Technical University of Crete

Department of Electronic and Computer Engineering

Systems Division

Master Thesis

Bilinear Neuro-Fuzzy Indirect Adaptive Control of Unknown Nonlinear Dynamical Systems

Filipp N. Andreadis

Postgraduate Student

Chania, Crete 2012

Dedication

Many people helped me in a very special way in order for me to finish this hardworking thesis. I feel I should mention all of them. First, I would like to thank professor Mr. M.A.Christodoulou who introduced me to the special field of neuro-fuzzy controllers and gave me right from the beginning the opportunity to work in some very interesting and important aspects of Adaptive Control.

Of course, I shall mention my supervisor professor Mr. M.Zervakis who helped me a lot and showed great patience for the successful finish of this master thesis.

Furthermore, I would like to express my deep gratitude to associate professor Mr. I.Boutalis for his valuable hints and our closed cooperation for the completion of this research study.

Last but not least, I would like to thank from the bottom of my heart my parents Nick and Margaret, my brother John and my companion in life Zoe, for their great support and deep understanding. Due to that support and love I stand in this place today. Thank you for being in my life. This thesis is dedicated to them.

Contents

List of Figures

Chapter 1

Abstract

1.1 Abstract

Automatic Control of systems has played a vital role in the development of engineering. It plays a determinative role in the successful function of special systems such as spacecraft systems, automatic navigation aircraft systems, mortar shell driving systems, robotics etc. In parallel, it comprises one of the most important components of contemporary industrial and constructive processes. Theoretical development of the automatic control systems has a continuous evolution which was accelerated from the Second World War period until today. Theoretical aspects and control techniques for linear systems have met a particular progress while a special interest has been drawn during the last few decades to the control of non-linear systems, but without reaching a universal applicability. A basic prerequisite of conventional linear or non-linear control techniques, is the presence of an accurate mathematical model of the dynamical behavior of the under control system.

It is known today, that both physical and modern man-made systems can be particularly complex (multi-variable) and are characterized by many nonlinearities. This makes their dynamical mathematical description especially difficult or even impossible, a fact that often leads to their treatment as unknown (black box) systems. The complexity of these systems hinders the design of suitable control techniques. This becomes even more difficult because the dynamical mathematical model required by the conventional approaches is most of the times unknown. Even though the mathematical description is possible, there exist difficulties in the adaptation of the feedback controllers when the system is time-varying with an unknown to the designer way. These drawbacks create the demand for the development of new approximation models and control techniques that have the ability to learn and adapt to varying environmental conditions or internal dynamical behavior of the system.

Artificial neural networks and adaptive fuzzy systems constitute a reliable choice for modeling unknown systems, since during the last years they are considered as universal approximators. In this way, they can approximate any nonlinear function to any prescribed accuracy provided that sufficient hidden neurons and training data are available. Recently, the combination of artificial neural networks and adaptive fuzzy systems has lead to the creation of new approaches, fuzzy-neural or neuro-fuzzy approaches that capture the advantages of both fuzzy logic and neural networks and intend to approach systems in a more successful way. The neural and fuzzy approaches, are most of the time equivalent, differing between each other mainly in the structure of the approximator chosen.

This thesis, is based on the development of an adaptive recurrent neuro-fuzzy approximator for the identification and control of unknown multi-variable nonlinear dynamical systems, which present various nonlinearities. It extends the operational flexibility of the approximator by admiring a bilinear form in respect to the unknown parameters and proposing new updating laws for the on-line parameter updating. This approximator offers a new *Neuro-Fuzzy* (NF) dynamical description of systems that cannot be mathematically described in an accurate way. The central idea of the new approximator is an alternative description of a classical fuzzy system, which combines the definition of some *Weighted Indicator Functions* (WIF) with the fuzzy partitioning of the system output variables. In the sequence, the discontinuous WIF functions are approximated by high order neural networks (HONN's). In other words, the central idea of the approximator used regarding the fuzzy logic, is the following: Every high order neural network approximates a group of fuzzy rules associated with every center that has resulted from the fuzzy partitioning of the system output variables. Moreover, after considering the demands of the initial design assumptions in the usual neuro-fuzzy adaptive systems, it is concluded that the new neuro-fuzzy approximator used in this thesis (F-RHONN's) can perform with the existence of much less initial knowledge.

In this thesis is presented the design, analysis and simulation of new neuro-fuzzy approximators and controllers that can be used for the approximation and control of non-linear affine in the control systems in bilinear form. These systems have the form: $\dot{x} = f(x) + G(x) \cdot u$. The approximator that is being used for our bilinear approach is a new dynamical neuro-fuzzy model, which separates the real system to neuro-fuzzy subsystems. Each one of the neuro-fuzzy subsystems separately approximates the $f(x)$ and $G(x)$ terms. The method of parameter hopping is suitably adapted to the new control data and it is reassured in this way that all signals in the closed-loop remain bounded making the system Lyapunov stable. Moreover, the controllers that are being proposed are designed in such a way that the closed-loop error dynamics become linear as well as stable. Suitably adapting it, the method of parameter hopping once again reassures the existence of the control signal. This method is incorporated in the weight and partitions centers updating laws and maintains the closed-loop system Lyapunov stable.

Concluding briefly, in this master thesis the main aspect was to develop an indirect adaptive regulation of unknown nonlinear dynamical systems in bilinear form. This method is based on a Neuro-Fuzzy Dynamical Systems definition for nonlinear systems which uses the concept of *Fuzzy Dynamical Systems* (FDS) operating in conjunction with *High Order Neural Network Functions* (F-HONNFs). In this problem the plant is considered unknown, and ,so, it is approximated by a special form of a fuzzy dynamical system while in the sequel the fuzzy rules are approximated by appropriate HONNFs. Thus the identification scheme leads up to a Recurrent High Order Neural Network, which however takes into account the fuzzy output partitions of the initial FDS. This scheme does not require a-priori experts' information on the number and type of input variable membership functions making it less vulnerable to initial design assumptions. At first, the system is identified around an operation point, and then it is regulated to zero adaptively. By using the above analysis and development of the special neuro FDS, this thesis extends its applicability for the control of nonlinear systems, in a special bilinear however form. Weight and Partitions Center updating laws are provided for the HONNFs and the centers of the output membership functions respectively, which guarantee that both the identification error and the system states reach zero exponentially fast, while keeping all signals in the closed loop bounded. We assure the existence of the control signal by applying a method of parameter hopping, which is incorporated in the weight and center updating law. The applicability of the method is tested on a DC Motor system, where it is shown that by following the proposed procedure one can obtain asymptotic regulation.

Chapter 2

Introduction and Goals of the Thesis

2.1 Brief Introduction

In our world, there are two principal objectives in the scientific study of the environment: we want to understand (identify) and to control. Those two goals are in continuous interaction with each other, since deeper understanding allows firmer control, while, on the other hand, systematic application of scientific theories inevitably generates new problems while require further investigation, and so on.

It was the design of the autopilots for high-performance aircraft which primarily motivated the research in adaptive control in the early 1950's. According to Webster's dictionary, to adapt means to change (oneself) so that one's behavior will conform to new or changed circumstances. The words *adaptive systems* and *adaptive control* have been used as early as 1950 [61]. This generic definition of adaptive systems has been used to label approaches and techniques in a variety of areas despite the fact that the problems considered and approaches followed often have very little in common. In this thesis, the following specific definition of adaptive control is used:

Adaptive Control is the combination of a parameter estimator, which generates parameter estimates online, with a control law in order to control classes of plants whose parameters are completely unknown and/or could change with time in an unpredictable manner. [53]

For most engineering systems, there are two important information sources. Firstly, we have sensors which provide numerical measurements of variables (data for neural networks) and secondly, we have human experts who provide linguistic instructions and descriptions (fuzzy logic) about the system. Conventional engineering approaches can only make use of numerical information and have difficulty in incorporating linguistic information. Therefore, in our approach we tried to relax this difficulty by using an underlying fuzzy description but keeping the least information (initial estimates of fuzzy output centers) coming from experts,as our approach undertakes the final estimate of those information automatically.

Problem solving using a neuro-fuzzy *(NF)* or fuzzy-neural network approaches has become a popular research topic in the past decades [19], [62], [63], [64], [65], [66], [67], and many more. Many characteristics of the NF network contribute to this phenomenon. Some of them are, as compared to the general neural networks, faster convergence speed, and the combination of the adaptive learning capabilities from neural networks with the generality of representation from fuzzy logic [68], providing what has sometimes been referred to as *"grey box"* models. Moreover, the NF network approach automates the design of fuzzy rules and makes the combinational learning of numerical data as well as expert knowledge expressed as fuzzy if-then rules possible. In contrast to the pure neural network or fuzzy system, the new NF method possesses both of their advantages. It brings the low-level learning and computational power of neural networks into fuzzy systems, and provides the high-level human-like thinking and reasoning of fuzzy systems into neural networks [69], [70].

A recurrent neural and NF network, which naturally involves dynamic elements in the form of feedback connections used as internal memories, has been attracting great interest in the past few years [19], [71], [72], [74], [75]. Unlike the feedforward neural network whose output is a function of its current inputs only and is limited to static mapping, recurrent neural and NF network perform dynamic mapping. Like feedforward neural networks, these networks function as black boxes. We do not know the meaning of each weight and node in these networks. Recently, the concept of incorporating fuzzy logic into a recurrent neural network is proposed in some papers,[54], [55], [56], [57], [58], [59], [60], [63], [71], [74], [75], [76]. Since the NF networks have so many advantages over the feedforward neural networks, it seems worth constructing a recurrent network based on a NF approach.

In this thesis, we consider the adaptive control problem for nonlinear systems having the following form:

$$\dot{x} = f(x) + G(x) \cdot u \tag{2.1}$$

where the state $x \in R^n$ is assumed to be completely measured, the control u is in R^q, f is an unknown smooth vector field called the drift term and G is a matrix with rows containing the unknown smooth controlled vector fields g_{ij}.

In controller design based on the feedback linearization technique, the most commonly used control structure is $u = [G(x)]^{-1} \cdot [-f(x) + v]$ (for square systems, number of inputs equals with number of states), with v being a new control variable. When the nonlinearities $f(x)$ and $G(x)$ are unknown, many adaptive control schemes have been developed, [77], [78], [79], in which the unknown elements of the matrix $G(x)$ are usually approximated by a function approximator $\hat{g_{ij}}(x, W_g)$ (where W_g is an estimated weight or parameter matrix). Consequently, the estimate W_g must be such that $G(x)$ is non-singular. Several attempts have been made to deal with such a problem, as follows:

1. choosing the initial parameter $W_g(0)$ sufficiently close to the ideal value by off-line training before the controller is put into operation [17].

2. using projection algorithms to guarantee the estimate W_g inside a feasible

set, in which $\hat{g_{ij}}(x, W_g) \neq 0$ (some a-priori knowledge for the studied systems is required for constructing the projection algorithms [53], [77], [78], [80].

3. modifying the adaptive controller by introducing a sliding mode control portion to keep the control magnitude bounded [79].

4. applying neural networks or fuzzy systems to approximate the inverse of *G(x)* in [77] and [81], which requires the upper bound of the first time derivative of *G(x)* being known a-priori.

In the neuro or neuro fuzzy approaches, most of the already presented works [9]-[15] deal with idirect adaptive control (trying first to identify the dynamics of the systems and then generating a control input according to the certainty equivalence principle), whereas few authors [16] and [17] face the direct approach (i.e. directly generating the control input to guarantee stability), because it is not always clear how to construct the control law without knowledge of the system dynamics.

Recently [19], [20], [54], [55], [56], [57], [58], [59], [60] high order neural network function approximators (HONNFs) have been proposed in order to identify nonlinear dynamical systems of the form (2.1), approximated by a Fuzzy Dynamical System (FDS). The above approximation depends on the fact that fuzzy rules could be identified with the help of HONNFs.

In this master thesis HONNFs are also used for the neuro fuzzy indirect adaptive control of unknown nonlinear dynamical systems in bilinear form, which includes two interrelated phases: first the identification of the model-plant and second the adaptive control of it.

The identification phase usually consists of two main categories: structure identification and parameter identification. Structure identification involves finding the main input variables out of all possible, specifying the membership functions, the partition of the input space and determining the number of fuzzy rules which is often based on a substantial amount of heuristic observation to express proper strategy's knowledge. Most of structure identification methods are based on data clustering, such as subtractive clustering [11], mountain clustering [10] and fuzzy C-means clustering [8]. The above approaches require that all input-output data are ready before we start to identify the plant. So, those approaches are called off-line.

In our approach we use the structure identification that is also made off-line and it is based either on human expertise or on gathered data. However, the required a-priori information obtained by linguistic information or data is very limited. The only required information in this approach is an initial estimate (coming from the experts) of the centers of the output fuzzy membership functions and it is not necessary on the underlying fuzzy rules, because this is automatically estimated by the HONNFs. So, in the sequel, the centers and the neural weights are estimated on-line automatically. Based on these facts the proposed method is less vulnerable to initial design assumptions.The parameter identification part is then easily addressed by HONNFs, based on the linguistic information regarding the structural identification of the output part and from the numerical data obtained from the actual system to be modeled.

One of our consideration is that the nonlinear system is affine in the control and could be approximated with the help of two independent fuzzy subsystems. Every fuzzy subsystem is approximated by a family of HONNFs, each one being related with a group of fuzzy rules. Weight updating laws are given and we prove that when the structural identification is appropriate then the error reaches zero very fast. Moreover, an appropriate state feedback is constructed in order to achieve asymptotic regulation of the output, while keeping all signals of the system bounded in the closed loop. The existence of the control signal is always assured by suitably using a method of parameter hopping, which is incorporated in the weight and partitions centers updating laws.

The master thesis is organized as follows. Chapter 3 presents some basic facts of the presented work in the field of Adaptive Neuro-Fuzzy Identification and Control so far, while Chapter 4 sets some preliminaries related to the Recurrent Neural Networks. Chapter 5 deals with preliminaries related to the concept of Adaptive Fuzzy Systems (AFS) and the formal models used in the remaining thesis, and Chapter 6 reports on the ability of HONNFs to act as fuzzy rule approximators. The indirect neuro fuzzy adaptive regulation of affine in the control dynamical systems is developed in Chapter 7, where the method of Parameter Hopping is explained and the associated weight adaptation laws are given. Simulation results on the control of a DC Motor system are given in Chapter 8, showing that by following the proposed procedure one can obtain asymptotic regulation. Finally, Chapter 9 concludes the work of this master thesis, while the Appendix includes the proofs of the theorems we used.

2.2 Innovative Goals of the Thesis

The goals of the presented thesis cover the following issues:

The use of a NF approximator scheme for the identification of dynamical unknown nonlinear systems which presents the following distinct advantages in comparison with other approximations form the literature, in order extend the control of nonlinear systems in special bilinear form.

The union between artificial neural networks, in the sense of high order neural networks, and fuzzy logic in order to generate a more powerful and general approximator.

From the fuzzy logic aspect, the reduction of strong requirements concerning the careful selection of the fuzzy partitions of input and output variables, the selected type of the membership functions and the proper number of fuzzy rules which contribute to the success of the adaptive fuzzy system. Here is needed only an initial estimate of the fuzzy partitions centers, which in the sequel are estimated automatically by our algorithm.

From the neural network aspect, the alternative approximation of weighted indicator functions with the help of high order neural networks for our special bilinear case.

Based on the NF approximator to address a problem of indirect control (DC Motor control) as well as to theoretically solidify its behavior under different uncertainties and errors during the modeling procedure.

The alternative use of a parameter hopping method in order to assure the existence of the control signal and the robustness of the closed-loop system.

Those goals have been achieved during the hardworking design and implementation of this master thesis.

Chapter 3

Adaptive Neuro-Fuzzy Identification and Control: A General Overview

3.1 Adaptive Neuro-Fuzzy Identification and Control : Basics

Many researchers have been active in the adaptive dynamic identification and control area [82], [83], [53], [84], [85], [19], [86], to mention just a few. There are different adaptive control schemes where neural, fuzzy or NF have gained the interest of many researchers in the last years.

Neural Networks were conventionally introduced to adaptive control systems in the presence of nonlinear uncertainties that could not be globally linearly parameterized with respect to the unknown parameters. Starting from the seminar paper of [87], where rigorous stability proofs were first introduced, the field has evolved significantly over the past two decades. Topics of interest included computation of the gradients needed for backpropagation-type tuning, [130], [88], use of radial basis functions (RBFs) for feedback [81], dead-zone methods for NN parameters tuning, [17], projection-based adaptation, [89], use of e-modification schemes in adaptive laws, [90], use of dynamic NNs for feedback, [16], use of NNs for general nonlinear systems in state feedback and output feedback, [75], [91], an adaptive neuro-fuzzy control design where the local submodels are realized through nonlinear dynamical input-output mappings, [131], and use of NNs in decentralized control [132], to name only a few. NNs have also proved to be a useful tool in a wide range of applications from robotic manipulations to aircraft control [92], and beyond.

The application of fuzzy control in engineering has become also a very effective method, fuzzy logical system is a systemical inference method, which is transferred into a control strategy based on linguistics information provided by the experts. The prominent character is the robustness of the controller. So it can be applied to resolve many complex systems that cannot be controlled by regular control methods, such as nonlinear systems, time-varying systems, and delay systems. It still needs to be improved and perfected continuously to be adaptive, self-organized and self-learning. There are many discussions about the perfecting and improving of fuzzy controllers such as fuzzy controllers with good self-adaptation [4], [12], [78], [94], [95], [96], [127], [133] and so on. In recent years, many researchers, [71], [97], [134], associated neural network and genetic algorithms with fuzzy control and in this way made fuzzy control reaches a higher level.

The class of adaptive control schemes studied in this thesis is characterized by the combination of an online parameter estimator for the dynamic identification, which provides estimates of the unknown parameters at each instant of time, with a control law that is motivated from the known parameter case. The way by which the parameter estimator, also referred to as adaptive law in this thesis, is combined with the control law gives rise to an interesting approach. The indirect adaptive control scheme.

The indirect adaptive control scheme is separated into two steps: *(a)* the dynamics of the system are identified and *(b)* a control input is generated according to the certainty equivalence principle. In the sequel, we will make a short discussion about the bibliographical reference and give a brief description about the identification and control scheme that is presented in this master thesis.

3.2 Identification

System modeling, [74], [75], [98], [99], [105], has played an important role in many engineering fields such as control, pattern recognition, communications and so on. The main idea in conventional approaches is to find a global function of systems based on mathematical tools. However, it is well known that these methods have been found to be unsatisfactory in coping with ill-defined and uncertain systems. In order to circumvent these problems, model-free approaches using either fuzzy logic or neural networks have been proposed. Functionally, a fuzzy system or a neural network can be described as a function approximator. The capability of sufficiently complex multilayered feedforward networks to approximate an unknown mapping $f: R^r \to R^s$ arbitrarily well firstly has been investigated by [1], [22], [23] (and for sigmoid hidden layer activation functions).

The problem of identifying complex nonlinear systems where we have less information for the model or when we consider the controlled plant as a blackbox, something common in many physical systems, may be resolved with the help of neural network or fuzzy inference systems or a combination of them, leading to NF approaches. This is due to the universal approximation abilities of both neural networks and fuzzy inference systems, [1], [78], [100]. For the engineers, the stability issues are very important to be ensured before they move further and apply their NF networks to real systems. Therefore, first of all the derived identification algorithms have to be proven to be Lyapunov stable.

Under these specifications many researchers such as, [53], [78], [101], [135], gave the very basics and most useful information to system identification which is the first step in the subsequent control of unknown nonlinear dynamical systems with various nonlinearities. From the neural network point of view, [19], [22], [86], [87], [98], [99], [102], [103], [104], [136], [137], [138], develop on-line nonlinear system identification schemes using recurrent neural networks. Also, from the fuzzy logic point of view, [4], [8], [97], [105], [106], [107], describes an encoding fuzzy scheme for learning the plant model from data with the help of genetic algorithms or clustering methods. Finally, many researchers such as, [9], [59], [62], [71], [74], [67], [108], [109], deal with NF networks for realizing on-line structure and parameter identification.

In existing NF networks, in early 1990's almost all these systems are trained by the backpropagation (BP) algorithm [2], [4], [5], [9], [110]. The major drawback of the BP algorithms is that it arises from a non-convex optimization problem and therefore it presents slow speed of learning and entrapment in local minima. Therefore, the optimal solution is not guaranteed and BP powered neural approximators can not be implemented in real-time processes. Nowadays, although some special NF networks (fuzzy neurons and fuzzy weights) have been presented [21], [63], [66], [69], [72], [75], the typical approach of NF networks is to build standard neural networks which are designed to approximate a fuzzy system through the structure of neural networks, a methodology that was chosen during the development of this thesis.

