptions. In particular no smoothness of the system or of the cost function was required. Second, it was shown that a suboptimal feedback may indeed be approximated by a neural network, which establishes applicability of the proposed approach to optimal control with summable quality criterion.

As in chapter 3 disadvantages of the theory developed lie in the area of neural controllers with continuous activation functions. Corollaries 4.13 and 4.14 are hardly satisfactory, as they do not specify the location of the excluded set of initial conditions $C \setminus \tilde{C}$. Unfortunately under general the assumptions posed here it seems impossible to obtain any stronger results.

The cost function considered here did not involve cost of control actions. In the typical linear regulator problem, terms penalizing large control actions are usually added to ensure that the control signals remain small. Most often this is done to preserve linearity, since specification of hard constraints on control actions, for example in form of a compact set U of admissible control values, would render the problem nonlinear, and very difficult to solve analytically. In the developments presented here, this reason to include terms dependent on the cost of control actions does not exist, since the problem is already nonlinear, and control values are constrained to a compact set U. However, there may still be situations, when the quality criterion does include terms dependent on control. The most intuitively obvious case would be with additive cost of control, that is with

$$q^k(x_0, \{u_i\}_{i=0}^{k-1}) = \sum_{i=1}^{k}(h_x(x_i) + h_u(u_{i-1})) \tag{4.132}$$

It seems that with similar properties of cost function h_u, as those postulated for h_x it should be possible to extend most of the results obtained here to the more general formulation (4.132). It may be, however, more difficult for the case of combined cost of the state and control $h(x_k, u_k)$. Although not commonly used, such formulation may reflect situations when certain control actions incur different kinds of cost depending on the current region of state space. Still, there should be no major difficulties involved in extending the presented theory to that case, provided that the cost function h possesses the same properties as assumed here.

Chapter 5
Application to power system control

This chapter presents an application of the neural network methodology to practical controller design for nonlinear dynamical systems. The example is centered around the idea of synthesis of near-optimal feedback through learning from a set of precalculated optimal trajectories. Even though the previous chapters were devoted to the theory concerning this problem, the results obtained there are not applicable to this example, and the techniques used here can only be called heuristic. Thus, the incompatibility between the existence theory of chapters 3 and 4 and the actual design techniques presented here, emphasize the gap between available theory and practice, which in the field of neural networks is probably wider than in any other subdiscipline of control systems.

The application discussed here is the problem of stabilization of a power transmission system subject to a large fault. Recently the research program on Flexible AC Transmission Systems (FACTS) has been launched jointly by industry and academia to improve power transmission capabilities of such systems [34]. Newly developed FACTS devices allow the dynamic alteration of parameters of transmission lines, and consequently enlarge the available stability margins. This causes, besides the intended benefits, also some problems with the design of control algorithms, since those devices affect the system dynamics in a highly nonlinear fashion and the most popular linear control techniques are no longer effective or even applicable. Among various approaches to FACTS control, neural networks also have been proposed, and successfully demonstrated to posses significant advantages in this context. This chapter summarizes the studies concerning this problem, which were reported in [35] [17] and [36].

5.1 Suboptimal control of the swing equation

This section illustrates the concept of synthesis of nearly optimal feedback control by means of a neural network. The considered system is an idealized single-machine infinite-bus power system, controlled by means of a series capacitor placed on the transmission line. The simplest model of such a system is the so called swing equation

$$\begin{aligned} \dot{\omega} &= \frac{1}{M}(P_m - \frac{E_1 E_2}{X - X_c}\sin\delta) \\ \dot{\delta} &= \omega_B(\omega - 1) \end{aligned} \tag{5.1}$$

where δ is the rotor angle, ω the normalized rotor speed, the control signal is the reactance X_c of the series capacitor placed on the line, and M, E_1E_2, P_m and X are parameters characterizing operating conditions of the system. It is assumed that signal X_c can be changed instantaneously to any desired value within the range $X_c \in [X_c^{min}, X_c^{max}]$. This is an idealization of a real power system operation, when the controlled capacitor is a dynamical system itself and its interaction with the power transmission system is considerably more complex than postulated here. However, as it is often done for the sake of simplicity, the dynamics of the actuator for X_c is ignored here.

5.1.1 Time optimal control

The general control task is to maintain the system state at the equilibrium $\omega_{eq} = 1$, $\delta_{eq} < \pi$, corresponding to some admissible value of the compensation $X_C^{min} \leq X_{c,eq} \leq X_c^{max}$, so that it is possible to maintain the system in equilibrium, after it has been reached. Specifically, it is required that if a disturbance occurs, the state of the system should be transferred to the equilibrium in minimal time. In an actual power system such a disturbance may correspond to a short circuit fault driving the system system state away from the equilibrium. In this example an arbitrary initial condition is assumed. If the initial condition is known, the corresponding time optimal control problem may be relatively easily solved using a numerical optimization method. If

$X - X_c^{max} > 0$, it is possible to introduce a transformed control signal $u = \frac{1}{X - X_c}$ which is in one-to-one relationship of the original control signal X_c, and is constrained by $u^{min} = \frac{1}{X - X_c^{min}}$ and $u^{max} = \frac{1}{X - X_c^{max}}$. Then the model may be rewritten as

$$\begin{aligned} \dot{\omega} &= \frac{1}{M}(P_m - uE_1E_2 \sin \delta) \\ \dot{\delta} &= \omega_B(\omega - 1) \end{aligned} \tag{5.2}$$

which is a system affine in control signal u. A usual solution of the time optimal-control problem for such systems is of bang-bang type, that is the control takes only two values, u^{min} or u^{max}, with finitely many switches between them. Also, any non-bang-bang control signal on a finite interval may be approximated with a bang-bang control arbitrarily closely, in the sense of distance between resulting state trajectories [29]. A convenient algorithm for the calculation of time optimal control in bang-bang form for systems affine in control is the Switching Time Variation Method (STVM) [37]. If the optimal solution contains a singular arc, that is if the optimal control signal is not bang-bang, then the STVM provides successive approximations of the optimal control, with an increasing number of switches. The STVM was used to calculate time-optimal control for system (5.1), which indeed is of bang-bang type. For this particular example a switching line may be generated in the phase plane (ω, δ) to divide it into two parts corresponding to minimal and maximal compensation, respectively. In case of higher dimensional systems, however, it may be very difficult to deduce the shape of the switching surface from a set of trajectories. Formulation of control policy through a switching line has another drawback, which applies also to the system in question - it requires availability of the state vector as a decision variable. These shortcomings will be addressed by the synthesis of the output feedback controller approximating optimal trajectories by means of a neural network.