The research by [19], [73], [103] introduced RHONNs, which have been proposed of the identification of nonlinear dynamical systems of the general form:

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

and RHONNs ensures exponential error convergence to zero. Motivated by this work the approximator was extended in [55] to include fuzzy logic, which is the sequel generated a new NF approximator. In this approximatior, every HONN approximates a group of fuzzy rules associated with a center that has resulted from the fuzzy partitioning of the system output variables or otherwise approximates a WIF (as already has been mentioned), unlike the classical determination of *Mamdani* or *Takagi-Sugeno* type approximators [106], [111].

Figure *3.1* shows the overall scheme of the proposed NF algorithm which approximates the plant model (3.1) based on state variables measurement and input signals. The arrow that passes through the NF model is indicative of the fact that the output error is used to train the NF network. The NF network's input is the plant's output (through the sigmoidals terms) and the desired NF network output is the plant's output. The error difference between the actual input of the plant and the output of the NF network is to be minimized and can be used to train the NF network. Once an identification NF model of the plant is available, this model can be used for the design of the controller ,as shown in the next subsection.

The identification procedure can be generally divided into two phases. Firstly the training phase, where the whole procedure is repeated consequently for several loops using training inputs usually random values in a certain interval, and secondly the testing phase, where a certain input signal (depending on the bounds of the training input signal) is applied in conjunction with a stable

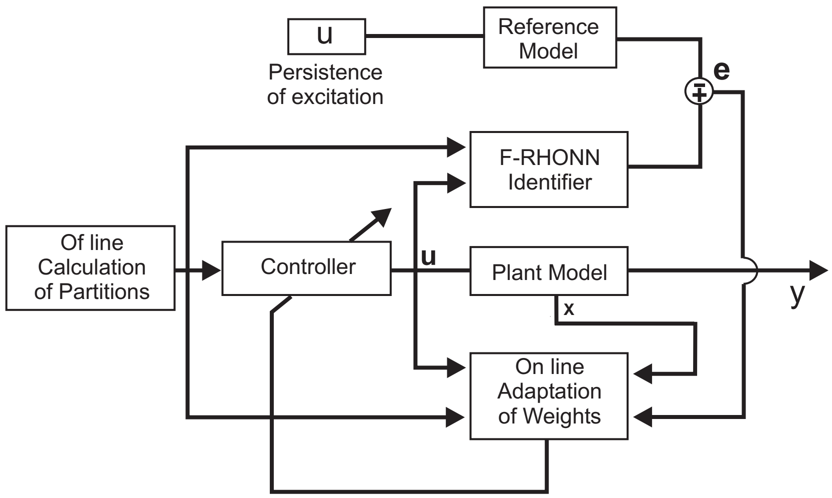

Figure 3.1: Overall scheme of the indirect adaptive neuro-fuzzy control system [54]

weight matrix (optimal weight values) extracted from the previous phase in order to test the accuracy of the approximator in reproducing the behavior of the plant in probably unknown data.

In this thesis the use of this approximator is extended for the control of nonlinear systems in bilinear form.

3.3 Indirect Control

In conventional control approaches, most of the control schemes usually devised assuming exact knowledge of (3.1) with pure or no nonlinearities. However, this is not a plausible assumption, especially when the underlying physical process is highly nonlinear and complex as has already been mentioned. To overcome this problem, appropriate identification schemes have to be applied, which will provide us with an approximate model of the plant which is absolutely necessary for the indirect control schemes. Thus, in the indirect adaptive control schemes, the dynamics of the system are firstly identified and then a control input is generated according to the certainty equivalence principle.

We can present some representative works that deal with Indirect Adaptive control.

1. Rovithakis and Christodoulou (1996) in [73] introduce techniques for controllers design keeping control objectives of a system, such as fast convergence, robustness of the proposed model and stability. This methodology is also extended for the control of system with various nonlinearities.

2. Sanchez and Bernal (2000) in [139] used the properties of Recurrent Neural Networks for the development of a control law in which is incorporated a sliding mode technique.

3. Diao and Passino (2002) in [65] investigate the indirect adaptive control of unknown nonlinear time-varying systems. The fuzzy systems used for the approximation of the dynamics of the system are in Takagi-Sugeno type and the subsystems are i/o linearizable.

4. Golea in [18] (2003) proposed an adaptive control scheme for MIMO continuous-time nonlinear systems. The methodology is based on a new fuzzy model observer with Takagi-Sugeno fuzzy models in order to estimate the unknown dynamics of the examined system.

5. Nounou and Passino (2004) in [119] introduced a scheme for canceling the nonlinearities of the system by using a feedback controller. This methodology is used for controlling nonlinear discrete-time systems with an update law based on gradient technique.

6. Kung and Chen (2005) in [118] designed an adaptive controller for nonlinear dynamic systems with states that are not all available for measurement. This controller is an observer-based fuzzy sliding mode and is being used for the indirect control of the above class of unknown systems.

7. Abid (2007) in [117] proposed a scheme for indirect adaptive control of SISO nonlinear systems with parametric uncertainties. The functions of the plant are estimated by specific fuzzy systems and the constructed adaptive control law is based on sliding mode with Lyapunov approaches.

8. Baruch (2008) in [63] introduced an alternative adaptive control scheme for complex nonlinear systems with a design of a local indirect adaptive trajectory tracking control system. As for the identification part the fuzzy rules used follow a recurrent neural procedure and the designed control laws are based on a fuzzy rule-based control system.

9. Theodoridis, Boutalis and Christodoulou (2008) in [54] introduced a new neuro-fuzzy description model for the indirect adaptive control of unknown nonlinear systems. The estimation of the system is done by a F-RHONN model which needs only an estimation of the centers of the fuzzy output variables.

Chapter 4

The RHONNs

4.1 Identification of Dynamical Systems using RHONNs

The use of multi-layer neural networks for pattern recognition and for modeling of "static" systems is currently well-known. Given pairs of input-output data the network is trained to learn the particular input-output map. Theoretical work by several researchers, including [22], and [23], have proven that, even with one hidden layer, neural networks can approximate any continuous function uniformly over a compact domain, provided the network has a sufficient number of neural networks for modeling and identification of dynamical systems. These networks, which naturally involve dynamic elements in the form of feedback connections, are known as recurrent neural networks.

Several training methods for recurrent networks have been proposed in the literature. Most of these methods rely on the gradient methodology and involve the computation of partial derivatives, on sensitive functions. In this respect, they are extensions of the backpropagation algorithm for feedforward neural networks [24]. Examples of such learning algorithms include the recurrent backpropagation [25], the backpropagation-through-time algorithms [28], the real-time recurrent learning algorithm [29], and the dynamic backpropagation [27] algorithms. The last approach is based on the computation of sensitivity models for generalized neural networks. These generalized neural networks, which were originally proposed in [26], combine feedforward neural networks and dynamical components in the form of stable rational transfer functions.

Although the training methods mentioned above have been used successfully in many empirical studies, they share some fundamental drawbacks. One drawback is the fact that, in general, they rely on some type of approximation for computing the partial derivative. Furthermore, these training methods require a great deal of computational time. A third disadvantage is the inability to obtain analytical results concerning the convergence and stability of these schemes.

Recently, there has been a concentrated effort towards the design and analysis of learning algorithms that are based on the Lyapunov stability theory [30], [31], [33], [32], [34], [35], [36], [37], [38] targeted at providing stability, convergence and robustness proofs, in this way, bridging the existed gap between theory and applications.

In this section is discussed the identification problem which consists of choosing an appropriate identification model and adjusting its parameters according to some adaptive law, such that the response of the model to an input signal (or a class of input signals), approximates the response of the real system to the same input. Since a mathematical characterization of a system is often a prerequisite to analysis and controller design, system identification is important not only for understanding and predicting the behavior of the system, but also for obtaining an effective control law. For identification models recurrent high-order neural networks are used. High-order networks are expansions of the first-order Hopfield [39] and Cohen-Grossberg [40] models that allow higher-order interactions between neurons. The superior storage capacity of them has been demonstrated in [41], [42], while the stability properties of these models for fixed-weight values have been studied in [43],[44]. Furthermore, several authors have demonstrated the feasibility of using these architectures in applications such as grammatical inference [45] and target detection [46].

The idea of recurrent neural networks with dynamical components distributed throughout the network in the form dynamical neurons and their application for identification of dynamical systems was proposed in [38]. In this section we combine distributed recurrent networks with high-order connections between neurons. At first it is shown that recurrent high-order neural networks are capable of modeling a large class of dynamical systems. In particular, it is shown that if enough higher-order connections are allowed in the network then there exist weight values such that the input-output behavior of the RHONN model approximates that of an arbitrary dynamical system whose state trajectory remains in a compact set. In the sequel, weight adjustment laws are developed for system identification under the assumption that the system to be identified can be modeled exactly by the RHONN model. It is shown that these adjustment laws guarantee boundedness of all the signals and weights and furthermore, the output error converges to zero. Then, this analysis is extended to the case where there is a nonzero mismatch between the system and the RHONN model with optimal weight values. This methodology is applied to the identification of a simple robotic manipulator system and some final conclusions are drawn.

4.2 The RHONN Model

[73]

Recurrent neural networks (RNN) models are characterized by a two way connectivity between units (i.e. ,neurons). This distinguishes them from feedforward neural networks, where the output of the unit is connected only to inputs of the next layer. In the most simple case, the state history of each neuron is governed by a differential equation of the form:

$$\dot{x}_i = -a_i x_i + b_i \sum_j w_{ij} y_j \tag{4.1}$$

Where x_i is the state of the $i-th$ neuron,a_i,b_i are constants, w_{ij} is the synaptic weight connecting the $j-th$ input to the $i-th$ neuron and y_j is the $j-th$ input to the above neuron. Each y_j is either an external input or the state of a neuron passed through a sigmoid function (i.e., $y_j = s(x_j)$), where $s(\cdot)$ denotes the sigmoid nonlinearity [73].

The dynamic behavior and the stability properties of neural network models of the form (4.1) have been studied extensively by various researchers [39],[40],[43],[44]. These studies exhibited encouraging results in application areas such as associative memories, but they also revealed the limitations inherent in such a simple model.

In a recurrent second order neural network, the input to the neuron is not only a linear combination of the components y_i, but also of their product $y_i y_k$. One can pirsue this line further to include higher order interactions represented by triplets $y_i y_k y_l$, quadraplets, etc. forming the recurrent high order neural networks (RHONNs).

Let us now consider a RHONN consisting of n neurons and m inputs. The state of each neuron is governed by a differential equation of the form:

$$\dot{x}_i = -a_i x_i + b_i \left[\sum_{k=1}^{M} w_{ik} \prod_{j \in I_k} y_j^{d_j(k)} \right] \tag{4.2}$$

Where $\{I_1, I_2, ..., I_L\}$ is a collection of L not-ordered subsets of $\{1, 2, ..., m+n\}$, a_i,b_i are real coefficients, w_{ik} are the adjustable synaptic weights of the neural network and $d_j(k)$ are non-negative integers. The state of the $i-th$ input neuron is again represented by x_i and $y := [y_1, y_2 \cdots y_{m+n}]^T$ is the input vector to each neuron defined by:

$$y = \left\{ \begin{array}{c} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \\ y_{n+1} \\ . \\ . \\ . \\ y_{n+m} \end{array} \right\} = \left\{ \begin{array}{c} s(x_1) \\ s(x_2) \\ . \\ . \\ . \\ s(x_n) \\ u_1 \\ u_2 \\ . \\ . \\ . \\ u_m \end{array} \right\} \tag{4.3}$$

where $u := [u_1, u_2 \cdots u_m]^T$ is the external input vector to the network. The function $s(\cdot)$ is monotone-increasing, differentiable and is usually represented by sigmoids of the form:

$$s(x) = a\frac{1}{1+e^{-\beta x}} - \gamma \tag{4.4}$$

where the parameters a,β represent the bound and slope of sigmoid's curvature and γ is a bias constant. In the special case where $a = \beta = 1$,$\gamma = 0$, we obtain the logistic function and by setting $a = \beta = 2$,$\gamma = 1$, we obtain the hyperbolic tangent function. These are the sigmoid activation functions most commonly used in neural network applications.

It is now introduced the $L-$dimensional vector z, which is defined as

$$z = \left\{ \begin{matrix} z_1 \\ z_2 \\ \cdot \\ \cdot \\ \cdot \\ z_L \end{matrix} \right\} = \left\{ \begin{matrix} \prod_{j \in I_1} y_j^{d_j(1)} \\ \prod_{j \in I_2} y_j^{d_j(2)} \\ \cdot \\ \cdot \\ \cdot \\ \prod_{j \in I_L} y_j^{d_j(L)} \end{matrix} \right\} \tag{4.5}$$

Hence, the RHONN model (4.2) becomes

$$\dot{x}_i = -a_i x_i + b_i \left[\sum_{k=1}^{L} w_{ik} z_k \right]. \tag{4.6}$$

Moreover, if the adjustable parameter vector is defined as

$w_i = b_i[w_{i1}, w_{i2} \cdots w_{iL}]^T$,

then (4.6) becomes

$$\dot{x}_i = -a_i x_i + w_i^T z. \tag{4.7}$$

The vectors $[w_i : i = 1, 2, \cdots, n]$ represent the adjustable weights of the network, while the coefficients $[a_i : i = 1, 2, \cdots, n]$ are part of the underlying network architecture and are fixed during training.

In order to guarantee that each neuron x_i is bounded-input bounded-output (BIBO) stable, it is assumed that $[a_i > 0, i = 1, 2, \cdots, n]$. In the special case of a continuous-time Hopfield model [39], we have $a_i = \frac{1}{R_i C_i}$, where $R_i > 0$ and $C_i > 0$ are the resistance and capacitance connected at the $i - th$ node of the network respectively [73].

The dynamic behavior of the overall network is described by expressing (4.7) in vector notation as:

$$\dot{x}_i = Ax + W^T z, \tag{4.8}$$

where $x = [x_1, x_2, \cdots, x_n]^T \in R^n, W = [w_1, w_2, \cdots, w_n]^T \in R^{Lxn}$ and $A = diag\,[-a_1, -a_2, \cdots, -a_n]$ is a nxn diagonal matrix. Since $[a_i > 0, i = 1, 2, \cdots, n]$, A is a stability matrix. Although it is not written explicitly, the vector z is a function of both the neural network state x and the external input u.

4.2.1 Approximation Properties

Consider now the problem of approximating a general nonlinear dynamical system whose input-output behavior is given by

$$\dot{\chi} = F(\chi, u), \tag{4.9}$$

where $\chi \in R^n$ is the system state, $u \in R^n$ is the system input and $F : R^{n+m} \to R^n$ is a smooth vector field defined on a compact set $\mathcal{Y} \subset R^{n+m}$.

The approximation problem consists of determining whether by allowing enough higher-order connections, there exists weights W, such that the RHONN model approximates the input-output behavior of an arbitrary dynamical system of the form (4.9).

In order to have a well-posed problem, it is assumed that F is continuous and satisfies a local Lipschitz condition such that (4.9) has a unique solution, in the sense of Caratheodory [49], and $\{\chi(t), u(t)\} \in \mathcal{Y}$ for all t in some time interval $J_T = \{t : 0 \leq t \leq T\}$. The interval J_T represents the time period over which the approximation is to be performed. Based on the above assumptions the following result is obtained:

Theorem 1 *Suppose that the system (4.9) and the model (4.8) are initially at the same state $x(0) = \chi(0)$, then for any $\epsilon > 0$ and any finite $T > 0$, there exists an integer L and a matrix $W^* \in R^{Lxn}$ such that the state $x(t)$ of the RHONN model (4.8) with L high-order connections and weight values $W = W^*$ satisfies:*

$$\sup_{0 \leq t \leq T} |x(t) - \chi(t)| \leq \epsilon.$$

The proof of the above theorem can be studied in the Appendix [73].

The above theorem proves that if sufficiently large number of connections is allowed in the RHONN model then it is possible to approximate any dynamical system to any degree of accuracy. This is strictly an existence result; it does not provide any constructive method for obtaining the optimal weights W^*. In what follows, the learning problem of adjusting the weights adaptively is considered, such that the RHONN model identifies general dynamic systems.

4.3 Learning Algorithms

[73]

In this section weight adjustment laws are developed under the assumption that the unknown system is modeled exactly by a RHONN architecture of the form (4.8). This analysis is extended in the next section to cover the case where there exists a nonzero mismatch between the system and the RHONN model with optimal weights values. This mismatch is referred to as modeling error.

Although the assumption of no modeling error is not very realistic, the identification procedure of this section is useful for two reasons:

- The analysis is more straightforward and thus easier to understand.
- The techniques developed for the case of no modeling error are also very important in the design of weight adaptive laws in the presence of modeling errors.

Based on the assumption of no modeling error, there exist unknown weight vectors $w_i^*, i = 1, 2, \cdots, n$, such that each state χ_i of the unknown dynamic system (4.9) satisfies:

$$\dot{\chi}_i = -a_i \chi_i + w_i^* z(\chi, u), \;\; \chi_i(0) = \chi_i^0. \tag{4.10}$$

where χ_i^0 is the initial $i-th$ state of the system. In the following, unless there is no confusion, the arguments of the vector field z will be omitted.

As in standard in system identification procedures, the input $u(t)$ and the state $\chi(t)$ remain bounded for all $t \geq 0$. Based on the definition of $z(\chi, u)$, as given in (4.5), this implies that $z(\chi, u)$ is also bounded. In the sections that follow different approaches for estimating the unknown parameters w_i^* of the RHONN model are developed.

4.3.1 Filter Regressor RHONN

The following lemma [73] is useful in the development of the adaptive identification scheme presented in this section.

Lemma 1 *The system described by*

$$\dot{\chi}_i = -a_i \chi_i + w_i^* z(\chi, u), \;\; \chi_i(0) = \chi_i^0 \tag{4.11}$$

can be expressed as

$$\dot{\zeta}_i = -a_i \zeta_i + z_i, \;\; \zeta_i(0) = 0, \tag{4.12}$$

$$\chi_i = w_i^{*T} \zeta_i + e^{-a_i t} \chi_i^0 \tag{4.13}$$

The proof of the above lemma can be studied in the Appendix.

Using *Lemma* 1, the dynamical system described by (4.9) is rewritten as

$$\chi_i = w_i^{*T} \zeta_i + \epsilon_i, \;\; i = 1, 2, \cdots, n, \tag{4.14}$$

where ζ_i is a filtered version of the vector z(as described by (4.5)) and $\epsilon_i := e^{a_i t} \chi_i^0$ is an exponentially decaying term which appears if the system is in a nonzero initial state. By replacing the unknown weight vector w_i^* in (4.14), by its estimate w_i and ignoring the exponentially decaying term ϵ_i, the RHONN model is obtained:

$$x_i = w_i^T \zeta_i, \;\; i = 1, 2, \cdots, n. \tag{4.15}$$

The exponentially decaying term $\epsilon_i(t)$ can be omitted in (4.15) since, as we shall see later, it does not affect the convergence properties of the scheme. The state error $e_i = x_i - \chi_i$ between the system and the model satisfies:

$$e_i = \phi_i^T \zeta_i - \epsilon_i, \tag{4.16}$$

where $\phi_i = w_i - w_i^*$ is the *weight estimation error*. The problem now is to derive suitable adaptive laws for adjusting the weights w_i, for $i = 1, \cdots, n$. This can be achieved by using well-known optimization techniques for minimization of the *quadratic cost functional*

$$J(w_1, \cdots, w_n) = \frac{1}{2}\sum_{i=1}^{n} e_i^2 = \frac{1}{2}\sum_{i=1}^{n}\left[(w_i - w_i^*)^T \zeta_i - \epsilon_i\right]^2. \tag{4.17}$$

Depending on the optimization method that is employed, different wight adjustment laws can be derived. Here, we consider the gradient and the least-squares method [50]. The gradient method yields

$$\dot{w}_i = -\Gamma_i \zeta_i e_i, \;\; i = 1, 2, \cdots, n, \tag{4.18}$$

where Γ_i is a positive definite matrix referred to as the adaptive gain or learning rate. With this we obtain

$$\begin{cases} \dot{w}_i = -P_i \zeta_i e_i \\ \dot{P}_i = -P_i \zeta_i \zeta_i^T P_i \end{cases} \quad i = 1, 2, \cdots, n \tag{4.19}$$

where $P(0)$ is a symmetric positive definite matrix. In the above formulation, the least-squares algorithm can be thought of as a gradient algorithm with a time-varying learning rate.

The stability and convergence properties of the weight adjustment laws given by (4.18),(4.19) are well-known in the adaptive control literature(see, for example, [48],[51]).

Theorem 2 *Consider the RHONN model given by*

$$x_i = w_i^T \zeta_i, \;\; i = 1, 2, \cdot\cdot\cdot, n, \tag{4.20}$$

whose parameters are adjusted according to:

$$\dot{w}_i = -\Gamma_i \zeta_i e_i, \ \ i = 1, 2, \cdots, n, \tag{4.21}$$

where Γ_i is a positive definite matrix referred to as the adaptive gain or learning rate.
Then for $i = 1, 2, \cdots, n$ it is proved that:

a) $e_i, \phi_i \in L_\infty$ (e_i and ϕ are uniformly bounded)
b) $\lim_{t\to\infty} e_i(t) = 0$

The proof of the above theorem can be studied in the Appendix.

Remark 1 *The stability proof for the least-square algorithm:*

$$\dot{w}_i = -P_i \zeta_i e_i \dot{P}_i = -P_i \zeta_i \zeta_i^T P_i \tag{4.22}$$

where $i = 1, 2, \cdots, n$, where $P(0)$ is a symmetric positive definite matrix. In the above formulation, the least-squares algorithm can be thought of as a gradient with a time-varying learning rate.

proceeds along the same lines as in the proof of the previous theorem by considering the Lyapunov function:

$$V = \frac{1}{2} \sum_{i=1}^{N} (\phi_i^T P_i^{-1} \phi_i + \int_t^\infty \epsilon_i^2(\tau) d\tau).$$

A problem that may be encountered in the application of the least-squares algorithm is that P may become arbitrarily small and thus slow down adaptation in some directions [50],[48]. This so-called problem can be prevented by using one of various modifications which prevent $P(t)$ form going to zero. One such modification is the so-called, where if the smallest eigenvalue of $P(t)$ becomes smaller than ρ_1 then $P(t)$ is reset to $P(t) = \rho_o I$, where $\rho_o \geq \rho_1 > 0$ are some design constraints.

Remark 2 *The above theorem does not imply that the weight estimation error $\phi_i = w_i - w_i^*$ converges to zero. In order to achieve convergence of the weights to their correct value the additional assumption of persistent excitation needs to be persistently exciting if there exist positive scalars c and d and T such that for all $t \geq 0$*

$$cI \le \int_{t}^{t+T} \zeta_i(\tau)\zeta_i(\tau)^T d\tau \le dI,$$

where is the LxL identity matrix.

Remark 3 *The learning algorithms developed above can be extended to the case where the underlying neuron structure is governed by the higher-order Cohen-Grossberg model [40],[43]:*

$$\dot{x}_i = -a_i(x_i)\left\{ b_i(x_i) + \sum_{k=1}^{L} w_{ik} \prod_{j \in I_k} y_j^{d_j(k)} \right\} \tag{4.23}$$

where $a_i(\cdot), b_i(\cdot)$ satisfy certain conditions required for the boundedness of the state variables [43]. It can be seen readily that in (4.23) the differential equation is still linear in the weights and hence a similar parameter estimation procedure can be applied.

The filtered-regressor RHONN model considered in this subsection relies on filtering the vector z, which is sometimes referred to as the regressor vector. By using this filtering technique, it is possible to obtain a very simple algebraic expression for the error, which allows the application of well-known optimization procedures for designing and analyzing weight adjustment laws but there is an important drawback to this method, namely the complex configuration and heavy computational demands required in the filtering of the regressor. Generally, the dimension of the regressor will be larger than the dimension of the system, i.e., $L > n$, it might be very expensive computationally to employ to many filters. In the next subsection a simple structure that requires only n filters is considered and hence, fewer computations.

4.3.2 Filtered Error RHONN

In developing this identification scheme the start is again from the differential equation that describes the unknown system, i.e.,

$$\dot{\chi}_i = -a_i\chi_i + w_i^{*T} z, \;\; i = 1, 2, \cdots, n. \tag{4.24}$$

Based on (4.24), the identifier is now chosen as:

$$\dot{x}_i = -a_i x_i + w_i^T z, \;\; i = 1, 2, \cdots, n. \tag{4.25}$$

where w_i is again the estimate of the unknown vector w_i^*. In this case the state error $e_i := x_i - \chi_i$ satisfies:

$$\dot{e}_i = -a_i e_i + \phi_i^T z, \ \ i = 1, 2, \cdots, n. \tag{4.26}$$

where $\phi_i = w_i - w_i^*$. The weights w_i, for $i = 1, 2, \ldots, n$ are adjustable according to the learning laws:

$$\dot{w}_i = -\Gamma_i z e_i, \tag{4.27}$$

where the adaptive gain Γ_i is a positive definite LxL matrix. In the special case that $\Gamma_i = \gamma_i I$, where $\gamma_i > 0$ is a scalar, then Γ_i in (4.27) can be replaced by γ_i.

The next theorem shows that the identification scheme has similar convergence properties as the filtered regressor RHONN model with the gradient method for adjusting the weights.

Theorem 3 *Consider the filtered error RHONN model given by (4.25) whose weights are adjustable according to (4.27). Then for* $i = 1, 2, \ldots, n$

(a) $e_i, \phi_i \in L_\infty$
(b) $\lim_{t\to\infty} e_i(t) = 0$

The proof of the above theorem can be studied in the Appendix.

4.4 Robust Learning Algorithms

[73]
The derivation of the learning algorithms developed in the previous section made the crucial assumption of no modeling error. Equivalently, it was assumed that there exist weight vectors w_i^*, for $i = 1, 2, \cdots, n$ such that each state of the unknown dynamical system (4.9) satisfies

$$\dot{\chi}_i = -a_i \chi_i + w_i^{*T} z(\chi, u) \tag{4.28}$$

In many cases this assumption will be violated. This is mainly due to an insufficient number of higher-order terms in the RHONN model. In such cases, if standard adaptive laws are used for updating the weights, then the presence of the modeling error in problems related to learning in dynamic environments, may cause the adjusted weight values (and, consequently, the error $e_i = x_i - \chi_i$) to drift to infinity. Examples of such behavior, which is usually referred to as, can be found in the adaptive control literature of linear systems [50],[51].

In this section the standard weight adjustment laws are being modified in order to avoid the parameter drift phenomenon. These modified weight adjustment laws will be referred to as *robust learning algorithms*.

In formulating the problem it is noted that by adding and subtracting $a_i\chi_i + w_i^{*T} z(\chi, u) + v_i(t)$, the dynamic behavior of each state of the system (4.9) can be expressed by a differential equation of the form:

$$\dot{\chi}_i = -a_i\chi_i + w_i^{*T} z(\chi, u) + v_i(t) \tag{4.29}$$

where the modeling error $v_i(t)$ is given by

$$v_i(t) := F_i(\chi(t), u(t)) + a_i\chi(t) - w_i^{*T} z(\chi(t), u(t)) \tag{4.30}$$

The function $F_i(\chi, u)$ denotes the $i{-}th$ component of the vector field $F(\chi, u)$, while the unknown optimal weight vector w_i^* is defined as the value of the weight vector w_i that minimizes the L_∞-norm difference between $F(\chi, u) + a_i\chi$ and $w_i^T z(\chi, u)$ for all $(\chi, u) \in \mathcal{Y} \subset R^{n+m}$, subject to the constraint that $|w_i| \leq M_i$, where M_i is a large design constraint. The region $\mathcal{Y}$ denotes the smallest compact subset of R^{n+m} that includes all the values that (χ, u) can take, i.e., $(\chi(t), u(t)) \in \mathcal{Y}$ for all $t \geq 0$. Since by assumption $u(t)$ is uniformly bounded and the dynamical system to be identified is BIBO stable, the existence of such $\mathcal{Y}$ is assured. It is pointed out that in our analysis we do not require knowledge of the region $\mathcal{Y}$, nor upper bounds for the modeling error $v_i(t)$.

In summary, for $i = 1, 2, \ldots, n$, the optimal weight vector w_i^* is defined as

$$w_i^* := \arg \min_{|w_i| \leq M_i} \left\{ \sup_{(\chi,u)\in\mathcal{Y}} |F_i(\chi, u) + a_i\chi - w_i^T z(\chi, u)| \right\} \tag{4.31}$$

The reason for restricting w_i^* to a ball of radius M_i is twofold: firstly, to avoid any numerical problems that m ay arise owing to having weight values that are too large, and secondly, to allow the use of the σ-modification [50], which will be developed below to handle the parameter drift problem. The formulation developed above follows the methodology of [30] closely. Using this formulation, we now have a system of the form (4.29) instead of (4.28). It is also noted that since $\chi(t)$ and $u(t)$ are bounded, the modeling error $v_i(t)$ is also bounded, i.e., $\sup_{t\geq 0}|v_i(t)| \leq 0$ for some finite constant $\bar{v}_i$.

In what follows robust learning algorithms based on the filtered error RHONN identifier are developed; however, the same underlying idea can be extended readily to the filtered-regressor RHONN. Hence, the identifier is chosen as in (4.25), i.e.,

$$\dot{x}_i = -a_i x_i + w_i^T z, \;\; i = 1, 2, \cdots, n \tag{4.32}$$

where w_i is the estimate of the unknown optimal weight vector w_i^*. Using (4.29),(4.32), the state error $e_i = x_i - \chi_i$ satisfies

$$\dot{e}_i = -a_i e_i + \phi_i^T z - v_i, \tag{4.33}$$

where $\phi_i = w_i - w_i^*$. Owing to the presence of the modeling error v_i, the learning laws given by (4.27) are modified as follows:

$$\dot{w}_i = \left\{ \begin{array}{c} -\Gamma_i z e_i, \;\; if|w_i| \leq M_i \\ -\Gamma_i z e_i - \sigma_i \Gamma_i w_i, \;\; if|w_i| > M_i \end{array} \right\} \tag{4.34}$$

where σ_i is a positive constant chosen by the designer. The above weight adjustment law is the same as (4.27) if w_i belongs to a ball of radius M_i. In the case that the weight leave this ball, the weight adjustment law is modified by the addition of the leakage term $\sigma_i \Gamma_i w_i$, whose objective is to prevent the weight values from drifting to infinity. This modification is known as the [50].

In the following theorem the vector notation $v := [v1, \cdots, v_n]^T$ and $e := [e_1, \cdots, e_n]^T$ is being used.

Theorem 4 *Consider the filtered error RHONN model given by (4.32) whose weights are adjusted according to (4.34). Then for* $i = 1, 2, \cdots, n$

$$\begin{gathered} (a)\ e_i, \phi_i \in L_\infty \\ (b) \text{there exist constants } \lambda, m \text{ such that} \\ \int\limits_0^t |e(\tau)^2| d\tau \le \lambda + m \int\limits_0^t |v(\tau)^2| d\tau \end{gathered}$$

The proof of the theorem can be studied in the Appendix.

Remark 4 *It is noted that the* σ *modification causes the adaptive law (4.34) to be discontinuous; therefore standard existence and uniqueness results of solutions to differential equations are in general not applicable. In order to overcome the problem of existence and uniqueness of solutions, the trajectory behavior of* $w_i(t)$ *can be made "smooth" on the discontinuity hypersurface* $\{w_i \in R^L : |w_i| = M_i\}$ *by modifying the adaptive law (4.34) to*

$$\dot{w}_i = \left\{ \begin{array}{c} -\Gamma_i z_i e_i, \ \ if \ \ \{|w_i| < M_i\} \ \ or \ \ \{|w_i| = M_i \ \ and \ \ w_i^T \Gamma_i z e_i > 0\} \\ \frac{-\Gamma_i z e_i + w_i^T \Gamma_i z e_i}{w_i^T \Gamma_i w_i} \Gamma_i w_i \ \ , if \ \ \{|w_i| = M_i\} \ \ and \ \ \{-\sigma_i w_i^T \Gamma_i w \le w_i^T \Gamma_i z e_i \le 0\} \\ -\Gamma_i z e_i - \sigma_i \Gamma_i w_i \ \ , if \ \ \{|w_i| > M_i\} \ \ or \ \ \{|w_i| = M_i\} \ \ and \ \ \{w_i^T \Gamma_i z e_i < -\sigma_i w_i^T \Gamma_i w\} \end{array} \right\} \tag{4.35}$$

As shown in [52], the adaptive law (4.35) retains all the properties of (4.34) and, in addition, guarantees the existence of a unique solution, in the sense of Caratheodory [49]. The issue of existence and uniqueness of solutions in adaptive systems is treated in detail in [52].

Chapter 5

Adaptive Fuzzy Systems

5.1 Notion and Representation of Adaptive Fuzzy Systems

Fuzzy Sets and Systems have gone through substantial development since the introduction of fuzzy set theory by Zadeh [68], [121], [122], [123] about four decades ago. They have found a great variety of applications ranging from control engineering, qualitative modeling, signal processing, machine intelligence, decision making, motor industry, robotics and so on [105], [124], [125].

Following a similar idea in neural networks for their universal function approximation capability [81], it is shown [107] that a fuzzy system is capable of approximating any smooth nonlinear functions over a convex compact region. Fuzzy basic function based fuzzy systems are used to represent those unknown nonlinear functions. The parameters of the fuzzy systems including membership functions characterizing linguistic terms in fuzzy rules are updated according to some adaptive laws which are derived based on Lyapunov Stability Theorem [2], [4], [12], [78], [94], [126], [127], [133].

The performance, complexity and adaptive law of an adaptive fuzzy system representation can be quite different depending upon the type of the fuzzy system (Mamdani [111], or Takagi-Sugeno [106]). It also depends upon whether the representations is linear or nonlinear in its adjustable parameters. Adaptive fuzzy controllers depend also on the type of the adaptive fuzzy subsystems they use. Suppose that the adaptive fuzzy system is intended to approximate the nonlinear function $f(x)$. In the Mamdani type, linear in the parameters form, the following fuzzy logic representation is used [78], [100]:

$$f(x) = \sum_{l=1}^{M} \theta_l \xi_l(x) = \theta^T \xi(x) \tag{5.1}$$

where M is the number of fuzzy rules, $\theta = (\theta_1, \theta_2, ..., \theta_M)^T$, $\xi(x) = (\xi_1(x), \xi_2(x), ...\xi_M(x))^T$ and $\xi_l(x)$ is the fuzzy basis function defined by

$$\xi_l(x) = \frac{\prod_{i=1}^{n} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{F_i^l}(x_i)} \tag{5.2}$$

θ_l are adjustable parameters, and $\mu_{F_i^l}$ are given membership functions of the input variables (can be gaussian, triangular, or any other type of membership functions).

In Takagi-Sugeno formulation $f(x)$ is given by

$$f(x) = \sum_{l=1}^{M} g_l(x)\xi_l(x) \tag{5.3}$$

where $g_l(x) = \alpha_{l,0} + \alpha_{l,1}x_1 + ... + \alpha_{l,n}x_n$, with x_i, $i = 1...n$ being the elements of vector x and $\xi_l(x)$ being defined in (5.2). According to Passino and Yurkovich [100], (5.3) can also be written in the linear to the parameters form, where the adjustable parameters are all $\alpha_{l,i}$, $l = 1...M$, $i = 1...n$.

From the above definitions it is apparent that in both, Mamdani and Takagi-Sugeno forms the success of the adaptive fuzzy system representations in approximating the nonlinear function $f(x)$ depend on the careful selection of the fuzzy partitions of input and output variables. Also, the selected type of the membership functions and the proper number of fuzzy rules contribute to the success of the adaptive fuzzy system. This way, any adaptive fuzzy or NF approach, following a linear in the adjustable parameters formulation becomes vulnerable to initial design assumptions related to the fuzzy partitions and the membership functions chosen. In this thesis, this quite huge for the applications drawback is largely overcome by using the concept of rule indicator functions, which are in the sequel approximated by high-order neural networks approximators (HONNs). In this way there is not any need for initial design assumptions related to the membership values and the fuzzy partitions of the if part [59].

More precisely, the underlying fuzzy model that is used in this thesis is the centroid of area defuzzification procedure of Mamdani type which is explained in the following subsection.

5.2 Centroid of Area Defuzzification

The centroid of area method is the most prevalent and physically appealing of all the defuzzification methods. Final crisp output when using centroid defuzzification is equal to weighted average of centroid of consequents membership functions [128] and is given as:

$$f(x) = \frac{\sum_{i=1}^{r} \omega_i(x) v_i \bar{x}_{f_i}}{\sum_{i=1}^{r} \omega_i(x) v_i} \tag{5.4}$$

where ω_i is the firing strength of the $i-th$ rule, v_i is the area of the consequent membership functions of $i-th$ rule and $\bar{x}_{f_i}$ is the $i-th$ fuzzy center of the consequent membership function.

In a more detailed form, we can rewrite Eq.(5.4) as:

$$f(x) = \frac{\omega_1(x)v_1}{\sum_{i=1}^{r} \omega_i(x)v_i}\bar{x}_{f_1} + \cdots + \frac{\omega_r(x)v_r}{\sum_{i=1}^{r} \omega_i(x)v_i}\bar{x}_{f_r} \tag{5.5}$$

Now, if it is choosen an indicator function as: $I_i(x) = \omega_i(x)v_i$ then Eq.(5.5) can be written as:

$$f(x) = \frac{I_1(x)}{\sum_{i=1}^{r} I_i(x)}\bar{x}_{f_1} + \cdots + \frac{I_r(x)}{\sum_{i=1}^{r} I_i(x)}\bar{x}_{f_r} \tag{5.6}$$

Hence, if it is defined $(I')_i(x) = \frac{I_i(x)}{\sum_{i=1}^{r} I_i(x)}$ as the weighted indicator function (WIF), the system output finally is given by the following form:

$$f(x) = \sum_{i=1}^{r} (I')_i \cdot \bar{x}_{f_i} \tag{5.7}$$

Eq. (5.7) provides the functional representation of the fuzzy system in terms of WIF [59]. This representation is used in the sequel for deriving the NF approach.

Chapter 6

F-RHONNs Model

6.1 NF Model

Let us consider a nonlinear function $f(x,u)$, where $f : R^{n+m} \rightarrow R^n$ is a smooth vector field defined on a compact set $\Psi \subset R^{n+m}$, with input space $u \in U_c \subset R^m$ and state-space $x \in X \subset R^n$. Also, we assume that the dynamic equation which describes the i/o behavior of a system has the following form:

$$\dot{x}(t) = f(x(t), u(t)), \tag{6.1}$$

or in a per-state form:

$$\dot{x}_i(t) = f(x_i(t), u_i(t)), \tag{6.2}$$

where $f_i(\cdot), i = 1, 2, ..., n$, is a continuous function and t denotes the temporal variable. In order to proceed further we have the following assumption [59].

> Assumption 1: *Notice that since $\Psi \subset R^{n+m}$ then Ψ is closed and bounded set. Also, it is noted that even if Ψ is not compact we may assume that there is a time instant T such that $(x(t), u(t))$ remain in a compact subset of Ψ for all $t < T$;i.e. if $\Psi_T := (x(t), u(t)) \in \Psi, t < T$. The interval Ψ_T represents the time period over which the approximation is to be performed.*

It is considered, that function $f(x,u)$ is approximated by a fuzzy system using appropriate fuzzy rules. In this framework let Ω_f be defined as the universe of discourse of $(x,u) \in X \cup U \subset R^{n+m}$ belonging to the $(j_1, j_2, ..., j_{n+m})^{th}$ input fuzzy patch and pointing-through the vector field $f(\cdot)$ - to the subset which belong to the $(l_1, l_2, ..., l_n)^{th}$ output fuzzy patch. Also, Ω_{f_i} is a subset of Ω_f containing input pairs values associated with f_i. Furthermore, $\Omega^p_{f_i}$, with $p = 1, 2, ..., q$, q being the number of fuzzy partitions of the $i-th$ state variable, is defined as the $p-th$ subregion of Ω_{f_i} such that $\Omega_{f_i} = \cup^q_{p=1}\Omega^p_{f_i}$.

> Definition 1: *According to the above notation the indicator function (IF) connected to $\Omega^p_{f_i}$ is defined as follows:*
>
> $$I^p_i(x(t), u(t)) = \begin{Bmatrix} \alpha^p_i(x(t), u(t)) & if\ (x(t), u(t)) \in \Omega^p_{f_i} \\ 0 & otherwise \end{Bmatrix} \tag{6.3}$$
>
> *where $\alpha^p_i(x(t), u(t))$ denotes the firing strength of the rule*[59].

Then, assuming a standard defuzzification procedure (e.g. centroid of area or weighted average), the functional representation of the fuzzy system that approximates the real one can be written as:

$$\hat{f}_i(x(t), u(t)) = \frac{\sum_{p=1}^{q} I_i^p \cdot \bar{x}_{f_i}^p}{\sum_{p=1}^{q} I_i^p} \tag{6.4}$$

where the summation is carried over all the available fuzzy rules [59].

Definition 2: *We can define the weighted IF (WIF) by the following equation:*

$$(I')_i^p = \frac{I_i^p}{\sum_{p=1}^{q} I_i^p} \tag{6.5}$$

which is the IF defined in (6.3) divided by the sum of all IF participating in the summation of (6.4).