5.1.2 Structure of the suboptimal control system

The neural controller is designed to operate in discrete time with discretization time Δ, that is the control signal is constant, $X_c(t) = u_k$, on every interval $t \in [k\Delta, (k+1)\Delta)$. The discrete time equivalent of the system (5.1) is not given analytically, however it is known the the state $x_{k+1} = [\omega((k+1)\Delta), \delta((k+1)\Delta]^T$ may be expressed as a unique and continuous function of the state $x_k = [\omega(k\Delta), \delta(k\Delta)]^T$

$$x_{k+1} = f(x_k, u_k) \tag{5.3}$$

Since the right-hand side of the equation (5.1) is differentiable and Lipschitz continuous it follows that the state transition map $f : \Re^2 \times \Re \to \Re^2$ is also differentiable and Lipschitz continuous. With control values constrained to an interval, and the linearized system controllable, the system satisfies the assumptions of corollary 3.14, so that it is possible to synthesize a Heavyside network to realize an almost time-optimal feedback $u_k = \phi(x_k)$. However, as training of a discontinuous network represents a rather difficult task, a continuous network will be used. Also, only angle δ will be used as an input to the controller. Since the time optimal policy requires information about the whole state vector, it is necessary that the controller reconstruct, at least partially, the current state from the past input-output measurements. It is assumed that it is possible to reconstruct current angular velocity $\omega_k = \omega(k\Delta)$ using present and previous values of the angle, $\delta_k = \delta(k\Delta)$, $\delta_{k-1} = \delta((k-1)\Delta)$ and the previous control value u_{k-1}. Although not substantiated by rigorous theoretical considerations, this assumption is supported by analogy from linear systems, where the state of an n-dimensional observable system may be reconstructed from input-output data from at most $n-1$ past output and control values. Thus, the structure of the neural controller is

$$u_k = \phi(\delta_k, \delta_{k-1}, u_{k-1}) \tag{5.4}$$

Observe, that even though the feedback mapping $\phi : \Re^3 \to \Re$ is realized by a static, feedforward neural network, the controller is a dynamical system, as it is fed its own previous outputs. Because of this feedback loop, and of $\delta((k-1)\Delta))$ being also a controller input, the resulting closed-loop system is no longer two-dimensional. The extended four-dimensional state vector is

$$\xi_k = [\omega_k, \delta_k, \delta_{k-1}, u_{k-1}]^T \tag{5.5}$$

and its dynamics can be represented by a function $f_c : \Re^4 \rightarrow \Re^4$ such that

$$\xi_{k+1} = f_c(\xi_k) \tag{5.6}$$

whose coordinate functions $f_c^i : \Re^4 \rightarrow \Re$ can be expressed as

$$\begin{array}{rcccl} \xi_{k+1}^1 & = & f_c^1(\xi_k) & = & f^1(\xi_k^1, \xi_k^2, \phi(\xi_k^2, \xi_k^3, \xi_k^4)) \\ \xi_{k+1}^2 & = & f_c^2(\xi_k) & = & f^2(\xi_k^1, \xi_k^2, \phi(\xi_k^2, \xi_k^3, \xi_k^4)) \\ \xi_{k+1}^3 & = & f_c^3(\xi_k) & = & \xi_k^2 \\ \xi_{k+1}^4 & = & f_c^4(\xi_k) & = & \phi(\xi_k^2, \xi_k^3, \xi_k^4) \end{array} \tag{5.7}$$

where ξ_k^i and ξ_{k+1}^i are the elements of ξ_k and ξ_{k+1}, respectively, and $f^i : \Re^3 \rightarrow \Re^2$ are coordinate functions of the state transition function f of the discretized system (5.3). The particular form of function f_c depends on the parameters - weights - of the neural controller ϕ. Therefore the task of training the controller may be rephrased as the task of adjustment of function f_c through adaptation of neural network weights.

5.1.3 Synthesis of the suboptimal controller

The neural network controller (5.4) is trained so that solutions of the closed-loop system (5.7) follow closely the truly optimal trajectories. This is obtained by presenting the network with a set of examples, that is time-optimal trajectories corresponding to different initial conditions. The objective of the training process is that, for the same set of initial conditions, the distance between the actual and the optimal trajectories is minimized. Then, it is hoped that the problem is regular enough for the network to learn the proper rule governing generation of optimal control signal, so that it will be able to provide near optimal performance for initial conditions other than those corresponding to the training data.

Suppose that open-loop optimization has been performed for M initial conditions, resulting in time-optimal trajectories $(\hat{\omega}(t)_i, \hat{\delta}(t)_i)$, $i = 1, \ldots, M$. Those are subsequently sampled with discretization step Δ, resulting in discrete-time trajectories $\hat{x}_{k,i} = (\hat{\delta}_{k,i}, \hat{\omega}_{k,i})$, $i = 1, \ldots, M$, $k = 0, \ldots, K_i$, where $\hat{x}_{k,i}$ is the k-th sample of the i-th trajectory, and K_i is the smallest index such that $K_i\Delta$ is greater than the corresponding optimal time. Then the quality of approximation is measured by the following performance criterion

$$J = \sum_{i=1}^{M} \sum_{k=1}^{K_i} h(\xi_{k,i}, \hat{x}_{k,i}) \tag{5.8}$$

where $\xi_{k,i}$ denote elements of the i-th actual trajectory of the closed-loop system obtained with the same initial condition as the desired, optimal trajectory $\hat{x}_{k,i}$, and h is the function measuring the distance between the two. The synthesis of the proper feedback consists of minimization of criterion J with respect to the weights of the network. This is done iteratively, using in each iteration model (5.7) to calculate, through simulation, the value of criterion J and the corresponding direction of change of weights, such that J decreases. This is a standard optimization problem, as long as the gradient of J as a function of weights is available. To obtain this, it is necessary to perform sensitivity analysis of the whole dynamical system (5.7) with respect to the weights. Consider only a part of the criterion J which corresponds to the i-th trajectory,

$$J_i = \sum_{k=1}^{K_i} h(\xi_{k,i}, \hat{x}_{k,i}) \tag{5.9}$$

Denote by w the entire vector of the adjustable parameters of the controller (5.4), and thus of the whole system (5.7). To calculate the sensitivity of J_i with respect to w it is necessary to define the adjoint dynamical system

$$\eta_{k-1} = \frac{\partial f_c(\xi)}{\partial \xi}^T_{|\xi=\xi_{k-1,i}} \eta_k + \frac{\partial h(\xi)}{\partial \xi}^T_{|\xi=\xi_{k-1,i}} \tag{5.10}$$

with final condition

$$\eta_{K_i} = \frac{\partial h(\xi)}{\partial \xi}^T_{|\xi=\xi_{K_i,i}} \tag{5.11}$$

Thus calculated adjoint variables η_k represent sensitivities of the performance criterion J_i with respect to the values of the state at different points of the discrete time trajectory. Then the sensitivity with respect to the parameter is found through

$$\frac{\partial J_i}{\partial w} = \sum_{k=0}^{K_i-1} -\frac{\partial f_c(\xi)}{\partial w}^T_{|\xi=\xi_{k,i}} \eta_k \tag{5.12}$$

A certain technical difficulty arises here, since the state transition function f_c of the system (5.7) is not given explicitly in discrete time, but only as a result of solving numerically equations (5.1) from $k\Delta$ to $(k+1)\Delta$. Function f_c, however, is not needed itself to calculate (5.12), and availability of its sensitivities with respect to ξ and w is sufficient. These may be found numerically for each sampling step k, again using the appropriate adjoint equation, this time, however, formulated in the continuous time

$$\dot{\zeta}(t) = \begin{bmatrix} -\frac{\partial \tilde{f}}{\partial x}^T & 0 \\ 0 & -\frac{\partial \tilde{f}}{\partial x}^T \end{bmatrix} \zeta(t) \tag{5.13}$$

where $\tilde{f}$ corresponds to the right hand side of (5.1), and $x(t)$ is the solution of (5.1) on the interval $(k\Delta, (k+1)\Delta]$. This time-varying linear equation has to be solved backwards in time from $(k+1)\Delta$ to $k\Delta$ with final condition

$$\zeta(k\Delta) = [-1, 0, 0, -1]^T \tag{5.14}$$

Then the the value of vector $\zeta(k\Delta) = \zeta_k$ represents sensitivities of elements of the state vector x_{k+1} with respect to the state x_k if the control value on the interval is independent of the state. Since the control value does depend on the state through (5.4), it is necessary also to calculate sensitivities of the state x_{k+1} with respect to the control value u_k

$$\begin{aligned} \frac{\partial F_1}{\partial u} &= \int_{k\Delta}^{(k+1)\Delta} -\frac{\partial \tilde{f}^i}{\partial u}\zeta^1 dt \\ \frac{\partial F_2}{\partial u} &= \int_{k\Delta}^{(k+1)\Delta} -\frac{\partial \tilde{f}^i}{\partial u}\zeta^3 dt \end{aligned} \tag{5.15}$$

which may be computed together with integrating (5.13). Then, the first row of the Jacobian of the state transition function f_c calculated at moment k for the argument ξ_k is given by

$$\frac{\partial f_c^1}{\partial \xi^1} = -\zeta_k^1 \tag{5.16}$$

$$\begin{aligned}
\frac{\partial f_c^1}{\partial \xi^2} &= -\zeta_k^2 + \frac{\partial f_c^2}{\partial u}\frac{\partial \phi}{\partial \xi^2} \\
\frac{\partial f_c^1}{\partial \xi^3} &= \frac{\partial f_c^1}{\partial u}\frac{\partial \phi}{\partial \xi^3} \\
\frac{\partial f_c^1}{\partial \xi^4} &= \frac{\partial f_c^1}{\partial u}\frac{\partial \phi}{\partial \xi^4}
\end{aligned}$$

and the second row is given by

$$\frac{\partial f_c^2}{\partial \xi^1} = -\zeta_k^3 \tag{5.17}$$

$$\begin{aligned}
\frac{\partial f_c^2}{\partial \xi^2} &= -\zeta_k^4 + \frac{\partial f_c^2}{\partial u}\frac{\partial \phi}{\partial \xi^2} \\
\frac{\partial f_c^2}{\partial \xi^3} &= \frac{\partial f_c^2}{\partial u}\frac{\partial \phi}{\partial \xi^3} \\
\frac{\partial f_c^2}{\partial \xi^4} &= \frac{\partial f_c^2}{\partial u}\frac{\partial \phi}{\partial \xi^4}
\end{aligned} \tag{5.18}$$

The last two rows of the Jacobian matrix are easily computed from (5.7) as f_c^3 and f_c^4 are given analytically. The fourth coordinate function of f_c is the function realized by the neural network, so sensitivities of f_c^4 with respect to ξ^2, ξ^3 and ξ^4 are calculated through backpropagation method, discussed in chapter 2. With the Jacobian of f_c calculated at each point ξ_k, it is finally possible to solve (5.10) for the adjoint variables η. To find the gradient of J_i, sensitivities of f_c with respect to parameter w are necessary. But those are easily calculated by

$$\frac{\partial f_c^i}{\partial w} = \frac{\partial f_c^i}{\partial u}\frac{\partial \phi}{\partial w} \tag{5.19}$$

This finally allows calculation of the gradient (5.12). The above procedure is repeated for each $i = 1, \ldots, M$, to obtain gradient of J

$$\frac{\partial J}{\partial w} = \sum_{i=1}^{M} \frac{\partial J_i}{\partial w} \tag{5.20}$$

Any gradient-based optimization procedure may be now used to perform minimization of criterion J with respect to the weights. Each iteration involves solution of (5.7) forward in time, to obtain the actual trajectories for given initial conditions and current weight vector w. Then the gradient of J is found through solving backward in time the discrete-time adjoint system (5.10), which in turn involves solving, for each discretization step, a continuous time adjoint system (5.13).