Thus, Eq.(6.4) can be rewritten as:

$$\hat{f}_i(x(t), u(t)) = \sum_{p=1}^{q} (I')_i^p \cdot \bar{x}_{f_i}^p \tag{6.6}$$

Based on the fact that functions of high order neurons are capable of approximating discontinuous functions [19], [20], high order neural networks (HONNs) are used in order to approximate a $(I')_i^p$. A HONN is defined as:

$$N_i^p(x(t), u(t); w, k) = \sum_{l=1}^{k} w_{f_i}^{pl} \prod_{j \in I_l} \Phi_j^{d_j(l)}, \tag{6.7}$$

where $I_l = I_1, I_2, ..., I_k$ is a collection of k not-ordered subsets of $1, 2, ..., n+m$, $d_j(l)$ are non-negative integers. Φ_j are the elements of the following vector,

$$\Phi = [\Phi_1 ... \Phi_n \Phi_{n+1} ... \Phi_{n+m}]^T = [s(x_1) ... s(x_n) s(u_1) ... s(u_m)]^T,$$

where s denotes the sigmoid function defined as:

$$s(x) = \frac{\alpha}{1 + e^{-\beta x}} - \gamma, \tag{6.8}$$

with α, β being positive real numbers and γ being a real number. Special attention, has to be given in the selection of parameters α, β, γ so that $s(x)$ fulfil the persistency of excitation condition ($s \in [-\gamma, -\gamma + \alpha]$) $when\, \gamma > 0$ required in some system identification tasks. Also, $w_{f_i}^{pl}$ is the HONN weights with $i = 1, 2, ..., n,\ p = 1, 2, ..., q,\ and\ l = 1, 2, ..., k$. Thus, Eq.(6.7) can be written as:

$$N_i^p(x(t), u(t); w, k) = \sum_{l=1}^{k} w_{f_i}^{pl} s_l(x(t), u(t)), \tag{6.9}$$

where $s_l(x(t), u(t))$ are high order terms of sigmoid functions of the state and/or input [59].

The next lemma [59] states that a HONN of the form in Eq.6.9 can approximate the weighted indicator function (WIF), $(I')_i^p$.

> Lemma 1: *Consider the WIF $(I')_i^p$ and the family of HONNs $N_i^p(x(t), u(t); w, k)$. Then for any ϵ_i^p, there is a vector of weights w and a number of k high order connections such that:*
>
> $$\sup_{(x(t),u(t))\in\Psi} (I')_i^p(x(t), u(t)) - \textstyle\sum_{l=1}^{k} w_{f_i}^{pl} s_l(x(t), u(t)) \leq \epsilon_i^p$$
>
> *The magnitude of approximation error $\epsilon_i^p \geq 0$ depends on the choice of the member of high order terms.*

Furthermore, someone can say that we are provided with rules of the form:

$$R_i^p : \ IF \ (x(t), u(t)) \in \Omega_{f_i}^p \ THEN \ HONN_p \ is \ (I')_i^p(t)$$

Following the above analysis, actually it is given a membership value according to the output fuzzy partitioning to every HONN which participates to the estimation of $f_i(x, u)$ [59].

As a consequence we have the following definition.

> Definition 3: *The center membership value (CMV) $\bar{x}_{f_i}^p$ which is the $p - th$ fuzzy center of the $i - th$ state variable (or equivalently f_i) influences a HONN by a degree of implementation $\bar{x}_{f_i}^p$.*

Therefore, rule R_i^p can be equivalently expressed as:

$$R_i^p : \ IF \ (x(t), u(t)) \in \Omega_{f_i}^p \ THEN \ HONN_p \ is \ (I')_i^p(t) \ withCMV \ \bar{x}_{f_i}^p$$

Now, the rules which participate to the construction of the $i - th$ state variable output can be grouped according to the following form:

$$R_i^p: IF\,(x(t),u(t)) \in$$
$$\Omega_{f_i}\, THEN\, HONN_1\, is\,(I')_1^p(t)\ with\,CMV\ \bar{x}_{f_i}^1\ and HONN_2$$
$$is\,(I')_i^2(t)\, with\,CMV\ \bar{x}_{f_i}^2\ and...and\, HONN_q\, is\,(I')_i^q(t)\, with\,CMV\ \bar{x}_{f_i}^q.$$

As it is clear enough by the above fuzzy rule definition, the $i-th$ state variable of the system output is determined as follows:

$$R_i^p:\ IF\ (x(t),u(t)) \in \Omega_{f_i}\ THEN\ f_i(x,u) = (I')_i^1(t)\cdot \bar{x}_{f_i}^1 + ... + (I')_i^q(t)\cdot \bar{x}_{f_i}^q$$

where it is clear enough that the information about the antecedent partitioning of the rules as well as the number of rules is not necessary here to be determined. Therefore, the rules are not treated here with the classical way of Mamdani or Takagi-Sugeno definition but their consequent parts are determined directly from F-HONNs.

Following the above notation, Eq. (6.6) in conjunction with Eq. (6.9) can be rewritten as:

$$\hat{f}_i(x(t),u(t)) = \sum_{p=1}^{q} \bar{x}_{f_i}^p \cdot (\sum_{l=1}^{k} w_{f_i}^{pl}\cdot s_l(x(t),u(t))), \tag{6.10}$$

or in a more compact form:

$$\dot{\hat{f}} = X_f \cdot W_f \cdot s_f(x,u) \tag{6.11}$$

An alternative, recurrent NF form of Eq. (6.2) which will be used in the subsequent analysis of the thesis is:

$$\dot{\hat{x}} = A\hat{x} + \hat{f}. \tag{6.12}$$

Considering that f is approximated by the NF model described above, Eq. (6.12) can be rewritten as:

$$\dot{\hat{x}} = A\hat{x} + X_f W_f s_f(x,u), \tag{6.13}$$

where A is a $n\times n$ stable matrix which for simplicity can be taken to be diagonal as $A = diag[-\alpha_1, -\alpha_2, ..., -\alpha_n]$, with $\alpha_i > 0$. Also, X_f is a matrix containing the centers of the partitions of every fuzzy output variable of $f(x,u)$, $s_f(x,u)$ is a vector containing high order combinations of sigmoid functions of the state x and control input u. Also, W_f is a matrix containing respective neural weights according to (6.9) and (6.10). For notational simplicity we assume that all output fuzzy variables are partitioned to the same number, q, of partitions. Under these specifications X_f is a $n\times n\cdot q$ block diagonal matrix of the form $X_f = diag(\bar{x}_{f_1}, \bar{x}_{f_2}, ..., \bar{x}_{f_n})$ with $\bar{x}_{f_i}$ being a $q-dimensional$ row vector of the form:

$$\bar{x}_{f_i} = [\bar{x}_{f_i}^1 \bar{x}_{f_i}^2 \cdots \bar{x}_{f_i}^q].$$

Also, $s_f(x) = [s_1(x)s_2(x)...s_k(x)]^T$, where each $s_l(x)$ with $l = 1, 2, ..., k$ is a high order combination of sigmoid functions of the state variables and input signals. Finally, W_f is a $n \cdot q \times k$ matrix with neural weights. W_f is assumes the form $W_f = [W_{f_1} W_{f_2} \cdots W_{f_n}]^T$, where each W_{f_i} is a matrix $[w^{pl}_{f_i}]_{q \times k}$.

From the above definitions and Eq. (6.9) it is obvious that the accuracy of the approximation of $f_i(x, u)$ depends on the approximation abilities of HONNs and on an initial estimate of the centers of the output membership functions. These centers can be obtained by experts or by off-line techniques based on gathered data. Any other information related to the input membership functions is not necessary because it is replaced by the HONNs [59].

In the next chapter we overcome the need of the centers estimation by the experts by introducing a bilinear parameter estimation algorithm. All we need is an initial estimate of the centers of the partitions of the output membership functions and in the sequel, our algorithm estimate them automatically. This is one of the contribution of this master thesis.

Chapter 7

Bilinear Neuro-Fuzzy Indirect Adaptive Control of Unknown Nonlinear Dynamical Systems

7.1 Neuro-Fuzzy Representation and Identification

At first we consider affine in the control, nonlinear ,in general, dynamical systems of the form

$$\dot{x} = f(x) + G(x) \cdot u \tag{7.1}$$

where the state $x \in R^n$ is assumed to be completely measured, the control signal u is in R^n, f is an unknown smooth vector field which is called the drift term and G ia a matrix with columns the unknown smooth controlled vector fields g_i, $i = 1, 2, ..., n$ and $G = [g_1, g_2, \ldots, g_n]$. The above class of continuous-time nonlinear systems are called affine, because in (7.1) the control input appears linear with respect to G. The main reason for considering this class of nonlinear systems is that most of the systems encountered in engineering, are by nature or technical design, affine. Furthermore, we note that non affine systems of the form given in (2.1) can be converted into affine, by passing the input through integrators, a procedure which is widely known as dynamic extension.

In our approach, referred to as indirect adaptive fuzzy-HONNF control, the parameters of the plant are estimated on-line except of the fuzzy partitions which are used to calculate the controller parameters. The following mild assumptions are also imposed on (7.1), to guarantee the existence and the uniqueness of solution for any finite initial condition and $u \in U$.

Proposition: Given a class U of admissible inputs, then for any $u \in U$ and any finite initial condition, the state trajectories are uniformly bounded for any finite $T > 0$. Hence, $|x(T)| < \infty$.

Proposition: The vector fields f, g_i, $i = 1, 2, ..., n$ are continuous with respect to their arguments and satisfy a local Lipchitz condition so that the solution $x(t)$ of (7.1) is unique for any finite initial condition and $u \in U$.

Following the analysis of the privious section, we are using an affine in the control fuzzy dynamical system, which approximates the system in (7.1) and uses two fuzzy subsystem blocks for the description of $f(x)$ and $G(x)$ as follows:

$$f(\chi) = A\chi + \sum \bar{f}^{l_1,\ldots,l_n}_{j_1,\ldots,j_n} \times I^{l_1,\ldots,l_n}_{j_1,\ldots,j_n}(\chi) \tag{7.2}$$

$$g_i(\chi) = \sum (\bar{g}_i)^{l_1,\ldots,l_n}_{j_1,\ldots,j_n} \times I1^{l_1,\ldots,l_n}_{j_1,\ldots,j_n}(\chi) \tag{7.3}$$

where the summation is carried out over the number of all available fuzzy rules, I, I_1 are appropriate fuzzy rule indicator functions and the meaning of indices $\bullet_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}$ has already been described in Section 6.1.

According to Lemma for indicator HONNF's, every indicator function can be approximated with the help of a suitable HONNF. Therefore, every I, I_1 can be replaced with a corresponding HONNF as follows:

$$f(\chi) = A\chi + \sum \bar{f}_{j_1,\ldots,j_n}^{l_1,\ldots,l_n} \times N_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}(\chi) \tag{7.4}$$

$$\bar{g}_i(\chi) = \sum (\bar{g}_i)_{j_1,\ldots,j_n}^{l_1,\ldots,l_n} \times {N_1}_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}(\chi) \tag{7.5}$$

where N, N_1 are appropriate HONNFs.

In order to simplify the model structure, since some rules result to the same output partition, we could replace the NNs associated to the rules having the same output with one NN and therefore the summations in (7.4),(7.5) are carried out over the number of the corresponding output partitions. Therefore, the affine in the control fuzzy dynamical system in (7.2), (7.3) is replaced by the following equivalent affine Recurrent High Order Neural Network (RHONN), which depends on the centers of the fuzzy output partitions $\bar{f}_l$ and $\bar{g}_{i,l}$

$$\dot{\hat{\chi}} = A\hat{\chi} + \sum_{l=1}^{Npf} \bar{f} \times N_l(\chi) + \sum_{i=1}^{n} \left(\sum_{l=1}^{Npg_i} (\bar{g}_i)_l \times N_{1l}(\chi) \right) u_i \tag{7.6}$$

Or in a more compact form

$$\dot{\hat{\chi}} = A\hat{\chi} + XWS(\chi) + X_1 W_1 S_1(\chi) u \tag{7.7}$$

Where A is a $n \times n$ stable matrix which for simplicity can be taken to be diagonal as $A = diag[a_1, a_2, ..., a_n]$, X, X_1 are matrices containing the centres of the partitions of every fuzzy output variable of $f(x)$ and $g(x)$ respectively, $S(\chi), S_1(\chi)$ are matrices containing high order combinations of sigmoid functions of the state χ and W, W_1 are matrices containing respective neural weights according to (7.6) and (7.7). The dimensions and the contents of all the above matrices are chosen so that $XWS(\chi)$ is a $n\times 1$ vector and $X_1W_1S_1(\chi)$ is a $n\times n$ matrix. Without compromising the generality of the model we assume that the vector fields in (7.3) are such that the matrix G is diagonal. For notational simplicity we assume that all output fuzzy variables are partitioned to the same number, m, of partitions. It should be noted that each output fuzzy variable may have a different number of let's say m_i partitions where

$$k = \sum_{i=1}^{n} m_i$$

Then the matrix X is of dimension $n \times k$ and is block diagonal. Without loss of generality (the same results are true for non-equal partition-numbers for each variable), X is a $n \times n \cdot m$ block diagonal matrix of the form $X = diag(X^1, X^2, \dots, X^n)$ with each X^i being an m-dimensional raw vector of the form

$$X^i = \begin{bmatrix} \bar{f}_1^i & \bar{f}_2^i & \cdots & \bar{f}_m^i \end{bmatrix}$$

where $\bar{f}_p^i$ denotes the centre of the $p-th$ partition of f_i. Also, $S(\chi) = \begin{bmatrix} s_1(\chi) & \dots & s_k(\chi) \end{bmatrix}^T$, where each $s_i(\chi)$ with $i = \{1, 2, ..., k\}$, is a high order combination of sigmoid functions of the state variables and W is a $n \cdot m \times k$ matrix with neural weights. W assumes the form $W = \begin{bmatrix} W^1 & \cdots & W^n \end{bmatrix}^T$, where each W^i is a matrix $\begin{bmatrix} w_{j\,l}^i \end{bmatrix}_{m\times k}$. Also, X_1 is a $n \times n \cdot m$ block diagonal matrix $X_1 = diag({}^1X^1, {}^1X^2, \dots, {}^1X^n)$ with each ${}^1X^i$ being an m-dimensional raw vector of the form

$${}^1X^i = \begin{bmatrix} \bar{g}_1^{i,i} & \bar{g}_2^{i,i} & \cdots & \bar{g}_m^{i,i} \end{bmatrix},$$

where $\bar{g}_k^{i\,i}$ denotes the center of the k-th partition of g_{ii}. W_1 is a $m \cdot n \times n$ block diagonal matrix $W_1 = diag({}^1W^1, {}^1W^2, \dots, {}^1W^n)$, where each ${}^1W^i$ is a column vector $\begin{bmatrix} {}^1w_{j\,l}^i \end{bmatrix}_{m\times 1}$ of neural weights. Finally, $S_1(\chi)$ is a $n \times n$ diagonal matrix with each diagonal element $s_i(\chi)$ being a high order combination of sigmoid functions of the state variables.

According to the above definitions the configuration of the F-HONNF approximator is shown in Fig. (7.1). When the inputs are given into the fuzzy-neural network shown in Fig. (7.1), the output of layer IV gives indicator function outputs which activate the corresponding rules and are calculated by Eq. (6.9). At layer V, each node performs a fuzzy rule while layer VI gives the function output.

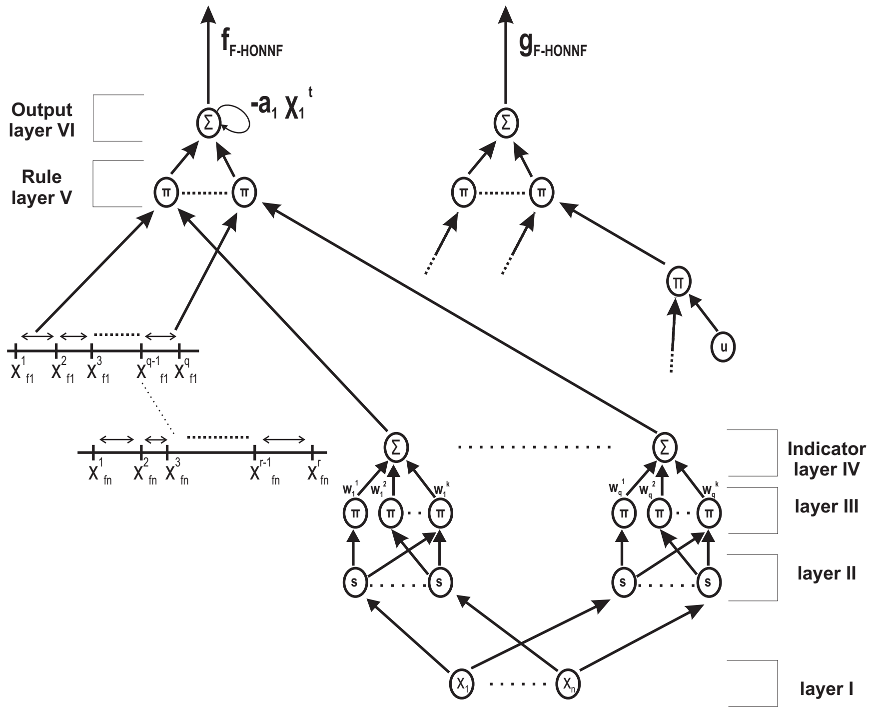

Figure 7.1: Overall scheme of indirect adaptive neuro-fuzzy approximator [54]

The approximator of indicator functions, has four layers. At layer I, the input nodes represent input and/re state measurable variables. At layer II, the nodes represent the values of the sigmoidal functions. At layer III, the nodes are the values of high order sigmoidal combinations. The links between layer III and layer IV are fully connected by the weighting factors $W = \begin{bmatrix} W^1 & \cdots & W^n \end{bmatrix}^T$, the adjusted parameters. Finally, at layer IV the output represents the values of indicator functions.

7.2 Parametric and partition centers uncertainty

We assume the existence of uncertainty in the partition centers and parameter weight uncertainty, so, we can take into account that the actual system (7.1) can be modeled by the following neural form:

$$\dot{\chi} = A\chi + X^*W^*S(\chi) + X_1^*W_1^*S_1(\chi)u \tag{7.8}$$

Define now, the error between the identifier states and the real states as

$$e = \hat{\chi} - \chi \tag{7.9}$$

Then from (7.7) and (7.8) we obtain the error equation

$\dot{e} = Ae + X^*W^*S(\chi) + X_1^*W_1^*S_1(\chi)u$

$-XWS(\chi) - X_1W_1S_1(\chi)u$

To this end add and subtract to the above error equation the terms $X^*WS(\chi)$ and $X_1^*W_1S(\chi))u$

and define: $\tilde{W} = W - W^*$ and $\tilde{W}_1 = W_1 - W_1^*$.

Then the error equation becomes:

$$\dot{e} = Ae - X^*\tilde{W}S(\chi) - \tilde{X}WS(\chi) - X_1^*\tilde{W}_1S_1(\chi)u - \tilde{X}_1W_1S_1(\chi)u \tag{7.10}$$

Our objective is to find suitable control and learning laws to drive both e and χ to zero, while all other signals in the closed loop remain bounded. Taking u to be equal to

$$u = -\left[X_1W_1S_1(\chi)\right]^{-1} XWS(\chi) \tag{7.11}$$

and substituting it into (7.7) we finally obtain

$$\dot{\hat{\chi}} = A\hat{\chi} \tag{7.12}$$

In the next theorem the weight and partitions centers update laws are given, which can serve identification and control objectives, provided the updating of the weights of matrices X_1 and W_1 is performed in such a way, so that the existence of $[X_1 W_1 S_1(\chi)]^{-1}$ is assured.

Theorem 5 *Consider the identification scheme given by (7.10). Provided that $[X_1 W_1 S_1(\chi)]^{-1}$ exists the learning laws:*

a) For the elements of W and X

$$\begin{cases} \dot{W} = sgn(X^*)^T P e S^T \\ \dot{X} = P e S^T W^T \end{cases} \tag{7.13}$$

b) For the elements of W_1 and X_1

$$\begin{cases} \dot{W}_1 = sgn(X_1^*)^T P e u^T S_1^T \\ \dot{X}_1 = P e u^T S_1^T W_1^T \end{cases} \tag{7.14}$$

guarantee the following properties.

- $e, \hat{\chi}, \tilde{W}, \tilde{W}_1, \tilde{X}, \tilde{X}_1 \in L_\infty, \quad e, \hat{\chi} \in L_2$
- $\lim_{t\to\infty} e(t) = 0, \quad \lim_{t\to\infty} \hat{\chi}(t) = 0$
- $\lim_{t\to\infty} \dot{\tilde{W}}(t) = 0, \quad \lim_{t\to\infty} \dot{\tilde{W}}_1(t) = 0$
- $\lim_{t\to\infty} \dot{\tilde{X}}(t) = 0, \quad \lim_{t\to\infty} \dot{\tilde{X}}_1(t) = 0$

where the matrices $sgn(X^*)$ and $sgn(X_1^*)$ are defined in the proof.