If the neural network (5.4) has linear output neurons, then a constrained minimization method must be used, to assure that at each time step the control value does not exceed the constraints for the control signal. An easier approach is to use output neurons with sigmoidal activation functions scaled so that their range matches the range of admissible controls. By this, a little of performance is sacrificed, since the control value never actually reaches the limits, so that the trajectories of the controlled system (5.7) inevitably reach the equilibrium slightly later than the optimal ones. This drawback, however, is offset by the possibility of using one of the unconstrained minimization algorithms, which are considerably easier to implement and more reliable than the constrained ones.

The above derivation of the gradient of the quality criterion is a part of standard sensitivity analysis for the dynamical systems, which can be found for example in [38]. It was provided here in full form to illustrate how to treat the problem of training a neural network placed in feedback loop in dynamical setting, when the performance criterion is not explicitly stated in terms of the network output.

The method discussed above is appropriate if the model of the system is known in analytical form, otherwise it is not possible to perform the calculation of sensitivities. It is a very convenient method of synthesis of near optimal feedback using simple numerical tools. It may be observed that the method may be used for approximation of any a priori unknown feedback policy given only in form of reference trajectories representing the desired behavior of the closed loop system.

The presented method of synthesis of optimal feedback for the swing equation will be used later in this chapter as a basis for more complex design method for a multimachine system.

5.2 Control of an interconnected power system

The controller synthesis method presented in the previous section is suited only for a system whose model is known and considered accurate with respect to the parameter values used. In real applications, such situations happen rarely, if at all. In particular, dynamic behavior of a power system can hardly be modeled as a two dimensional swing equation with fixed parameters. In this section a heuristic design method is discussed, which may incorporate the above suboptimal design for a fixed model, as an alement of more complex control system for plants with uncertain models. The approach is illustrated on application to a far more complex power system model, than that discussed before.

5.2.1 Hierarchical controller architecture

The design approach presented here is an heuristic attempt to address the problem of controller synthesis for uncertain nonlinear systems. While most efficient design techniques, including the one discussed in the previous section, rely upon accurate knowledge of the plant's dynamics, in practical systems such knowledge is rarely available. For example, in a power system application unpredictable load changes may significantly alter the dynamics of the transmission network. One of the possible approaches to this problem is adaptive control, which requires continual, short-term-memory-based, plant relearning. Another possibility is the inherently conservative robust control, where the design process is focused on the worst case scenarios. The method discussed here avoids on-line learning of the plant dynamics yet provides fast controller reconfiguration based on long-term memory.

The method is based on the assumption that, from the control-objective point of view, it is possible to specify a finite number of system models $\mathcal{M}_1, \ldots, \mathcal{M}_N$, representing possible variants of the characteristic behavior of the system. The nominal models $\mathcal{M}_i$ may correspond to various possible values of the parameters, to different system configurations or operating conditions. It is required that for each nominal model $\mathcal{M}_i$, there exists a design procedure allowing the synthesis of a nominal controller $\mathcal{C}_i$, which meets desired performance specifications when acting on its corresponding model. Thus obtained set of controllers $\mathcal{C}_1, \ldots \mathcal{C}_N$ constitutes the lower level of the control system. Their control signals u^i are then combined together through the action of the higher level of the hierarchical controller. The latter consists of a control mode classifier, which produces N classifying signals r^i corresponding to each of the lower level controllers $\mathcal{C}$. The classifying signals are used as multipliers, or weights to generate a linear combination of signals u_i, which is later fed through a saturating sigmoidal element ψ to ensure that the thus formed control signal satisfies the original constraints. The overall control action then may be represented in the form

$$u = \psi(\sum_{i=1}^{N} r^i u^i) \tag{5.21}$$

which resembles the structure of an artificial neuron (2.4), whose inputs are the outputs u_i from a set of fixed controllers, and whose weights r_i are adaptively changed by a supervisory algorithm.

Controllers $\mathcal{C}_i$ represent control laws that are verified only for nominal models $\mathcal{C}_i$. The responsibility of the classifying system is to achieve the control objective even when none of the nominal models is fully applicable. An appropriate weighting mapping will be realized by means of a feedforward neural network. Through training process, involving different operating conditions, the neural classifier will learn to identify, from a set of past measurements, an appropriate linear combination of N control signals, such that the control objective is met. The training will be based, as in the previous section, on a set of reference trajectories representing the desired behavior of the system. Then the quality criterion used during the training process will be the closeness of approximation of the reference trajectories by those resulting from actions of the hierarchical controller. From such a choice of the quality criterion, it follows that the particular weighting pattern which is chosen by the classifying network does not have to correspond to the location of the actual system in the model space. More precisely, assume that all the nominal models $\mathcal{M}_i$ are identically parametrized in a finite-dimensional space, and let p_i be the parameter corresponding to the model $\mathcal{M}_i$. Assume also, that the actual system is best characterized by a convex combination of parameters

$$\hat{p} = \sum_{i=1}^{N} \alpha_i p_i \tag{5.22}$$

Then the coefficients α_i may not be the proper classifying signals r^i to use in (5.21). Moreover, the proposed hierarchical controller is expected to generate a weighting pattern without any direct correspondence to the model parametrization.

The hierarchical configuration presented here does not depend on the implementation of the controllers $\mathcal{C}_i$. They may result from different design procedures available for models $\mathcal{M}_i$ and may utilize different output measurements, so that for each operating mode a distinct measurement vector may be used. In the following example, however, all the lower-level controllers are of the same structure.

5.2.2 Controller structure for power system application

The system to be controlled is a model of Western North American Power System (WNAPS) consisting of eleven coupled second-order equations corresponding to eleven major generating areas. As in the case of the previous second-order example, the system dynamics is controlled by means of a variable series capacitor placed on the Pacific intertie. Again, the control task is to bring the system dynamics to the equilibrium following a disturbance. This task will be approached using the hierarchical control method introduced in the previous section. To use the idea of simplified nominal models describing the critical behavior of the system, the dynamics of the 22nd order model of WNAPS is approximated by a time-varying second order model. Following [39] the inter-area swing angle is defined as:

$$\delta = \frac{\sum_{i=1}^{4} M_i \delta_i}{\sum_{i=1}^{4} M_i} - \frac{\sum_{i=5}^{8} M_i \delta_i}{\sum_{i=5}^{8} M_i} \tag{5.23}$$

where δ_i and M_i are respectively swing angles and inertias of the individual generators. Then the approximate second-order model of the system is given by

$$\begin{aligned} \dot{\omega} &= \frac{1}{M}(P_m - \frac{E_1 E_2}{X - X_c}\sin\delta) \\ \dot{\delta} &= \omega_B(\omega - 1) \end{aligned} \tag{5.24}$$