The proof can be studied in the Appendix.

7.3 A Novel Approach of Parameter Hopping

The weight updating laws presented previously are valid when the control law signal in 7.11 exists. Therefore, the existence of $[X_1 W_1 S_1(\chi)]^{-1}$ has to be assured. Since $S_1(\chi)$ is diagonal with its elements $s_i(\chi) \neq 0$ and X_1,W_1 are block diagonal the existence of the inverse is assured when $X_1^i \cdot W_1^i \neq 0, \ \forall i = 1, ..., n$. Therefore, we have to assure that $|X_1^i \cdot W_1^i| \geq \theta_i > 0$, with θ_i being a design parameter. In our case where both terms change, we shall introduce a specific update of both laws, one at a time, for keeping the above inequality true at all times. The idea is the following : we keep one term fixed and we change appropriately the other one. Then, we examine whether the above inequality of θ_i is still true. If not, we keep the first term fixed and we update the second one. The first one term that we keep fixed is matrix X_1^i and we update the neural weights of the matrix W_1^i, and that is because due to the initial estimation of the centers of the output membership functions by the experts, we consider them more difficult to change dramatically from their initial values (given by the experts) and we "pay more attention" by switching appropriately the vectors in matrix W_1^i.

The logic behind the appropriate change of our update laws is the same for both terms X_1^i, W_1^i who change when our condition of distance is being satisfied. So, using concepts from multidimensional geometry we modify the updating laws such that, when the vector approaches, within a safe distance θ_i , the forbidden hyper-plane $X_1^i \cdot W_1^i = 0$ and the direction of updating is toward the forbidden hyper-plane, it introduces a *hopping* which drives the vector in the direction of updating but on the other side of the space, where here the space is divided for each modification of each term into two sides by the forbidden hyper-plane. Theorem below introduces this hopping in the update laws respectively.

Theorem 6 *Consider the control scheme (7.10), (7.11), (7.12). The updating law:*

For the elements of ${}^1W^i$ *given by the modified form:*

$$
\begin{aligned}
&{}^1\dot{W}^i = sign({}^1X^{*i})^T p_i e_i u_i s_i(\chi) \quad if \ \left|{}^1X^i \cdot {}^1W^i\right| > \theta_i > 0 \\
&or \quad \left|{}^1X^i \cdot {}^1W^i\right| = \theta_i \ and \quad {}^1X^i \cdot {}^1\dot{W}^i \leq 0
\end{aligned}
$$

$$
{}^1\dot{W}^i = sign({}^1X^{*i})^T p_i e_i u_i s_i(\chi) -
$$

$$
- \frac{2}{tr\{({}^1X^i)^T \ {}^1X^i\}} \ {}^1X^i \, {}^1W^i \ ({}^1X^i)^T \quad otherwise
$$

guarantees the properties of theorem 5 and assures the existence of the control signal.

Theorem 7 *Consider the control scheme (7.10), (7.11), (7.12). The updating law:*

For the elements of ${}^1X^i$ *given by the modified form:*

$$
{}^1\dot{X}^i = ({}^1W^i)^T p_i e_i u_i s_i(\chi) \quad if \ \left|{}^1X^i \cdot {}^1W^i\right| > \theta_i > 0
$$
$$
or \quad \left|{}^1X^i \cdot {}^1W^i\right| = \theta_i \ and \quad {}^1\dot{X}^i \cdot {}^1W^i \leq 0
$$

$$
{}^1\dot{X}^i = ({}^1W^i)^T p_i e_i u_i s_i(\chi) -
$$
$$
- \frac{2}{tr\{({}^1W^i)^T \, {}^1W^i\}} \, {}^1X^i \, {}^1W^i \, ({}^1W^i)^T \quad otherwise
$$

guarantees the properties of theorem 5 and assures the existence of the control signal.

The proof, which follows the exact same steps for both terms can be studied in the Appendix.

Chapter 8

Simulation of a DC Motor System - Results

8.1 Simulation of a DC Motor

In this section we apply the proposed approach to solve the problem of controlling the speed of a 1 KW DC Motor with a normalized model described by the following dynamical equations

$$\begin{aligned} T_a \tfrac{dI_a}{dt} &= -I_a - \Phi\Omega + V_a \\ T_m \tfrac{d\Omega}{dt} &= \Phi I_a - K_0\Omega - m_L \\ T_f \tfrac{d\Phi}{dt} &= -I_f + V_f \\ \Phi &= \tfrac{aI_f}{1+bI_f} \end{aligned} \tag{8.1}$$

$$\tag{8.2}$$

Traditionally, the Angular Velocity of a DC Motor is controlled with changes in its armature voltage, while keeping constant the field excitation. Thus, the above nonlinear model is linearized and reduced to

$$\begin{aligned} T_a \tfrac{dI_a}{dt} &= -I_a - \Phi\Omega + V \\ T_m \tfrac{d\Omega}{dt} &= \Phi I_a - K_0\Omega - m_L \end{aligned} \tag{8.3}$$

now with Ω a constant value parameter.

So, the regulation problem of a DC Motor is translated as follows: Find a state feedback to force the Angular Velocity Ω and the Armature Current I_a to go to zero, while the Magnetic Flux varies.

To achieve such a goal, assuming that the dynamics of the system are unknown, we first assume that the system is described, within a degree of accuracy, by a Neuro-Fuzzy system of the form

$$\dot{\hat{\chi}} = A\hat{\chi} + XWS(\chi) + X_1W_1S_1(\chi)u \tag{8.4}$$

Where A is a $n \times n$ stable matrix which for simplicity can be taken to be diagonal as $A = diag[a_1, a_2, ..., a_n]$, X, X_1 are matrices containing the centers

Table 8.1: Parameter values for the DC motor.

Parameter	Value
$1/T_a$	148.88
$1/T_m$	42.91
K_0/T_m	0.0129
T_f	31.88
T_L	0.0
a	2.6
β	1.6

of the partitions of every fuzzy output variable of $f(x)$ and $g(x)$ respectively, $S(\chi), S_1(\chi)$ are matrices containing high order combinations of sigmoid functions of the state χ and W, W_1 are matrices containing respective neural weights.

The number of states being $n = 2$, the number of fuzzy partitions being $m = 5$ and the depth of high order sigmoid terms $k = 2$. In this case $s_i(x)$ assumes high order connection up to the second order.

Also, to regulate the motor speed to zero we apply the control law

$$u = -\left[X_1 W_1 S_1(\chi)\right]^{-1} X W S(\chi) \tag{8.5}$$

where the number of fuzzy partitions of each $g_{i\,i}$ is $m = 3$.

We simulated a 1 KW DC Motor with parameter values that can be seen in Table 8.1. Our two stage algorithm, was applied.

We considered the identification procedure known and for the indirect adaptive control we used the following values

For the block diagonal matrix $X = diag(X^1, X^2, \cdots, X^n)$ with $n = 2$ and the sub-matrices X^1, X^2, we gave the following values for the centers of the fuzzy partitions:

$$X = \begin{bmatrix} X^1 & 0 \\ 0 & X^2 \end{bmatrix}$$

where the centers of the fuzzy partitions of X^1, X^2 are the following:

$$X^1 = \begin{bmatrix} -163.3061 & -148.9226 & -153.1720 - 79.9453 & -175.4806 & -18.1483 \end{bmatrix}$$

$$X^2 = \begin{bmatrix} 25.1305 & 9.1845 & -15.2403 & 18.0842 & -9.2918 & -0.1840 \end{bmatrix}$$

For the block diagonal matrix $X_1 = diag({}^1X^1, {}^1X^2, \ldots, {}^1X^n)$ with $n = 2$ and the sub-matrices ${}^1X^1, {}^1X^2$, we gave the following values for the centers of the fuzzy partitions:

$$X_1 = \begin{bmatrix} {}^1X^1 & 0 \\ 0 & {}^1X^2 \end{bmatrix}$$

where the centers of the fuzzy partitions of X^1, X^2 are the following:

$${}^1X^1 = \begin{bmatrix} 148 & 149 & 150 \end{bmatrix}$$

$${}^1X^2 = \begin{bmatrix} 42 & 43 & 44 \end{bmatrix}$$

In the sequel, we give the values of the matrices (W, W_1) which contain the neural weights

For the block diagonal (2*6x2) matrix W, the initial values are

$$W = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

And for the block diagonal (2*3x2) matrix W_1, the initial values

$$W = \begin{bmatrix} 0.04 & 0 \\ 0.04 & 0 \\ 0.04 & 0 \\ 0 & 0.04 \\ 0 & 0.04 \\ 0 & 0.04 \end{bmatrix}$$

In order to estimate our actual system-model in the proposed neural form, we need to calculate the function sigmoidals, i.e., the matrices $S(\chi), S_1(\chi)$. For that reason, we used the following values for their parameters a, b, c:

$a_1 = 0.1;\ \ b_1 = 1;\ \ c_1 = 0$ sigmoidal parameters used in w update.

$a_2 = 6;\ \ b_2 = 1;\ \ c_2 = 0$ sigmoidal parameters used in w_1 update.

Last, we need to find an appropriate matrix $A > 0$ which is chosen in order to satisfy the Lyapunov equation

$$PA + A^T P = -I,$$

and a matrix A we found in order to achieve the satisfaction of the above equation is

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

This matrix P, which is calculated by the Lyapunov equation is used in the update laws for $W, X, W1, X1$.

The figures 8.1,8.2,8.3 give the evolution of the states of the DC motor,i.e., the armature current I_a, the angular velocity Ω and the magnetic flux Φ respectively. We used the initial values $\Omega = 0.3$ (being in fact the 30 per cent of the nominal value of velocity's actual value) and $I_a = 0.3$ for the Angular Velocity and the Armature Current respectively.

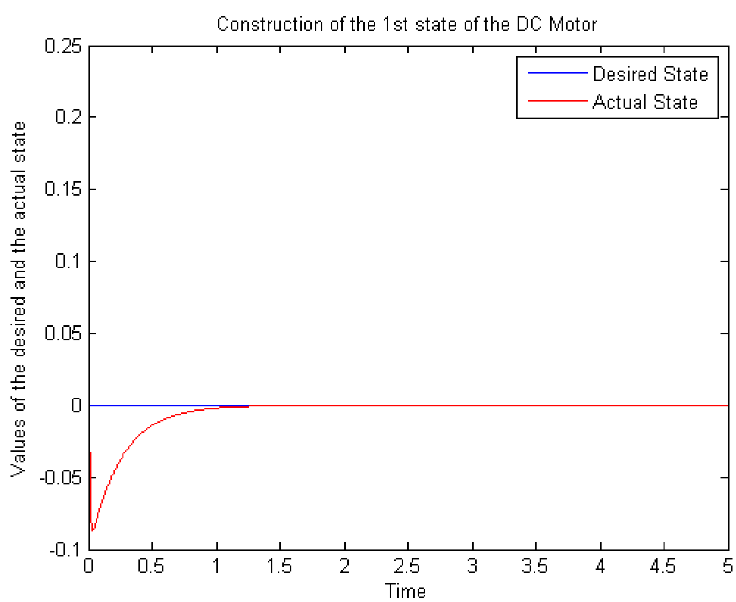

Figure 8.1: Evolution of the Armature Current of the DC Motor System

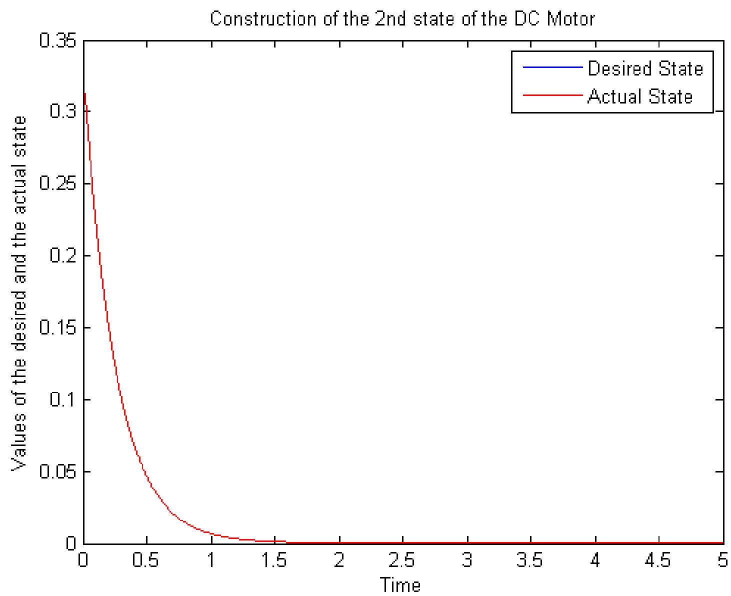

Figure 8.2: Evolution of the Angular Velocity of the DC Motor System

As can be seen,both I_a and Ω converge to zero very fast as desired.

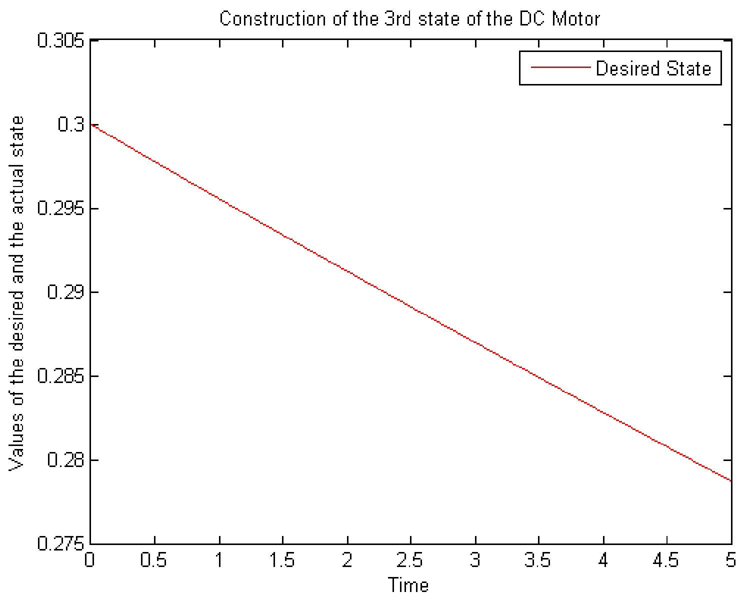

Figure 8.3: Evolution of the Magnetic Flux of the DC Motor System

As we can see above the magnetic flux remains bounded,as desired.

In the sequel, in Fig. 8.4 we give the evolution of our controller u, which is calculated in the controller stage.

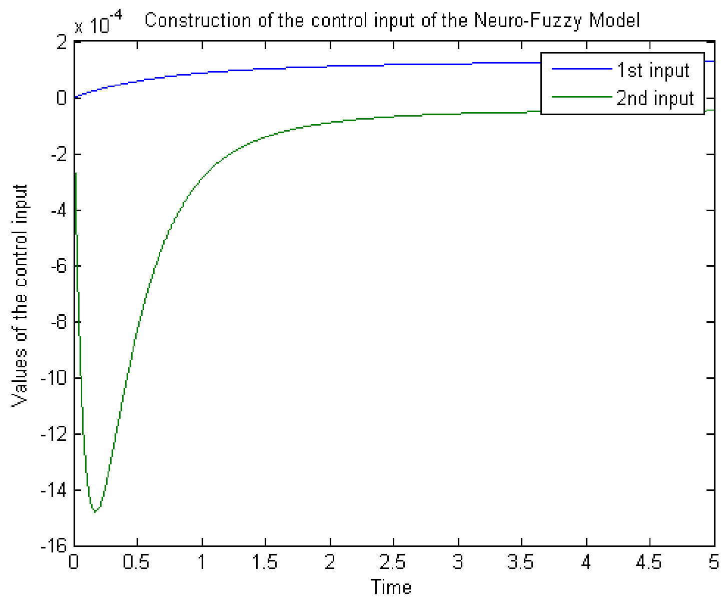

Figure 8.4: Evolution of the Controller of the proposed scheme

As can be seen above, the $2 - state$ control input remains bounded and converges to zero as time evolves.

Last, in Fig. 8.5 we give the evolution of the error between the F-RHONN approximator and the actual system.

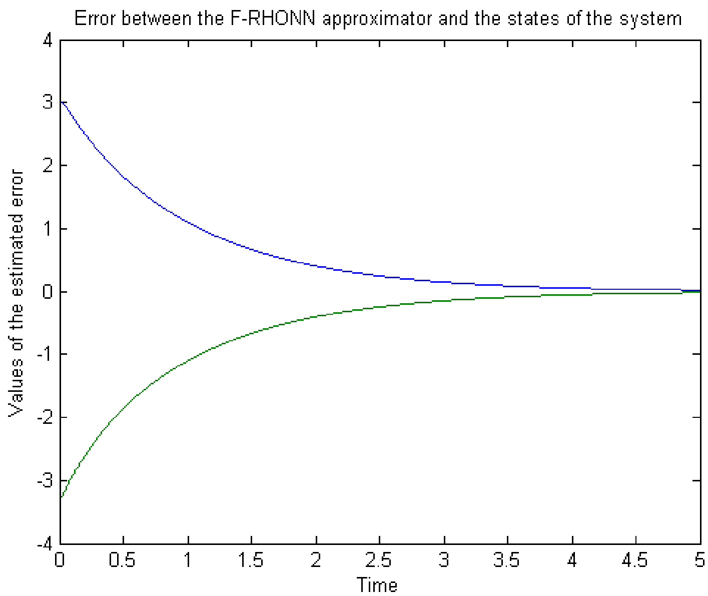

Figure 8.5: Evolution of the error signal between the F-HONNF approximator and the actual system .

As can be seen above, the error between the F-RHONN approximator and the actual system converges to zero as desired.

In the sequel we present the figures 8.6,8.7,8.8 for the same variables, but now we put initial values for $I_a, \Omega = 0.1$.

The three states of the DC motor system are the following

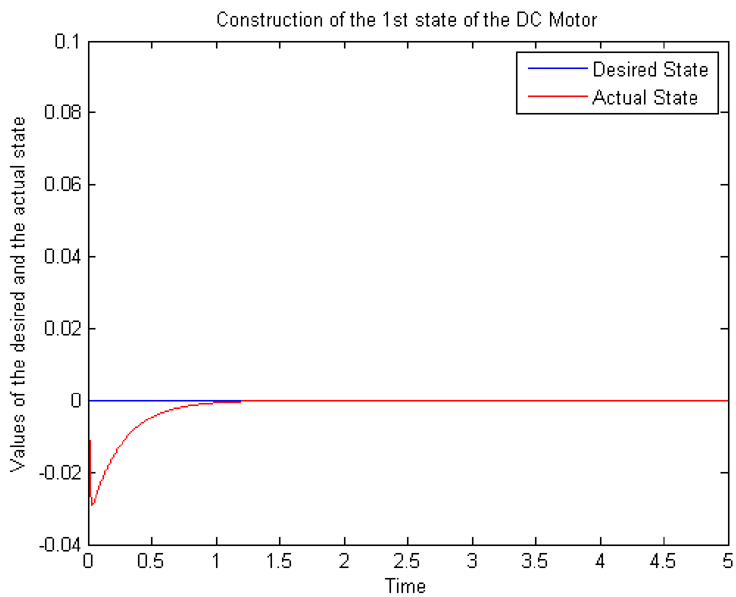

Figure 8.6: Evolution of the Armature Current of the DC Motor System

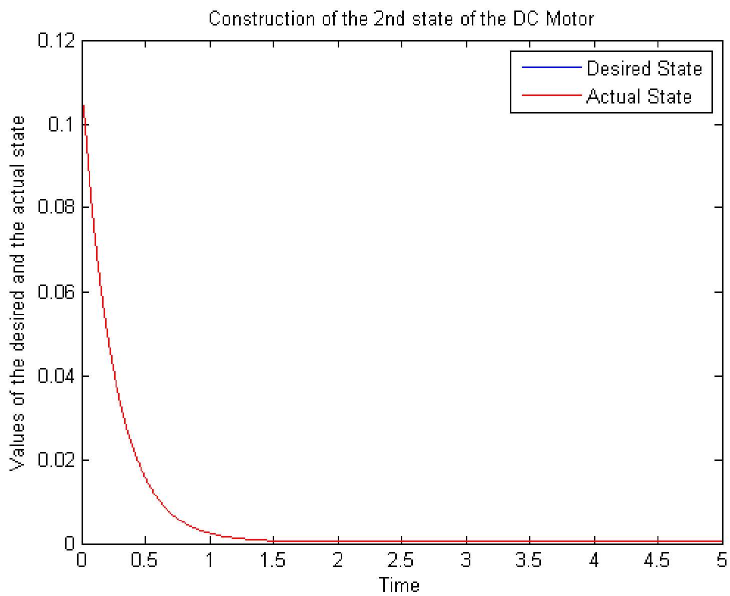

Figure 8.7: Evolution of the Angular Velocity of the DC Motor System

As can be seen,both Ω and I_a converge to zero very fast as desired despite the change of the initial values.