This model is of the same form as that discussed in section (5.1). Higher-order dynamics of the actual system are imbedded in the time-varying parameters E_1E_2 and P_m, which are assumed to be bounded by $E_1E_2^{min} \leq E_1E_2(t) \leq E_1E_2^{max}$ and $P_m^{min} \leq P_m \leq P_m^{max}$. For the particular system model used here, the extremal values are estimated as $E_1E_2^{min} = 0.4$, $E_1E_2^{max} = 0.75$, $P_m^{min} = 0.4$, $P_m^{max} = 0.9$, and other parameters are $M = 14.31$, $\omega_B = 120\pi$, $X = 1$. The control signal X_c is constrained by $0.2 \leq X_c \leq 0.6$. For the purpose of controller synthesis, four time invariant nominal models of the form (5.1) are constructed, that correspond to the extremal values of the varying parameters

$$\begin{array}{llll} \mathcal{M}_1: & E_1E_2^{min},\ P_m^{min} & \mathcal{M}_2: & E_1E_2^{min},\ P_m^{max} \\ \mathcal{M}_3: & E_1E_2^{max},\ P_m^{min} & \mathcal{M}_4: & E_1E_2^{max},\ P_m^{max} \end{array} \tag{5.25}$$

For each of the nominal second-order models a neural controller of the form (5.4) is designed using the training procedure discussed in section 5.1.3. Then, the classifier is assumed of the form $\Psi : \Re^5 \rightarrow \Re^4$

$$r = \Psi(\delta_k, \delta_{k-1}, \delta_{k-2}, u_{k-1}, u_{k-2}) \tag{5.26}$$

The reason for this particular structure of the input signals fed to the classifier was that it should discriminate different system behaviors under the control signal; therefore the depth of its memory with respect to past control and output signals should generally be greater than that of the nominal controllers. For the system in question, increasing this depth by one leads to satisfactory results.

The final control signal is formed according to

$$u_k = \psi(\sum_{i=1}^{4} \phi_i(\delta_k, \delta_{k-1}, u_{k-1})\Psi^i(\delta_k, \delta_{k-1}, \delta_{k-2}, u_{k-1}, u_{k-2})) \tag{5.27}$$

where Ψ^i denotes the i-th coordinate function realized by the neural classifier and ϕ_i is the mapping realized by the i-th nominal controller corresponding to the fixed second-order nominal model $\mathcal{M}_i$

5.2.3 Training of the classifier

The general idea of the training process for the classifier is the same as for the lower-level suboptimal controllers. The basic difference is that the training does not utilize knowledge of the exact model of the eleven machine system. While the sensitivity computations outlined in section 5.1.3 were feasible for a second-order system, they would become prohibitively complicated for a complex 22nd order system. Moreover, the very idea of the hierarchical controller is to circumvent the problem of inaccurate modeling through introduction of a family of simplified nominal models. Therefore, for the purpose of the controller synthesis, the full model is assumed unknown. Instead only simplified models of the form (5.1) are used. To assure that the resulting controller will perform as desired, the set of four nominal models, $\mathcal{M}_1, \ldots, \mathcal{M}_4$, is augmented with nine more training models, corresponding to the following intermediate combinations of the uncertain parameters E_1E_2 and P_m

$$\begin{array}{llll}
\mathcal{M}_5: & \frac{1}{2}\Delta E + E_1E_2^{min},\ P_m^{min} & \mathcal{M}_6: & \frac{1}{2}\Delta E + E_1E_2^{min},\ P_m^{max} \\
\mathcal{M}_7: & \frac{1}{2}\Delta E + E_1E_2^{min},\ \frac{1}{2}\Delta P + P_m^{min} & \mathcal{M}_8: & E_1E_2^{min},\ \frac{1}{2}\Delta P + P_m^{min} \\
\mathcal{M}_9: & E_1E_2^{max},\ \frac{1}{2}\Delta P + P_m^{min} & \mathcal{M}_{10}: & \frac{1}{4}\Delta E + E_1E_2^{min},\ \frac{1}{4}\Delta P + P_m^{min} \\
\mathcal{M}_{11}: & \frac{1}{4}\Delta E + E_1E_2^{min},\ \frac{3}{4}\Delta P + P_m^{min} & \mathcal{M}_{12}: & \frac{3}{4}\Delta E + E_1E_2^{min},\ \frac{1}{4}\Delta P + P_m^{min} \\
\mathcal{M}_{13}: & \frac{3}{4}\Delta E + E_1E_2^{min},\ \frac{3}{4}\Delta P + P_m^{min} & &
\end{array} \tag{5.28}$$

where $\Delta E = E_1E_2^{max} - E_1E_2^{min}$ and $P_m = P_m^{max} - P_m^{min}$ Thus, the training process utilizes $N_t = 13$ second-order models $\mathcal{M}_i$ of the form (5.1). For each of those models a set of time-optimal trajectories, obtained from application of the Switching Time Variation Method, is specified. The quality criterion is specified as before

$$J = \sum_{j=1}^{N_t} \sum_{i=1}^{M} \sum_{k=1}^{K_{i,j}} h(\xi_{k,i,j}, \hat{x}_{k,i,j}) \tag{5.29}$$

where $\xi_{k,i,j}$ and $\hat{x}_{k,i,j}$ are the k-th elements of the i-th actual and optimal trajectory corresponding to j-th training model $\mathcal{M}_j$, $j = 1, \ldots, N_t$. Since the training models are exactly known, the training procedure may be performed using the same type of sensitivity calculations as in section 5.1.3. Consider the portion of the quality criterion corresponding to the i-th trajectory for the j-th model, $\xi_{k,i,j}$

$$J_{i,j} = \sum_{k=1}^{K_{i,j}} h(\xi_{k,i,j}, \hat{x}_{k,i,j}) \tag{5.30}$$

Denote the discrete-time state transition map (5.6) of the j-th nominal model as f_j. Then the closed-loop dynamics is described by the following six-dimensional model

$$\xi_{k+1} = f_c(\xi(k)) \tag{5.31}$$

where the extended state is

$$\xi_k = [\omega_k, \delta_k, \delta_{k-1}, \delta_{k-2}, u_{k-1}, u_{k-2}]^T \tag{5.32}$$

and the coordinate functions f_c^i are defined as

$$\begin{aligned}
\xi_{k+1}^1 = f_c^1 &= f_j^1(\xi_k^1, \xi_k^2, u_k) \\
\xi_{k+1}^2 = f_c^2 &= f_j^2(\xi_k^1, \xi_k^2, u_k) \\
\xi_{k+1}^3 = f_c^3 &= \xi_k^2 \\
\xi_{k+1}^4 = f_c^4 &= \xi_k^3 \\
\xi_{k+1}^5 = f_c^5 &= u_k \\
\xi_{k+1}^6 = f_c^6 &= \xi_k^5
\end{aligned} \tag{5.33}$$