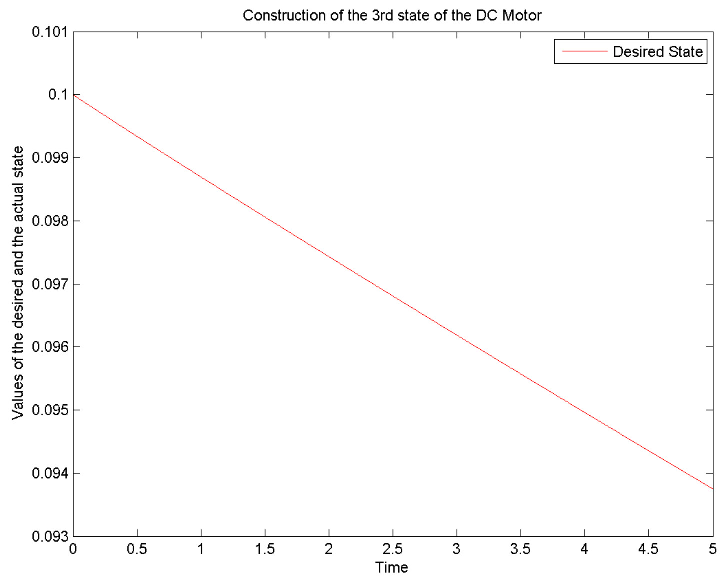

Figure 8.8: Evolution of the Magnetic Flux of the DC Motor System

As we can see above the magnetic flux remains bounded as well,as desired.

In the sequel, we give the evolution of the control signal u, which is calculated by the controller stage.

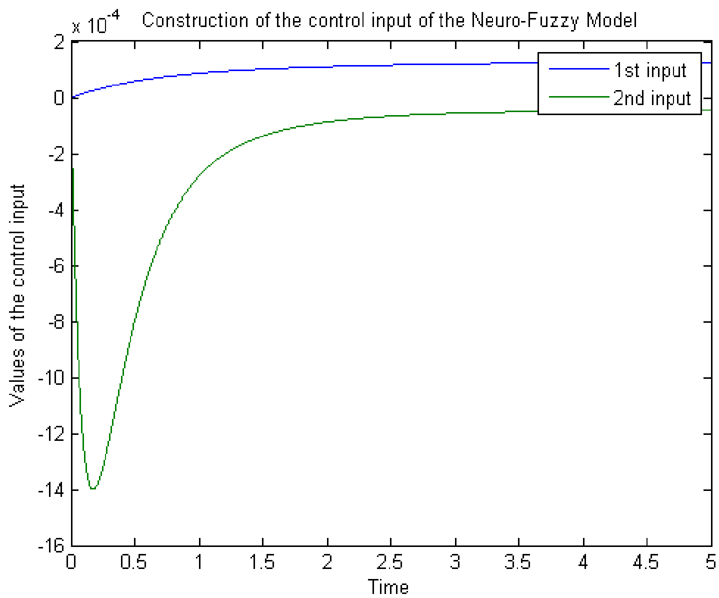

Figure 8.9: Evolution of the Controller of the proposed scheme

As can be seen above, the $2 - state$ control input remains bounded and converges to zero as time evolves.

Last, we give the evolution of the error between the F-RHONN approximator and the actual system.

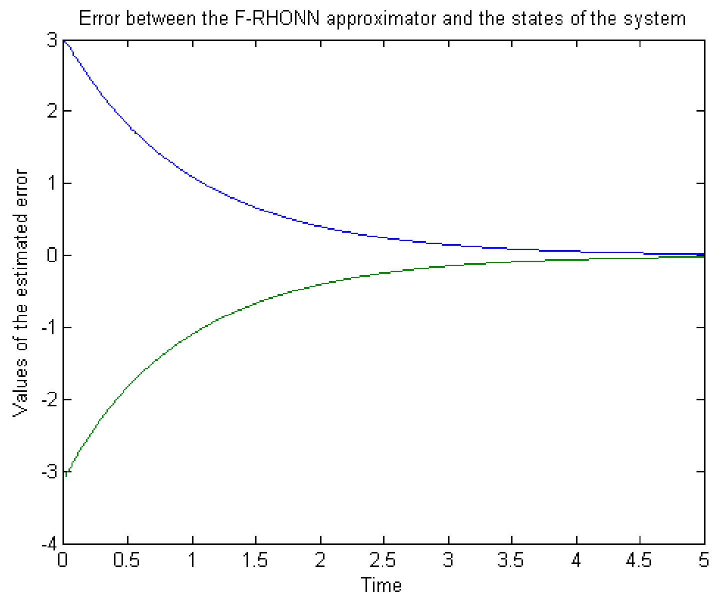

Figure 8.10: Evolution of the error signal between the F-HONNF approximator and the actual system .

As can be seen above, the error between the F-RHONN approximator and the actual system converges to zero as desired.

More examples of the application of bilinear parameter hopping in contrast with the simple case of not applying this optimization technique are given in the presentation of the thesis as well.

We proved the stability of the proposed scheme for different values of the initial parameters which guarantees the viable results of the mentioned theory.

Chapter 9

Conclusions

9.1 Conclusions

In this master thesis we considered an indirect adaptive control scheme in order to regulate unknown nonlinear plants in bilinear form. This approach is based on a Neuro-Fuzzy Dynamical Systems definition, which uses the concept of Fuzzy Dynamical Systems (FDS) operating in conjunction with High Order Neural Network Functions (HONNFs). Since the plant is initially considered unknown, first it is proposed its approximation by a special form of an affine in the control fuzzy dynamical system (FDS) and in the sequel the fuzzy rules are approximated by appropriate HONNFs. Once the system is identified around an operation point is regulated to zero adaptively. The used scheme does not require a-priori experts' information on the number and type of input variable membership functions making it less vulnerable to initial design assumptions. Weight and Center updating laws for the involved HONNFs and the centers of the output membership functions respectively are provided, which guarantee that both the identification error and the system states reach zero exponentially fast, while keeping all signals in the closed loop bounded. A method of parameter hopping assures the existence of the control signal and is incorporated in the weight and center updating law. Simulations illustrate the potency of the method by comparing its performance with this of conventional approaches. More specifically, the applicability of the method was tested on a DC Motor system where it is shown that by following the proposed procedure one can obtain asymptotic regulation.

We presented some basic research works that have already been done in the field of Adaptive Neuro-Fuzzy Control in Chapter 3. The literature in that field is very rich and quite challenging for future work through the development of more sophisticated control schemes. In Chapter 4 are presented some preliminaries related to Recurrent High Order Neural Networks and the model that was used for the development of the neuro-fuzzy model F-RHONN for the estimation of our actual system. The underlying neuro-fuzzy model which we used in the development of the bilinear scheme is presented thoroughly in Chapters 5,6, by introducing some basic facts of fuzzy systems description models and the way in which weighted indicator functions are used in order to approximate those fuzzy models. Chapter 7 introduces the indirect adaptive control of unknown nonlinear systems in a special bilinear form. This extension is based on the underlying neuro-fuzzy model of F-RHONNs. This scheme needs less information of the actual operation of the system (linguistic information related to the functionality of the examined system), and only an initial estimate of the centers of the output membership functions, in a way that our algorithm can start and estimate those centers automatically through the adaptive procedure.

One more quite reasonable assumption that we use in order to derive suitable update laws for the estimated parameters involved is that we know the sign of the centers of the fuzzy partitions.

The application of the neuro-fuzzy model in the approach of controlling unknown dynamical systems in bilinear form still requires an initial estimate of the centers of the fuzzy output variables, which in the sequel are estimated automatically. The complete absence of knowledge of operational information (such that specific information mentioned) of the examined system leads to a more complex approach that is still to be searched. The simulation results of the DC Motor are quite comforting for our control objectives and verify the theoretical solidification of the proposed updating laws.

There are many open topics in the field of neuro-fuzzy modeling where this approach could be used in order to overcome other difficulties. The bilinear model based on the underlying neuro-fuzzy model we used could be extended with modeling uncertainties, such as unmodeled dynamics, in the representation of the system and be controlled as such. Interest could be gained if one could face the problem of direct control of bilinear dynamical systems, or reduced model order problems. Those are some aspects of proposed future work in the estimation and control of bilinear dynamical systems and the neuro-fuzzy indirect or direct control of them.

Chapter 10

Appendix

10.1 Proofs of Theorems

In this section we present the proofs of the theorems and lemmas we used in the previous sections of this diploma thesis.

Theorem 8 *Consider the system:*

$$\dot{\chi} = F(\chi, u), \tag{10.1}$$

where $\chi \in R^n$ is the system state, $u \in R^n$ is the system input and $F : R^{n+m} \to R^n$ is a smooth vector field defined on a compact set $\mathrm{Y} \subset R^{n+m}$

and the model

$$\dot{x} = Ax + W^T z, \tag{10.2}$$

where $x = [x_1, x_2, \cdots, x_n]^T \in R^n, W = [w_1, w_2, \cdots, w_n]^T \in R^{Lxn}$ and $A = diag\,[-a_1, -a_2, \cdots, -a_n]$ is a nxn diagonal matrix. Since $[a_i > 0, i = 1, 2, \cdots, n]$, A is a stability matrix. Although it is not written explicitly, the vector z is a function of both the neural network state x and the external input u.

Suppose that the system (10.1) and the model (10.2) are initially at the same state $x(0) = \chi(0)$; then for any $\epsilon > 0$ and any finite $T > 0$, there exists an integer L and a matrix $W^* \in R^{Lxn}$ such that the state $x(t)$ of the RHONN model (10.2) with L high-order connections and weight values $W = W^*$ satisfies

$$\sup_{0 \le t \le T} |x(t) - \chi(t)| \le \epsilon.$$

Proof 1 *[73] From (10.2), the dynamic behavior of the RHONN model is described by*

$$\dot{x} = Ax + W^T z(x, u). \tag{10.3}$$

Adding and subtracting $A\chi$, (10.1) is rewritten as

$$\dot{\chi} = A\chi + G(\chi, u), \tag{10.4}$$

where $G(x, u) = F(x, u) - A\chi$. Since $x(0) = \chi(0)$, the state error $e = x - \chi$ satisfies the differential equation

$$\dot{e} = Ae + W^T z(x,u) - G(\chi,u), \;\; e(0) = 0. \tag{10.5}$$

By assumption, $(\chi(t), u(t)) \in \mathcal{Y}$ *for all* $t \in [0,T]$*, where* $\mathcal{Y}$ *is a compact subset of* R^{n+m}*. Let*

$$\mathcal{Y}_e = \left\{ (\chi,u) \in R^{n+m} : |(\chi,u) - (\chi_y, u_y)| \leq \epsilon, \;\; (\chi_y, u_y) \in \mathcal{Y} \right\}. \tag{10.6}$$

It can be seen easily that $\mathcal{Y}_e$ *is also a compact subset of* R^{n+m} *and* $\mathcal{Y} \subset \mathcal{Y}_e$*. In simple words* $\mathcal{Y}_e$ *is* ϵ *larger than* $\mathcal{Y}$*, where* ϵ *is the required degree of approximation. Since,* z *is a continuous function, it satisfies a Lipschitz condition in* $\mathcal{Y}_e$*, i.e., there exists a constant* l *such that for all* $(x_1,u), (x_2,u)$ *in*$\mathcal{Y}_e$

$$|z(x_1,u) - z(x_2,u)| \leq l|x_1 - x_2|. \tag{10.7}$$

In what follows, we show that the function $W^T z(x,u)$ *satisfies the conditions of the Stone-Weierstrass Theorem and can approximate any continuous function over a compact domain, therefore.*

From (4.2),(4.3) it is clear that $z(x,u)$ *is in the standard polynomial expansion with the exception that each component of the vector* x *is preprocessed by a sigmoid function* $s(\cdot)$*. As shown in [14], preprocessing of input via a continuous invertible function does not affect the ability of a network to approximate continuous functions; therefore, it can be shown readily that if* L *is sufficiently large, then there exist weight values* $W = W^*$ *such that* $W^*Tz(x,u)$ *can approximate* $G(x,u)$ *to any degree of accuracy, for all* (x,u) *in a compact domain.Hence, there exists* $W = W^*$ *such that*

$$\sup_{(\chi,u)\in\mathcal{Y}_e} |W^*Tz(\chi,u) - G(\chi,u)| \leq \delta, \tag{10.8}$$

where δ *is a constant to be designed in the sequel.*

The solution of (10.5) is

$$e(t) = \int_0^t e^{A(t-\tau)} \left[W^*Tz(x(\tau),u(\tau)) - G(\chi(\tau),u(\tau))\right] d\tau =$$

$$\int_0^t e^{A(t-\tau)} \left[W^*Tz(x(\tau),u(\tau)) - W^*Tz(\chi(\tau),u(\tau))\right] d\tau +$$

$$\int_0^t e^{A(t-\tau)} \left[W^*Tz(\chi(\tau),u(\tau)) - G(\chi(\tau),u(\tau))\right] d\tau.$$

Since A is a diagonal stability matrix, there exists a positive constant α such that $\|e^{At}\| \le e^{-\alpha t}$ for all $t \ge 0$. Also, let $L = l\|W^*\|$.Based on the aforementioned definitions of the constants α, L, ϵ, let δ in (10.8) be chosen as:

$$\delta = \frac{\epsilon\alpha}{2} e^{\frac{-L}{\alpha}} > 0. \tag{10.9}$$

First consider the case where $(x(t), u(t)) \in \mathcal{Y}_e$ for all $t \in [0, T]$. Starting from the equation of the solution of (10.5), taking the norms on both sides and using (10.7),(10.8) and (10.9), the following inequalities hold for all $t \in [0, T]$:

$$\begin{aligned}|e(t)| \le \int_0^t \|e^{A(t-\tau)}\| \left|W^{*T} z(x(\tau), u(\tau)) - W^{*T} z(\chi(\tau), u(\tau))\right| d\tau \\ + \int_0^t \|e^{A(t-\tau)}\| \left|W^{*T} z(\chi(\tau), u(\tau)) - G(\chi(\tau), u(\tau))\right| d\tau, \\ \le \int_0^t e^{-\alpha(t-\tau)} L|e(\tau)| d\tau + \int_0^t \delta e^{-\alpha(t-\tau)} d\tau, \\ \le \tfrac{\epsilon}{2} e^{-\frac{L}{\alpha}} + L \int_0^t e^{-\alpha(t-\tau)|e(\tau)|d\tau}.\end{aligned}$$

Using the Bellman-Gronwall Lemma [34], we obtain

$$|e(t)| \le \frac{\epsilon}{2} e^{-\frac{L}{\alpha}} + e^{L} \int_0^t e^{-\alpha(t-\tau)} d\tau \ \ \le \frac{\epsilon}{2}. \tag{10.10}$$

Now suppose (for the sake of contradiction), that (x, u) does not belong to $\mathcal{Y}_e$ for all $t \in [0, T]$; then, by continuity of $x(t)$, there exist a T^*, where $0 \le T^* \le T$, such that $(x(T^*), u(T^*)) \in \partial\mathcal{Y}_e$, where $\partial\mathcal{Y}_e$ denotes the boundary of $\mathcal{Y}_e$. If we carry out the same analysis for $t \in [0, T^*]$ we obtain that in this interval $|x(t) - \chi(t)| \le \frac{\epsilon}{2}$, which is clearly a contradiction. Hence, (10.10) holds for all $t \in [0, T]$.

Lemma 2 *The system described by*

$$\dot{\chi}_i = -a_i \chi_i + w_i^* z(\chi, u), \ \ \chi_i(0) = \chi_i^0. \tag{10.11}$$

can be expressed as

$$\dot{\zeta}_i = -a_i \zeta_i + z_i, \ \ \zeta_i(0) = 0, \tag{10.12}$$

$$\chi_i = w_i^{*T} \zeta_i + e^{-a_i t} \chi_i^0. \tag{10.13}$$

Proof 2 *From (10.12) we have*

$$\zeta_i = \int\limits_0^t e^{-a_i(t-\tau)} z(\chi(\tau), u(\tau)) d\tau$$

therefore,

$$w_i^{*T}\zeta_i + e^{-a_i t}\chi_i^0 = e^{-a_i t} + \int\limits_0^t e^{-a_i(t-\tau)} w_i^{*T} z(\chi(\tau), u(\tau)) d\tau. \tag{10.14}$$

Using (10.11), the right hand side of (10.14) is equal to $\chi(t)$ and this concludes this proof.

Theorem 9 *Consider the RHONN model*

$$x_i = w_i^T \zeta_i, \quad i = 1, 2, \cdots, n \tag{10.15}$$

whose parameters are adjusted according to

$$\dot{w}_i = -\Gamma_i \zeta_i e_i, \ \ i = 1, 2, \cdots, n. \tag{10.16}$$

Then for $i = 1, 2, \cdots, n$

a) $e_i, \phi_i \in L_\infty$ (e_i and ϕ are uniformly bounded)
b) $\lim_{t\to\infty} e_i(t) = 0$

Proof 3 *Consider the Lyapunov function candidate*

$$V = \frac{1}{2}\sum_{i=1}^n \left(\phi_i^T \Gamma_i^{-1} \phi_i + \int\limits_t^\infty \epsilon_i^e(\tau) d\tau \right). \tag{10.17}$$

Using (10.16) and $e_i = \phi_i^T \zeta_i - \epsilon_i$, where $\phi_i = w_i - w_i^*$ is the weight estimation error, the time derivative of V in (10.17) is expressed as

$$\dot{V} = \sum_{i=1}^n \left(-e_i \phi_i^T \zeta_i - \tfrac{1}{2}\epsilon_i^2\right) = \sum_{i=1}^n \left(-e_i(e_i + \epsilon_i) - \tfrac{1}{2}\epsilon_i^2\right)$$

$$= -\tfrac{1}{2}\sum_{i=1}^n \left(e_i^2 + (e_i + \epsilon_i)^2\right) \le 0.$$

Since $\dot{V} \leq 0$, we obtain that $\phi_i \in L_\infty$. Moreover, using $e_i = \phi_i^T \zeta_i - \epsilon_i$ and the boundedness of ζ_i, we have that e_i is also bounded. To show that $e_i(t)$ converges to zero, we first note that since V is a non-increasing function of time and also bounded from below, the $\lim_{t\to\infty} V(t) = V_\infty$ exists; therefore, by integrating both sides of the above expression of the derivative $\dot{V}$ from $t = 0 to \infty$, and taking bounds we obtain

$$\int_0^\infty \sum_{i=1}^n e_i^2(\tau) d\tau \leq 2(V(0) - V_\infty),$$

so for $i = 1, 2, \cdots, n$ $e_i(t)$ is square integrable. Furthermore, using $e_i = \phi_i^T \zeta_i - \epsilon_i$:

$$\dot{e}_i(t) = \dot{\phi}_i^T \zeta_i + \phi_i^T \dot{\zeta}_i - \dot{\epsilon}_i = -e_i \zeta_i^T \Gamma_i \zeta_i - a_i \phi_i^T \zeta_i + \phi_i^T z - \dot{\epsilon}_i$$

Since $e_i, \zeta_i, \phi_i, \dot{\epsilon}_i$ are all bounded, $\dot{e}_i \in L_\infty$. Hence, by applying Barbalat's Lemma [73] we obtain that $\lim_{t\to\infty} e_i(t) = 0$.

Theorem 10 *Consider the filtered error RHONN model given by*

$$\dot{x}_i = -a_i x_i + w_i^T z, \quad i = 1, 2, \cdots, n, \tag{10.18}$$

whose weights are adjusted according to

$$\dot{w}_i = -\Gamma_i z e_i \tag{10.19}$$

where the adaptive gain Γ_i is a positive definite LxL matrix, and w_i is the estimate of the unknown vector w_i^*. Then for $i = 1, 2, \cdots, n$

(a) $e_i, \phi_i \in L_\infty$
(b) $\lim_{t\to\infty} e_i(t) = 0$

Proof 4 *Consider the Lyapunov function candidate*

$$V = \frac{1}{2} \sum_{i=1}^n \left(e_i^2 + \phi_i^T \Gamma_i^{-1} \phi_i \right) \tag{10.20}$$

Then, using $\dot{e}_i = -a_i e_i + \phi_i^T z, \;\; i = 1, 2, \cdots, n, where \phi_i := w_i - w_i^*$, and (10.19), and the fact that $\dot{\phi}_i = \dot{w}_i$, the time derivative of V in (10.20) satisfies

$$\dot{V} = -\sum_{i=1}^n a_i e_i^2 \leq 0 \tag{10.21}$$

Since $\dot{V} \leq 0$, from (10.20) we obtain that $e_i, \phi_i \in L_\infty$ for $i = 1, 2, \cdots, n$. Using this result in $\dot{e}_i = -a_i e_i + \phi_i^T z,\ \ i = 1, 2, \cdots, n, where \phi_i := w_i - w_i^*$, we also have that $\dot{e}_i \in L_\infty$. Now by employing the same techniques as in proof of theorem *9 it can be shown readily that $e_i \in L_2$, i.e., $e_i(t)$ is square integrable; therefore, by applying Barbalat's Lemma we obtain that* $\lim_{t\to\infty} e_i(t) = 0$.