The control signal u_k is generated according to

$$u_k = \psi\left(\sum_{i=1}^{4} \phi_i(\xi_k^2, \xi_k^3, \xi_k^5)\Psi^i(\xi_k^2, \xi_k^3, \xi_k^4, \xi_k^5, \xi_k^6)\right) \tag{5.34}$$

In this phase of the synthesis process the lower level controllers are fixed, and the weight adaptation is performed with regard only to the classifier (5.26). The gradient of $J_{i,j}$ with respect the classifier weights can be calculated using exactly the same sensitivity techniques as in section 5.1.3. First, the discrete-time adjoint system of equations for criterion (5.30) and system (5.33) is formed, and then the Jacobian of the state transition map f_c is calculated for each sampling step, through integration of an appropriate continuous-time adjoint equation backwards in time from $(k+1)\Delta$ to $k\Delta$. As the derivations are almost identical to those demonstrated earlier, and by themselves quite standard, they are omitted here.

With the gradient of the quality criterion available, the training process is again a straightforward application of any standard optimization method. Here a variant of conjugate gradient was used.

5.2.4 Simulation results

For the system in question, the classifier was trained as a two layered network with sigmoidal activation functions. The lower-level controllers were also two-layered networks. After training, the hierarchical controller was applied to the full 22nd-order model of eleven machine power system. The feedback signal used as input to the controller was the synthesized angle (5.23). A typical example of the controller performance is shown in figure 1. The simulated disturbance corresponds to a a short circuit lasting 4/60 second, after which one of the transmission lines is disconnected and subsequently reconnected after another 8/60 second. It is observed that the controller successfully stabilizes the transient. Rapid changes of control signal and relatively large negative overshoot are typical for time- optimal control. It is worth noting that simulations with each of four nominal controllers in place of the hierarchical controller resulted in significantly worse control performance.

Even though the synthesis of the hierarchical controller did not depend on the model of the eleven-machine system, the controller uses information about its parameters $M_1, \ldots, M_8$ while synthesizing the inter-area swing angle (5.23). As the controller was trained to perform the almost time-optimal disturbance rejection for a wide range of dynamical behaviors, it is expected that it will display robustness with respect to inaccuracies of M_i. This was tested in simulations. Figure 2 shows

Figure 1. Hierarchical controller action following a short circuit fault

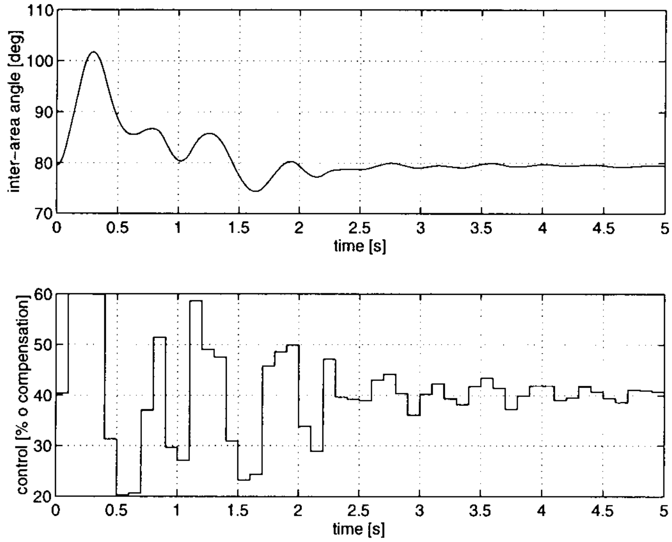
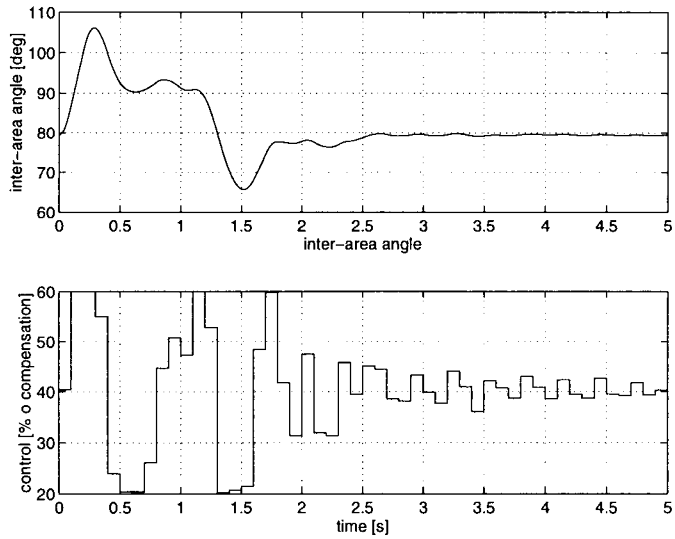


Figure 2. Hierarchical control in case of inaccurately modeled inertias



a typical simulation result, in which the parameters actually appearing in simulated eleven machine model differed from those used in (5.23) by up to 20%. The simulated disturbance is the same as in figure 1. Despite small deterioration of the controller performance, the inter-area angle is brought very quickly to its prefault equilibrium.

5.3 Comments

This chapter provided a brief illustration how to practically synthesize a near optimal controller for a nonlinear dynamical system using a neural network approach. The approach is based on learning from a set of open-loop trajectories obtained beforehand through a standard numerical optimization method. Then the distance between the optimal trajectories, and those obtained with the neural controller in the closed-loop, is the quality criterion minimized during the training process. For a plant with model known exactly, standard sensitivity analysis may be used to determine the gradient of the quality criterion with respect to the weights of the neural controller.