Theorem 11 *Consider the filtered error RHONN model given by*

$$\dot{x}_i = -a_i x_i + w_i^T z, \quad i = 1, 2, \cdots, n, \tag{10.22}$$

whose weights are adjusted according to

$$\dot{w}_i = \begin{Bmatrix} -\Gamma_i z e_i, & if|w_i| \leq M_i \\ -\Gamma_i z e_i - \sigma_i \Gamma_i w_i, & if|w_i| > M_i \end{Bmatrix} \tag{10.23}$$

Then for $i = 1, 2, \cdots, n$

(a) $e_i, \phi_i \in L_\infty$
(b)there exist constants λ, m such that

$$\int_0^t |e(\tau)^2| d\tau \leq \lambda + m \int_0^t |v(\tau)^2| d\tau$$

Proof 5 *Consider the Lyapunov function candidate*

$$V = \frac{1}{2} \sum_{i=1}^{n} \left(\phi_i^T \Gamma_i^{-1} \phi_i + \int_t^\infty \epsilon_i^e(t) dt \right). \tag{10.24}$$

Using $\dot{e}_i = -a_i e_i + \phi_i^T z - v_i$,and (10.23) it can be shown that

$$\dot{V} = \sum_{i=1}^{n} (-a_i e_i^2 - e_i v_i - I_{w_i}^* \sigma_i \phi^T w_i). \tag{10.25}$$

where $I_{w_i}^*$ is the indicator function defined as $I_{w_i}^* = 1$ if $|w_i| > M_i$ and $I_{w_i}^* = 0$ if $|w_i| \leq M_i$. Since $\phi_i = w_i - w_i^*$, we have that

$$\phi_i^T w_i = \tfrac{1}{2}\phi_i^T \phi_i + \tfrac{1}{2}(\phi_i^T \phi_i + 2\phi_i^T w_i^*) = \tfrac{1}{2}|\phi_i|^2 + \tfrac{1}{2}|w_i|^2 - \tfrac{1}{2}|w_i^*|^2.$$

Since, by definition, $|w_i^*| \leq M_i$ and $|w_i| > M_i$ for $I_{w_i}^* = 1$, we have that

$$I_{w_i}^* \tfrac{\sigma_i}{2}(|w_i|^2 - |w_i^*|^2) \geq 0;$$

therefore, (10.25) becomes

$$\dot{V} \leq \sum_{i=1}^{n}(-a_i e_i^2 - I_{w_i}^* \frac{\sigma_i}{2}|\phi_i|^2 - e_i v_i), \tag{10.26}$$

$$\leq \sum_{i=1}^{n}(-\tfrac{a_i}{2}e_i^2 - \tfrac{\sigma_i}{2}|\phi_i|^2) + \sum_{i=1}^{n}\Big((1-I_{w_i}^*)\tfrac{\sigma_i}{2}|\phi_i|^2 - \tfrac{a_i}{2}(e_i^2 + \tfrac{2}{a_i}e_i v_i)\Big).$$

So. we have that

$$\dot{V} \leq -\alpha V + \sum_{i=1}^{n}\left((1-I_{w_i}^*)\frac{\sigma_i}{2}|\phi_i|^2 + \frac{v_i^2}{2a_i}\right), \tag{10.27}$$

where

$$\alpha := \min\left\{a_i,\ \tfrac{\sigma_i}{\lambda_{\max}(\Gamma_i^{-1})};\ i = 1, 2, \cdots, n\right\}$$

and $\lambda_{\max}(\Gamma_i^{-1}) > 0$ *denotes the maximum eigenvalue of* Γ_i^{-1}*. Since*

$$(1-I_{w_i}^*)\tfrac{\sigma_i}{2}|\phi_i|^2 = \left\{\begin{array}{c} \tfrac{\sigma_i}{2}|\phi_i|^2 \quad if|w_i| \leq M_i \\ 0 \quad otherwise \end{array}\right\}$$

we obtain that $(1-I_{w_i}^*)\frac{\sigma_i}{2}|\phi_i|^2 \leq \sigma_i M_i^2$*. Hence, (10.27) can be written in the form*

$$\dot{V} \leq -\alpha V + K,$$

where $K := \sum_{i=1}^{n}(\sigma_i M_i^2 + \bar{v}_i^2/2a_i)$ *and* $\bar{v}_i$ *is an upper bound for* v_i*; therefore, for* $V \geq V_0 = K/\alpha$*, we have that* $\dot{V} \leq 0$*, which implies that* $V \in L_\infty$*. Hence* $e_i, \phi_i \in L_\infty$*.*

To prove the second part, we note that by completing the square in (10.26) we obtain

$$\dot{V} \leq \sum_{i=1}^{n}(-a_i e_i^2 - e_i v_i) \leq \sum_{i=1}^{n}\left(-\frac{a_i}{2}e_i^2 + \frac{v_i^2}{2a_i}\right). \tag{10.28}$$

Integrating both sides of (10.28) yields

$$V(t) - V(0) \leq \sum_{i=1}^{n}\left(-\tfrac{a_i}{2}\int_0^t e_i(\tau)^2 d\tau + \tfrac{1}{2a_i}\int_0^t v_i(\tau)^2 d\tau\right),$$

$$\leq -\frac{a_{min}}{2}\int_0^t |e(\tau)|^2 d\tau + \frac{1}{2a_{min}}\int_0^t |v(\tau)|^2 d\tau,$$

where $a_{min} := \min a_i$; $i = 1, 2, \cdots, n$; *therefore,*

$$\int_0^t |e(\tau)|^2 d\tau \leq \frac{2}{2a_{min}}[V(0) - V(t)] + \frac{1}{2a_{min}^2}\int_0^t |v(\tau)|^2 d\tau,$$

$$\leq \lambda + \mu\int_0^t |v(\tau)|^2 d\tau,$$

where $\lambda := (2/a_{min}) sup_{t\geq 0}[V(0) - V(t)]$ *and* $\mu := 1/a_{min}^2$. *This proves part (b) and concludes the proof of this theorem.*

Theorem 12 *Consider the identification scheme given by*

$$\dot{e} = Ae - X^*\tilde{W}S(\chi) - \tilde{X}WS(\chi) - X_1^*\tilde{W}_1S_1(\chi)u - \tilde{X}_1W_1S_1(\chi)u \tag{10.29}$$

Provided that $[X_1W_1S_1(\chi)]^{-1}$ *exists the learning laws:*

a) For the elements of W *and* X

$$\begin{cases} \dot{W} = sgn(X^*)^T PeS^T \\ \dot{X} = PeS^T W^T \end{cases} \tag{10.30}$$

b) For the elements of W_1 *and* X_1

$$\begin{cases} \dot{W}_1 = sgn(X_1^*)^T Peu^T S_1^T \\ \dot{X}_1 = Peu^T S_1^T W_1^T \end{cases} \tag{10.31}$$

guarantee the following properties.

- $e, \hat{\chi}, \tilde{W}, \tilde{W}_1, \tilde{X}, \tilde{X}_1 \in L_\infty, \quad e, \hat{\chi} \in L_2$
- $\lim_{t\to\infty} e(t) = 0, \quad \lim_{t\to\infty} \hat{\chi}(t) = 0$
- $\lim_{t\to\infty} \dot{\tilde{W}}(t) = 0, \quad \lim_{t\to\infty} \dot{\tilde{W}}_1(t) = 0$
- $\lim_{t\to\infty} \dot{\tilde{X}}(t) = 0, \quad \lim_{t\to\infty} \dot{\tilde{X}}_1(t) = 0$

where the matrices $sgn(X^*)$ *and* $sgn(X_1^*)$ *are defined in the proof.*

Proof 6 *Consider the Lyapunov function candidate*

$$V(e,\hat{\chi},\tilde{X},\tilde{W},\tilde{X}_1,\tilde{W}_1)=\tfrac{1}{2}e^TPe+\tfrac{1}{2}\hat{\chi}^TP\hat{\chi}+$$

$$+\tfrac{1}{2}tr\{\tilde{X}^T\tilde{X}\}+\tfrac{1}{2}tr\{\tilde{W}^T\Delta\tilde{W}\}$$

$$+\tfrac{1}{2}tr\{\tilde{X}_1^T\tilde{X}_1\}+\tfrac{1}{2}tr\{\tilde{W}_1^T\Delta_1\tilde{W}_1\}$$

Where $P>0$ is chosen to satisfy the Lyapunov equation

$$PA+A^TP=-I$$

and matrices Δ and Δ_1 are both diagonal $n\cdot m\times n\cdot m$ and defined as follows:

$$\Delta=diag\{(|\bar{f}_1^{1*}|,|\bar{f}_2^{1*}|,\ldots,|\bar{f}_m^{1*}|),(|\bar{f}_1^{2*}|,|\bar{f}_2^{2*}|,\ldots,|\bar{f}_m^{2*}|),\ldots,$$
$$(|\bar{f}_1^{m*}|,|\bar{f}_2^{m*}|,\ldots,|\bar{f}_m^{m*}|)\}$$

and

$$\Delta_1=diag\{(|\bar{g}_1^{1,1*}|,|\bar{g}_2^{1,1*}|,\ldots,|\bar{g}_m^{1,1*}|),$$
$$(|\bar{g}_1^{2,2*}|,|\bar{g}_2^{2,2*}|,\ldots,|\bar{g}_m^{2,2*}|),\ldots,$$
$$(|\bar{g}_1^{m,m*}|,|\bar{g}_2^{m,m*}|,\ldots,|\bar{g}_m^{m,m*}|)\}$$

Thus $\Delta\geq 0$ and $\Delta_1\geq 0$.

Taking the derivative of the Lyapunov function candidate and taking into account (7.12) we get

$$\dot{V}=\tfrac{1}{2}e^T\left(A^TP+PA\right)e+\tfrac{1}{2}\hat{\chi}^T\left(A^TP+PA\right)\hat{\chi}+$$

$$+\tfrac{1}{2}tr\{\dot{\tilde{X}}^T\tilde{X}\}+\tfrac{1}{2}tr\{\dot{\tilde{W}}^T\Delta\tilde{W}\}$$

$$+\tfrac{1}{2}tr\{\dot{\tilde{X}}_1^T\tilde{X}_1\}+\tfrac{1}{2}tr\{\dot{\tilde{W}}_1^T\Delta_1\tilde{W}_1\}\Rightarrow$$

$$\dot{V}=\tfrac{1}{2}e^T\left(A^TP+PA\right)e+\tfrac{1}{2}\hat{\chi}^T\left(A^TP+PA\right)\hat{\chi}+$$

$$\left(-\tfrac{1}{2}e^TP\tilde{X}WS-\tfrac{1}{2}e^TP\tilde{X}WS\right)-$$

$$\left(-\tfrac{1}{2}e^TPX^*\tilde{W}S-\tfrac{1}{2}e^TPX^*\tilde{W}S\right)-$$

$$-\left(\tfrac{1}{2}e^TP\tilde{X}_1W_1S_1u-\tfrac{1}{2}e^TP\tilde{X}_1W_1S_1u\right)-$$

$$-\left(\tfrac{1}{2}e^TPX_1^*\tilde{W}_1S_1u-\tfrac{1}{2}e^TPX_1^*\tilde{W}_1S_1u\right)+$$

$tr\{\dot{\tilde{W}}^T \Delta \tilde{W}\} + tr\{\dot{\tilde{W}}_1^T \Delta_1 \tilde{W}_1\} \Rightarrow$

$tr\{\dot{\tilde{X}}^T \tilde{X}\} + tr\{\dot{\tilde{X}}_1^T \tilde{X}_1\} \Rightarrow$

$\dot{V} = -\frac{1}{2}e^T e - \frac{1}{2}\hat{\chi}^T \hat{\chi} + e^T PX\tilde{W}S + e^T PX\tilde{W}_1 S_1 U + tr\{\dot{\tilde{W}}^T \tilde{W}\} + tr\{\dot{\tilde{W}}_1^T \tilde{W}_1\} \Rightarrow$

Take:

$$\begin{cases} tr\{\dot{\tilde{W}}^T \Delta \tilde{W}\} = e^T PX^* \tilde{W} S \\ tr\{\dot{\tilde{X}}^T \tilde{X}\} = e^T P\tilde{X} W S \\ tr\{\dot{\tilde{W}}_1^T \Delta_1 \tilde{W}_1\} = e^T PX_1^* \tilde{W}_1 S_1 u \\ tr\{\dot{\tilde{X}}_1^T \tilde{X}_1\} = e^T P\tilde{X}_1 W_1 S_1 u \end{cases}$$

Then the Lyapunov function becomes:

$\dot{V} = -\frac{1}{2}e^T e - \frac{1}{2}\hat{\chi}^T \hat{\chi} \leq 0$

Using the fact that whenever $tr\{\dot{X}^T \tilde{X}\} = A\tilde{X}B$, *where* A *is a row and* B *is a column vector,* $\Rightarrow \dot{X} = A^T B^T$, *we get:*

$$\begin{cases} \Delta \dot{W} = X^{*T} PeS^T \\ \dot{X} = PeS^T W^T \\ \Delta_1 \dot{W}_1 = X_1^{*T} Peu^T S_1^T \\ \dot{X}_1 = Peu^T S_1^T W_1^T \end{cases} \tag{10.32}$$

We write $X^{*T} = \Delta\{sgn(X^*)\}^T$
and $X_1^{*T} = \Delta_1\{sgn(X_1^*)\}^T$

where:

$sgn(X^*) = diag\{sgn(X^{1*}), sgn(X^{2*}), \ldots, sgn(X^{n*})\}$

where:

$sgn(X^{i*}) = [sgn(\bar{f_1}^{i,*}), sgn(\bar{f_2}^{i,*}), \ldots, sgn(\bar{f_m}^{i,*})]$

and:

$sgn(X_1^*) = diag\{sgn(X_1^{1*}), sgn(X_1^{2*}), \ldots, sgn(X_1^{n*})\}$

where:

$$sgn(X_1^{i*}) = [sgn(\bar{g_1}^{i,i,*}), sgn(\bar{g_2}^{i,i,*}), \ldots, sgn(\bar{g_m}^{i,i,*})]$$

Then equations (10.32) become:

$$\begin{cases} \dot{W} = sgn(X^*)^T PeS^T \\ \dot{X} = PeS^T W^T \\ \dot{W}_1 = sgn(X_1^*)^T Peu^T S_1^T \\ \dot{X}_1 = Peu^T S_1^T W_1^T \end{cases} \tag{10.33}$$

The update laws (10.33) are implementable, provided we know the signs of the partitions, which is a very reasonable assumption. However the centers of the partitions are automatically selected by our algorithm optimally.

Using the above Lyapunov function candidate V and proving that $\dot{V} \leq 0$ all properties of the theorem are assured [53].

For the proof of the next theorem we shall use some basic equations of the section of Neuro-Fuzzy Indirect Adaptive Control, and those are the following

$$\dot{e} = Ae - X^*\tilde{W}S(\chi) - \tilde{X}WS(\chi) - X_1^*\tilde{W}_1S_1(\chi)u - \tilde{X}_1W_1S_1(\chi)u \tag{10.34}$$

$$u = -\left[X_1W_1S_1(\chi)\right]^{-1} XWS(\chi) \tag{10.35}$$

$$\dot{\hat{\chi}} = A\hat{\chi} \tag{10.36}$$

Theorem 13 *Consider the control scheme (10.34), (10.35), (10.36). The updating law:*

For the elements of ${}^1W^i$ given by the modified form:

$${}^1\dot{W}^i = sign({}^1X^{*i})^T p_ie_iu_is_i(\chi) \quad if \ \left|{}^1X^i \cdot {}^1W^i\right| > \theta_i > 0$$

$$or \quad \left|{}^1X^i \cdot {}^1W^i\right| = \theta_i \ and \quad {}^1X^i \cdot {}^1\dot{W}^i \leq 0$$

$${}^1\dot{W}^i = sign({}^1X^{*i})^T p_ie_iu_is_i(\chi) -$$

$$-\frac{2}{tr\{({}^1X^i)^T\ {}^1X^i\}}\ {}^1X^i\,{}^1W^i\ ({}^1X^i)^T \quad otherwise$$

guarantees the properties of theorem 12 and assures the existence of the control signal.

Proof 7 *In order the properties of theorem 12 to be valid it suffices to show that by using the modified updating law for ${}^1W^i$ the negativeness of the Lyapunov function is not compromised. Indeed the* ***if*** *part of the modified form of ${}^1\dot{W}^i$ is exactly the same with (10.31) and therefore according to theorem 12 the negativeness of V is in effect. The if part is used when the weights are at a certain distance (condition if $\left|{}^1X^i \cdot {}^1W^i\right| > \theta_i$)from the forbidden plane or at the safe limit (condition $\left|{}^1X^i \cdot {}^1W^i\right| = \theta_i$) but with the direction of updating moving the weights far from the forbidden plane (condition ${}^1X^i \cdot {}^1\dot{W}^i \leq 0$).*

In the ***otherwise*** *part of ${}^1\dot{W}^i$, term $-\frac{2}{tr\{({}^1X^i)^T\ {}^1X^i\}}\ {}^1X^i\,{}^1W^i\ ({}^1X^i)^T$ determines the magnitude of weight hopping, which as explained later and is depicted in Fig. 10.2 has to be two times the distance of the current weight vector to the forbidden hyper-plane. Therefore the* ***existence*** *of the control signal is assured because the weights never reach the forbidden plane. Regarding the* ***negativeness*** *of $\dot{V}$ we proceed as follows.*

Let that ${}^1W^{*i}$ contains the initial values of ${}^1W^i$ provided from the identification part such that $\left|{}^1X^i \cdot {}^1W^{*i}\right| >> \theta_i$ and that ${}^1\tilde{W}^i = {}^1W^i - {}^1W^{*i}$. Then, the weight hopping can be equivalently written with respect to ${}^1\tilde{W}^i$ as $-2\theta_i{}^1\tilde{W}^i/\|{}^1\tilde{W}^i\|$. Under this consideration the modified updating law is rewritten as ${}^1\dot{W}^i = sign({}^1X^{*i})^T p_i e_i u_i s_i(\chi) - 2\theta_i{}^1\tilde{W}^i/\|{}^1\tilde{W}^i\|$. With this updating law it can be easily verified that $\dot{V} = -\frac{1}{2}e^Te - \frac{1}{2}\hat{\chi}^T\hat{\chi} - \Theta$, with Θ being a positive constant expressed as $\Theta = \sum 2\theta_i\left(({}^1\tilde{W}^i)^T){}^1\tilde{W}^i)\right)/\|{}^1\tilde{W}^i\|$, where the summation includes all weight vectors which require hopping. Therefore, the negativeness of $\dot{V}$ is actually enhanced.

The figure below shows the procedure, where a simplified 2-dimensional representation is given.

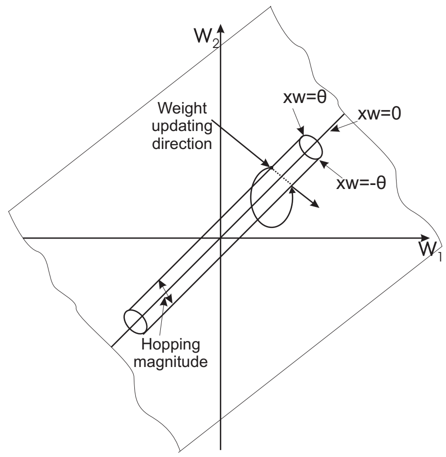

Figure 10.1: Pictorial Representation of Parameter Hopping [54].