For systems, whose model is either uncertain, or too complicated to use in the design process, the proposed approach is based on a concept of representing the range of unknown dynamics in question by a set of simplified nominal models. For each of those, a corresponding controller is designed, which may be the suboptimal neural controller. Then a classifier, realized as a neural network, is trained to interpolate between control signals of the nominal controllers. An important feature of the method is that the entire training process of the both levels of the hierarchical controller is performed entirely on the simplified models of the system, not on the system itself. This allows utilization of sensitivity techniques to calculate necessary gradients, and of precise search algorithms for fast minimization of the performance criterion. The trained controller is not sensitive with respect to inaccuracies of such approximate modeling, as long as the set of training models represents well the possible types of the dynamical behavior of the system.

The presented controller configurations are based on heuristic premises and lack proper theoretical justification. That they are successfully employed in a difficult application, further signifies the need for more rigorous study of neural network control. With better theoretical foundations, it would be possible to assess applicability of certain neural architectures to particular control problems before time and energy consuming efforts to train a controller, which quite possibly cannot be trained to perform the particular task.

Chapter 6
Conclusions

The theory presented in chapters 3 and 4 represents a small, but very significant, contribution to the field of neural network control. Currently practical applications of neural networks too often rely on the unverified assumption, that the particular network architecture being used is suitable for the given control task. Numerous studies have been published concerning various training and adaptation schemes, to obtain desired performance of the closed-loop system with a neural controller. Very seldom, however, the fundamental question of existence of a proper controller is addressed. When a controller design method is applied to some specific control task, it is of utmost importance to establish, that in the set of possible values of design parameters there is at least one particular value, such that the corresponding controller offers the desired performance of the whole control system. Then, of course, an efficient design procedure must be devised to find one of such feasible controllers. If, however, the set of the appropriate parameters is empty, the whole design process is destined to fail. Not knowing, whether within the considered class of controllers there exists a desirable controller, transforms the synthesis process into a gamble. In particular, in case of a failure, it may be impossible to decide if the reason was a deficient method of synthesis, or if the chosen controller structure is simply not adequate for the given control task. Thus, days may be spent on attempts to improve a seemingly faulty training procedure, while in fact the chosen network architecture does not allow for the training to ever converge to a satisfactory condition. It is, therefore, very important to choose a controller structure, such that a suitable controller may be, at least in principle, found. In the context of neural networks, this question is particularly significant in the area of continuous feedforward networks, which were recently shown to be insufficient for a general stabilization task. Since continuous neural networks form an overwhelming majority of practical implementations of neural controllers, it is urgent that classes of dynamical systems and control tasks be identified, so that an appropriate controller is guaranteed to exist within this family of networks.

The question of controller existence was investigated for two classes of optimal control problems - the time optimal control, and control with a summable quality index. The motivation for this study came from the earlier proposed approach to synthesize an almost optimal feedback controller, using a neural network to learn from off-line generated open-loop optimal trajectories. The method is an attractive way to exchange a difficult problem of optimal-feedback synthesis to a large number of simpler tasks of open-loop optimization with respect to the control signal. The method has been shown successful in approximating truly optimal trajectories, yet the applications were based merely on an assumption that the optimal feedback (also simply assumed to exist) may be approximated by a neural network. Thus, as remarked above, there was a need to analyze applicability of neural network control to approximation of optimal feedback.

The problem was considered in the discrete-time setting, which corresponds to a large class of practical computer-based control systems operating on sampled signals. Analysis of optimal control in discrete, rather than continuous, time introduces much needed regularity into the problem and allows for much easier mathematical apparatus to be utilized. Since control signals and state trajectories are sequences of finite-dimensional vectors, the whole analysis may be reduced to finate-dimensional spaces, if only the control horizons are guaranteed to be finite. In both considered classes of optimal-control problems, this finiteness of the control horizon is achieved through a natural assumption of compactness of the set of initial conditions of interest, and through proper relaxation of the final condition. Instead of control to the origin, the relaxed, or suboptimal, control problem is introduced to transfer the system state to some arbitrarily small neighborhood of the origin. Also, for the case of the summable quality index, the requirement of minimization is relaxed, so that the optimal quality is achieved within some acceptably small accuracy. With the target set and the accuracy small enough, the relaxed problem is from the practical point of view equivalent to the original optimal control problem.

For such relaxed, or suboptimal control problems, the existence of state feedback controllers was demonstrated for a class of discontinuous neural networks. Thus, not only an optimal feedback policy was shown to exist, but it also may be realized in form of an appropriate neural network. Unfortunately, training of discontinuous networks poses serious difficulties, and from the practical point of view it is desirable that not only continuous, but also differentiable networks be used. The theory obtained here for this kind of networks is somewhat weaker, as a small subset has to be excluded from the set of initial conditions of interest, in order to guarantee almost optimal performance. While the excluded set may be made of arbitrarily small measure, its location in the state space cannot be a priori specified. The interpretation of these results may be that the probability of failure for any individual transient may be made as small as desired (though possibly positive), if the initial condition results from a random displacement from the origin. This interpretation may be valid for systems in which large transients occur rather seldom, such as is the case, for example, with major faults in power transmission systems.

The theory developed in this thesis constitutes a significant step towards the more complete understanding of applicability of neural networks to various control problems. It is recognized that the most important, even though the most difficult, problem to investigate is control by continuous networks. Since they are so widely, and successfully, used in practice, it is quite likely that for a very large class of dynamical systems and control tasks that particular neural architecture satisfies the existence requirement. It is therefore of utmost importance to identify at least some of those hypothetical classes of problems. Equally useful may be the investigation of problems which definitely do not admit continuous neural controllers, so that the attempts to train a controller are not undertaken in vain. So far, the nonexistence, for instance, of stabilizing continuous controllers was demonstrated in [16] by a counterexample so pathological, that one is almost forced to assume that this will not occur in other, more regular, situations. It is worth emphasizing again here, that the theory obtained for continuous networks does not imply that for some initial conditions the controller will fail. The only valid conclusion, that may be drawn from those developments, is that the success of a continuous controller cannot be guaranteed everywhere.

The other way to address the problem of adequacy, or inadequacy, of continuous neural controllers is to further research methods to train discontinuous, Heaviside networks. Though highly unlikely, a development of an efficient training algorithm would constitute a major breakthrough in neural networks, and open completely new avenues of research and applications. Even if this does not happen, the ever increasing efficiency of available computational devices may soon render training of Heavyside networks as fast as that of smooth networks.

Possible continuation of the presented research includes investigation of output, instead of state, feedback controllers. The most obvious approach would be through separation of state estimation from state feedback control operating on the estimated states. Thus, assuming that it is possible to reconstruct the state based on input-output measurements, the suboptimal controller would be first designed to use the actual state, but then fed only with the estimate. In case of a sudden displacement of the state from the equilibrium, the suboptimality of such an approach would depend on control actions undertaken between the disturbance occurrence, and the moment when the state was reconstructed accurately enough for the feedback control to act suboptimally. The related important issue, not addressed in this dissertation, is sensitivity of the suboptimal feedback with respect to inaccuracies of state measurement. For the separation approach to be feasible, the state feedback controller should allow for small inaccuracies of the estimated state. These issues certainly need investigation.

Bibliography

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

[2] D. Hebb, *The organization of behavior.* New York: John Wiley and Sons, 1949.

[3] F. Rosenblatt, *Principles of neurodynamics.* Washington: Spartan Books, 1961.

[4] M. L. Minsky and S. A. Papert, *Perceptrons.* Cambridge: MIT Press, 1969.

[5] D. Rumelhart and J. McClelland, *Parallel distributed processing: explorations of microstructure of cocognition.* Cambridge: MIT Press, 1986.

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

[7] D. B. Parker, "Learning logic," Technical report TR47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, 1985.

[8] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Science*, vol. 79, pp. 2254–2258, 1982.

[9] K. S. Narendra and K. Parthasarathy, "Adaptive identification and control of dynamical systems using neural networks," in *28th IEEE Conference on Decision and Control*, pp. 1737–1738, 1989.

[10] K. S. Narendra and S. Mukhopadhyay, "Multilevel control of dynamical systems using neural networks," in *30th IEEE Conference on Decision and Control*, pp. 170–171, 1991.

[11] 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.

[12] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks: controllability and stabilization," *IEEE Transactions on Neural Networks*, vol. 4, pp. 192–206, 1993.

[13] K. J. Hunt, D. Sbarbaro, R. Żbikowski, and P. J. Gawthrop, "Neural networks for control systems - a survey," *Automatica*, vol. 28, pp. 1083–1112, 1992.

[14] K. J. Warwick, G. W. Irwin, and K. J. Hunt, *Neural networks for control systems.* London: Peter Peregrinus, 1992.

[15] W. T. Miller, R. S. Sutton, and P. J. Werbos, *Neural networks for control.* Cambridge: MIT Press, 1990.

[16] E. D. Sontag, "Feedback stabilization using two-hidden-layer nets," *IEEE Transactions on Neural Networks*, vol. 3, pp. 981–990, 1992.

[17] R. R. Zakrzewski and R. R. Mohler, "On time-suboptimal control using artificial neural networks," in *31st IEEE Conference on Decision and Control*, pp. 895–896, 1992.

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

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

[20] G. Cybenko, "Approximation by superposition of sigmoidal functions," *Mathematics of Control, Signal and Systems*, vol. 2, pp. 303–314, 1989.

[21] M. Stinchcombe and H. White, "Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions," in *Proceedings of the international joint conference on neural networks*, 1989.

[22] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, pp. 551–560, 1990.

[23] A. K. Rigler, J. M. Irvine, and T. P. Vogl, "Rescaling of variables in backpropagation learning," *Neural Networks*, vol. 4, pp. 225–229, 1991.

[24] T. Tolleanere, "Super SAB: fast adaptive back propagation with good scaling properties," *Neural Networks*, vol. 3, pp. 561–573, 1990.

[25] A. M. Minai and R. D. Williams, "Back-propagation heuristics: a study of the extended delta-bar-delta algorithm," in *Proceedings of International Joint Conference on Neural Networks*, vol. 1, pp. 595–600, 1990.

[26] D. G. Luenberger, *Optimization by vector space methods.* New York: John Wiley and Sons, 1974.

[27] H. J. Sussmann, "Optimal control theory," in *Three decades of mathematical system theory* (H. Nijmeijer and J. M. Schumacher, eds.), pp. 409–425, Berlin: Springer Verlag, 1989.

[28] S. Łojasiewicz and W. Szpunar, "Optimal feedback," in *Modern optimal control* (E. O. Roxin, ed.), pp. 289–301, New York: Marcel Dekker, 1989.

[29] H. Hermes and J. P. LaSalle, *Functional analysis and time optimal control.* New York: Academic Press, 1969.

[30] H. J. Sussmann, "Regular synthesis for time optimal control of single input real analytic systems on the plane," *SIAM Journal of Control and Optimization*, vol. 25, pp. 1145–1162, 1987.

[31] M. Vidyasagar, *Nonlinear systems analysis.* Englewood Cliffs: Prentice-Hall, 1993.

[32] B. D. O. Anderson and J. B. Moore, *Linear optimal control.* Englewood Cliffs: Prentice-Hall, 1971.

[33] A. E. Bryson and Y.-C. Ho, *Applied optimal control.* Washington: Hemisphere, 1975.

[34] J. F. Hauer, "Operational aspects of large scale facts controllers," in *EPRI Workshop on the Future in High Voltage Transmission Systems*, 1990.

[35] R. R. Zakrzewski, W. J. Kolodziej, and R. R. Mohler, "Application of hierarchical neural-network-based control to transient stabilization of interconnected power systems," in *American Control Conference*, 1994.

[36] R. R. Zakrzewski, R. R. Mohler, and W. J. Kolodziej, "Hierarchical intelligent control with flexible ac transmission systems application," *Control Engineering Practice*, 1994.

[37] R. R. Mohler, *Bilinear control processes.* New York: Academic Press, 1973.

[38] A. Wierzbicki, *Models and sensitivities of control systems.* Amsterdam: Elsevier, 1984.

[39] D. K. Kosterev and W. J. Kolodziej, "Bang-bang series capacitor transient stability control," *IEEE Transactions on Power Systems*, 1994.