Vectorial Proof of Parameter Hopping

In selecting the terms involved in parameter hopping we start from the vector definition of line, of a plane and the distance of a point to a plane. The equation of a line in vector form is given by

$$r = \alpha + \lambda t,$$

where α is the position vector of a given point of the line, t is a vector in the direction of the line and λ is a real scalar. By giving different numbers to λ we get different points of the line each one represented by the corresponding position vector r. The vector equation of a plane can be defined by using one point of the plane and a vector normal to it. In this case

$$r \cdot n = \alpha \cdot n = d,$$

is the equation of the plane, where α is the position vector of a given point on the plane, n is a vector normal to the plane and d is a scalar. When the plane passes through zero, then apparently $d = 0$. To determine the distance of a point B with position vector b from a given plane we consider Fig. and combine the above definitions as follows. Line BN is perpendicular to the plane and is described by vector equation

$$r = b + \lambda n,$$

where n is the normal to the plane vector. However, point N also lies on the plane and in case the plane passes through zero.

$$r \cdot n = 0 \Rightarrow (b + \lambda n) \cdot n = 0 \Rightarrow \lambda = \tfrac{-b \cdot n}{\|n\|}$$

Apparently, if one wants to get the position vector of B', the symmetrical of B in respect to the plane, this is given by

$$r = b - 2\tfrac{b \cdot n}{\|n\|} n$$

In our problem $b = W_1^i$, our plane is described by the equation $X_1^i \cdot W_1^i = 0$ and it has already been mentioned the normal to it is the vector X_1^i.

The figure below shows the vector explanation of what has already been mentioned.

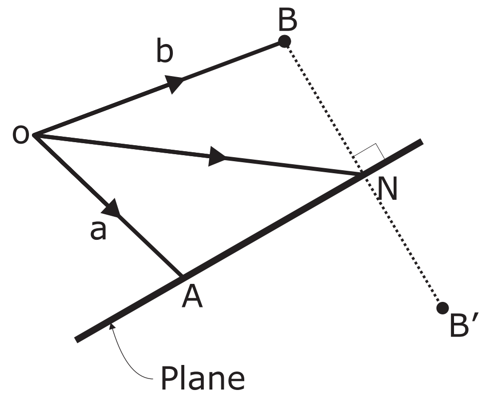

Figure 10.2: Vector Explanation of Parameter Hopping [54].

10.2 Optimization Techniques

An important part of every adaptive control scheme is the online estimator of the adaptive law used to provide an estimate of the plant or controller parameters at each time t. Most of these adaptive laws are derived by minimizing certain cost functions with respect to the estimated parameters. The type of the cost function and method of minimization determines the properties of the resulting adaptive law as well as the overall performance of the adaptive scheme.

In this subsection we introduce some notations as well as some simple optimization techniques that includes the method of *steepest descent*, referred to as *gradient method* and the *gradient projection method* for constrained minimization problems.

10.2.1 Notation and Mathematical Background

A real-valued function $f : R^n \to R$ is said to be continuously differentiable if the partial derivatives $\frac{\partial f(x)}{\partial x_1}, \cdots, \frac{\partial f(x)}{\partial x_n}$ exist for each $x \in R^n$ and are continuous functions of x. In this case we write $f \in \mathcal{C}^1$. More generally, we write $f \in \mathcal{C}^m$ if all partial derivatives of order m exist and are continuous functions of x. If $f \in \mathcal{C}^1$, the *gradient* of f at a point $x \in R^n$ is defined to be the column vector

$$\nabla f(x) \triangleq \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

If $f \in \mathcal{C}^2$, the *Hessian* of f at x is defined to be the symmetric $n \times n$ matrix having $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ as the $ijth$ element, i.e.,

$$\nabla^2 f(x) \triangleq \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]_{n \times n}$$

A subset $\mathcal{S}$ of R^n is said to be *convex* if for every $x, y \in \mathcal{S}$ and $\alpha \in [0,1]$ we have $\alpha x + (1-\alpha)y \in \mathcal{S}$. A function $f : \mathcal{S} \to R$ is said to be *convex over the convex set* $\mathcal{S}$ if for every $x, y \in \mathcal{S}$ and $\alpha \in [0,1]$ we have

$$f(\alpha x + (1-\alpha)y) \leq af(x) + (1-\alpha)f(y)$$

Let $f \in \mathcal{C}^1$ over an open convex set $\mathcal{S}$; then f is convex over $\mathcal{S}$ if and only if

$$f(y) \geq f(x) + (\nabla f(x))^T(y-x) \;\; \forall x, y \in \mathcal{S}$$

If $f \in \mathcal{C}^2$ over $\mathcal{S}$ and $\nabla^2 f(x) \geq 0 \ \ \forall x \in \mathcal{S}$, then f is convex over $\mathcal{S}$.

Let us now consider the unconstrained minimization problem

$$\begin{array}{c} \textit{minimize } J(\theta) \\ \textit{subject to } \theta \in R^n \end{array}$$

where $J : R^n \rightarrow R$ is a given function. We say that the vector θ^* is a global minimum for the above minimization problem if

$$J(\theta^*) \ \leq \ J(\theta) \ \ \forall \theta \ \in \ R^n.$$

A necessary and sufficient condition satisfied by the global minimum θ^* is given by the following lemma.

Lemma 3 *Assume that $J \in \mathrm{C}^1$ is convex over R^n. Then θ^* is a global minimum for the above minimization problem if and only if*

$$\nabla J(\theta^*) = 0$$

A vector $\bar{\theta}$ is called a regular point of the surface $\mathrm{S}_\theta = \{\theta \in R^n | g(\theta) = 0\}$ *if* $\nabla g(\bar{\theta}) \neq 0$*. At a regular point $\bar{\theta}$, the set*

$$\mathrm{M}(\bar{\theta}) = \{\, \theta \in R^n \mid \theta^T \nabla g(\bar{\theta}) = 0 \,\}$$

is called the tangent plane of g at $\bar{\theta}$.

10.2.2 The Method of Steepest Descent (Gradient Method)

This is one of the oldest and most widely known methods for solving the mentioned unconstrained minimization problem. It is also one of the simplest for which a satisfactory analysis exists. More sophisticated methods are often motivated by an attempt to modify the basic steepest descent technique for better convergence properties. The method of steepest descent proceeds from an initial approximation θ_0 for the minimum θ^* to successive points $\theta_1, \theta_2, \cdot, \cdot, \cdot, \in R^n$ in an iterative manner until some stopping condition is satisfied. Given the current point θ_k, the point θ_{k+1} is obtained by a linear search in the direction d_k, where

$$d_k \ = \ -\nabla J(\theta_k)$$

It can be shown that d_k is the direction from θ_k in which the initial rate of decrease of $J(\theta)$ is the greatest. Therefore, the sequence θ_k is defined by

$$\theta_{k+1} = \theta_k + \lambda_k d_k = \theta_k - \lambda_k \nabla J(\theta_k) \ \ (k = 0, 1, 2, \cdot, \cdot, \cdot), \tag{10.37}$$

where θ_0 is given and λ_k, known as the *step size*, or *step length*, is determined by the linear search method, so that θ_{k+1} minimizes $J(\theta)$ in the direction d_k from θ_k. A simpler expression for θ_{k+1} can be obtained by setting $\lambda_k = \lambda \ \forall k$, i.e.,

$$\theta_{k+1} = \theta_k - \lambda \nabla J(\theta_k). \tag{10.38}$$

In this case, the linear search for λ_k is not required, though the choice of the step length λ is a compromise between accuracy and efficiency. Considering infinitesimally small step lengths, (10.38) can be converted to the continuous-time differential equation

$$\dot{\theta} = -\nabla J(\theta(t)), \ \ \theta(t_0) = \theta_0, \tag{10.39}$$

whose solution $\theta(t)$ is the descent path in the time domain starting from $t = t_0$

The direction of steepest descent $d = -\nabla J$ can be scaled by a constant positive definite matrix $\Gamma = \Gamma^T$ as follows: We let $\Gamma = \Gamma_1 \Gamma_1^T$, where Γ_1 is an $n \times n$ nonsingular matrix, and consider the vector $\bar{\theta} \in R^n$ given by

$$\Gamma_1 \bar{\theta} = \theta.$$

Then the previous mentioned minimization problem is equivalent to

$$\textit{minimize } \ \bar{J}(\bar{\theta}) \triangleq J(\Gamma_1 \bar{\theta}) \ \textit{subject to } \ \bar{\theta} \in R^n.$$

If $\bar{\theta}^*$ is a minimum of $\bar{J}$, the vector $\theta^* = \Gamma_1 \bar{\theta}^*$ is a minimum of J. The steepest descent for the above minimization problem is given by

$$\bar{\theta}_{k+1} = \bar{\theta}_k - \lambda \nabla \bar{J}(\bar{\theta}_k). \tag{10.40}$$

Because $\nabla \bar{J}(\bar{\theta}) = \frac{\partial J(\Gamma_1 \bar{\theta})}{\partial \bar{\theta}} = \Gamma_1^T \nabla J(\theta)$ and $\Gamma_1 \bar{\theta} = \theta$, it follows from 10.40 that

$$\theta_{k+1} = \theta_k - \lambda \Gamma_1 \Gamma_1^T \nabla J(\theta_k).$$

Setting $\Gamma = \Gamma_1 \Gamma_1^T$, we obtain the scaled version for the steepest descent algorithm

$$\theta_{k+1} = \theta_k - \lambda \Gamma \nabla J(\theta_k). \tag{10.41}$$

The continuous-time version of 10.41 is given by

$$\dot{\theta} = -\Gamma \nabla J(\theta). \tag{10.42}$$

The convergence properties of 10.37,10.38,10.41 for different step lengths are given in any standard book on optimization.

10.2.3 Gradient Projection Method

Above, the search for the minimum of the function $J(\theta)$ given in our first minimization problem was carried out for all $\theta \in R^n$. In some cases, θ is constrained to belong to a certain convex set

$$\mathcal{S} \triangleq \{\theta \in R^n \mid g(\theta) \leq 0\} \tag{10.43}$$

in R^n, where $g(\cdot)$ is a scalar-valued function if there is only one constraint, and a vector-valued function if there is more than one constraint. In this case, the search for the minimum is restricted to the convex set defined by 10.43 instead of R^n.

Let us first consider the simple case where we have an equality constraint, i.e., the problem

$$\textit{minimize } J(\theta) \quad \textit{subject to } \; g(\theta) = 0, \tag{10.44}$$

where $g(\theta)$ is a scalar-valued function. One of the most common techniques for handling constraints is to use a descent method in which the direction of the descent is chosen to reduce the function $J(\theta)$ while remaining within the constrained region. Such a method is usually referred to as the *gradient projection method*.

We start with a point θ_0 satisfying the constraint, i.e., $g(\theta_0) = 0$. To obtain an improved vector θ_1, we project the negative gradient of J at θ_0, i.e., $-\nabla J(\theta_0)$, onto the tangent plane $M(\theta_0) = \{\theta \in R^n \mid \nabla g^T(\theta_0)\theta = 0\}$, obtaining the direction vector $Pr(\theta_0)$. Then θ_1 is taken as $\theta_0 + \lambda_0 Pr(\theta_0)$, where λ_0 is chosen to minimize $J(\theta_1)$. The general form of this iteration is given by

$$\theta_{k+1} \;=\; \theta_k \;+\; \lambda_k Pr(\theta_k), \tag{10.45}$$

where λ_k is chosen to minimize $J(\theta_k)$ and $Pr(\theta_k)$ is the new direction vector after projecting $-\nabla J(\theta_k)$ onto $M(\theta_k)$. The explicit expression for $Pr(\theta_k)$ can be obtained as follows: The vector $-\nabla J(\theta_k)$ can be expressed as a linear combination of the vector $Pr(\theta_k)$ and the normal vector $N(\theta_k) = \nabla g(\theta_k)$ to the tangent plane $M(\theta_k)$ at θ_k, i.e.,

$$-\nabla J(\theta_k) \;=\; \alpha \nabla g(\theta_k) \;+\; Pr(\theta_k) \tag{10.46}$$

for some constant α. Because $Pr(\theta_k)$ lies on the tangent plane $M(\theta_k)$, we also have $\nabla g^T(\theta_k) Pr(\theta_k) = 0$, which together with 10.46 implies that

$$-\nabla g^T \nabla J \;=\; \alpha \nabla g^T \nabla g,$$

i.e.,

$$\alpha \;=\; -(\nabla g^T \nabla g)^{-1} \nabla g^T \nabla J.$$

Hence, from 10.46 we obtain

$$Pr(\theta_k) \;=\; -[I - \nabla g (\nabla g^T \nabla g)^{-1} \nabla g^T] \nabla J. \tag{10.47}$$

We refer to $Pr(\theta_k)$ as the *projected direction onto the tangent plane* $M(\theta_k)$.

It is clear that when $g(\theta)$ is not a linear function of θ, the new vector θ_{k+1} given by 10.45 may not satisfy the constraint, so it must be modified. There are several successive approximation techniques that can be employed to move θ_{k+1} from $M(\theta_k)$ to the constraint surface $g(\theta) = 0$. One special case, which is often encountered in adaptive control applications, is when θ is constrained to stay inside a ball with a given center and radius, i.e., $g(\theta) \;=\; (\theta - \theta_0)^T(\theta - \theta_0) - M^2$, where θ_0 is a fixed constant vector and $M > 0$ is a scalar. In this case, the discrete projection algorithm which guarantees that $\theta_k \in \mathcal{S}\ \forall k$ is

$$\bar{\theta}_{k+1} \;=\; \theta_k \;+\; \lambda_k \nabla J,$$

$$\theta_{k+1} \;=\; \begin{cases} \bar{\theta}_{k+1} & if|\bar{\theta}_{k+1} - \theta_0| \le M, \\ \theta_0 \;+\; \frac{\bar{\theta}_{k+1} - \theta_0}{|\bar{\theta}_{k+1} - \theta_0|} & if|\bar{\theta}_{k+1} - \theta_0| > M \end{cases} \tag{10.48}$$

Letting the step length λ_k become infinitesimally small, we obtain the continuous-time version of 10.48, i.e.,

$$\dot{\theta} \;=\; Pr(\theta) \;=\; -[I \;-\; \nabla g(\nabla g^T \nabla g)^{-1} \nabla g^T] \nabla J. \tag{10.49}$$

Because of the sufficiently small step length, the trajectory $\theta(t)$, if it exists, will satisfy $g(\theta(t)) = 0 \;\forall t \geq 0$, provided that $\theta(0) = \theta_0$ satisfies $g(\theta_0) = 0$.

The scaled version of the gradient projection method can be obtained by using the change of coordinates $\Gamma_1 \bar{\theta} = \theta$, where Γ_1 is a nonsingular matrix that satisfies $\Gamma = \Gamma_1 \Gamma_1^T$ and Γ is the scaling positive definite constant matrix. The scaled version of 10.49 is given by:

$$\dot{\theta} \;=\; \bar{P}r(\theta),$$

where

$$\bar{P}r(\theta) \;=\; -[I \;-\; \Gamma \nabla g(\nabla g^T \Gamma \nabla g)^{-1} \nabla g^T] \Gamma \nabla J.$$

The minimization problem 10.44 can now be extended to:

$$minimize\ J(\theta) \quad subject\ to\ \ g(\theta) \leq 0, \tag{10.50}$$

where $\mathcal{S} = \theta \in R^n \,|\, g(\theta) \leq 0$ is a convex subset of R^n.

The solution to 10.50 follows directly from that of the unconstrained problem and 10.44. We start from an initial point $\theta_0 \in \mathcal{S}$. If the current point is in the interior of $\mathcal{S}$, defined as $\mathcal{S}_0 \triangleq \{\theta \in R^n \mid g(\theta) < 0$, then the unconstrained algorithm is used. If the current point is on the boundary of $\mathcal{S}$, defined as $\delta(\mathcal{S}) \triangleq \theta \in R^n \,|\, g(\theta) = 0$, and the direction of search given by the unconstrained algorithm is pointing away from $\mathcal{S}$, then we use the gradient projection algorithm. If the direction of search is pointing inside $\mathcal{S}$, then we keep the unconstrained algorithm. In view of the above, the solution of the constrained optimization problem 10.50 is given by:

$$\dot{\theta} \;=\; \begin{cases} -\Gamma \nabla J(\theta) \qquad if\ \theta \in \mathcal{S}_0 \\ \qquad or\, if\ \theta \in \delta(\mathcal{S})\, and\, -(\Gamma \nabla J)^T \nabla g \;\leq\; 0 \\ -\Gamma \nabla J(\theta) + \Gamma \frac{\nabla g \nabla g^T}{\nabla g^T \Gamma \nabla g} \Gamma \nabla J \quad otherwise, \end{cases} \tag{10.51}$$

where $\theta_0 \in \mathcal{S}$ and $\Gamma = \Gamma^T > 0$ is the scaling matrix.

Chapter 11

Bibliography-References

Bibliography

[1] K. Hornic, M. Stinchcombe, H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, Volume 2, pp. 359-366, 1989.

[2] J.S.R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system", *IEEE Trans. Syst. Man. Cyber.*, Volume 23, pp. 665-684, 1993.

[3] C.T. Lin, "A neural fuzzy control system with structure and parameter learning", *Fuzzy Sets and Systems*, Volume 70, pp. 183-212, 1995.

[4] K.B. Cho, B.H.Wang, "Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction", *Fuzzy Sets and Systems*, Volume 83, pp. 325-339, 1996.

[5] C.F.Juang, C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications", *IEEE Trans. Fuzzy Syst.*, Volume 6, pp. 12-32, 1998.

[6] R.P. Li, M. Mukaidono, "A new approach to rule learning based on fusion of fuzzy logic and neural networks", *IEICE Trans. Fuzzy Syst*, Volume E78-d, pp. 1509-1514, 1995.

[7] H.L. Hiew, C.P. Tsang, "Fuzzy Chaos and recursive partitioning", *B.Kosko(Ed.), Fuzzy Engineering, Prentice-Hall International Inc.* New Jersey, 1997.

[8] S.L. Chiu, "Fuzzy model identification based on cluster estimation", *Journal of Intelligent and Fuzzy Systems*, Volume 2, 1994.

[9] Y.H. Lin, G.A. Cunningham, "A new approach to fuzzy-neural system modelling", *IEEE Trans. Fuzzy Syst.*, Volume 3, pp. 190-197, 1995.

[10] C.F. Jang, C.T. Lin, "An online self-constructing neural fuzzy inference network and its applications", *IEEE Trans. Fuzzy Syst.*, Volume 6, pp. 12-32, 1998.

[11] S. Mitra, Y. Hayashi, "Neuro-fuzzy rule generation: survey in soft computing framework", *IEEE Trans. On Neural Networks*, Volume 11, pp. 748-768, 2000.

[12] B.S. Chen, C.H. Lee, Y.C. Chang, "Tracking design of uncertain nonlinear siso systems: Adaptive fuzzy approach", *IEEE Trans. Fuzzy Syst.*, Volume 4, pp. 32-43, 1996.

[13] J.T. Spooner, K.M. Passino, "Stable adaptive control using fuzzy systems and neural networks", *IEEE Trans. Fuzzy Syst.*, Volume 4, pp. 339-359, 1996.

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

[15] M.M. Polycarpou, M.J. Mears, "Stable adaptive tracking of uncertain systems using nonlinearly parameterized online approximators", *Int. J. Control*, Volume 70, no. 3, pp. 363-384, 1998.

[16] G.A. Rovithakis, M.A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks", *IEEE Trans. Syst, Man, Cybern.*, Volume 24, , pp. 400-412, 1994.

[17] F.C. Chen, C.C. Liu, "Adaptively controlling nonlinear continuous-time systems using multilayer neural networks", *IEEE Trans. Automat. Control*, Volume 39, pp. 1306-1310, 1994.

[18] N. Golea, A. Golea, K. Benmahammed, "Stable indirect fuzzy adaptive control", *Fuzzy Sets and Systems*, Volume 137, pp. 353-366, 2003.

[19] E.B. Kosmatopoulos, M.A. Christodoulou, "Recurrent neural networks for approximation of fuzzy dynamical systems", *International Journal of Intelligent Control and Systems*, Volume 1, pp. 223-233, 1996.

[20] M.A. Christodoulou, D.C. Theodoridis, Y.S. Boutalis, "Building Optimal Fuzzy Dynamical Systems Description Based on Recurrent Neural Network Approximation", *in Proc. Int. Conf. of Networked Distributed Systems for Intelligent Sensing and Control*, Kalamata, Greece, 2007.

[21] W. Yu, X. Li, "Fuzzy neural identification by online clustering with application on crude oil blending", *Int. Conf. on Fuzzy Systems*, Vancouver, BC, Canada, 2006.

[22] G. Cybenko, "Approximations by superpositions of a sigmoidal function", *Mathematics of Control Signals and Systems*, Volume 2, pp. 303-314, 1989.

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

[24] D. Rumelhart, D. Hinton, G. Williams, "Learning internal respectations by error propagation", *Parallel Distributed Processing*, D.Rumelhart and F.McClelland, Eds, Volume 1, Cambridge, MA, MIT Press, 1986.

[25] F.J. Pineda, "Generalization of backpropagation to recurrent networks", *Phys. Rev. Lett.*, Volume 59, no.19, pp. 2229-2232, 1988.

[26] K.S. Narendra, K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks", *IEEE Transactions on Neural Networks*, Volume 2, no.2, pp. 252-262, 1991.

[27] K.S. Narendra, K. Parthasarathy, "Identification and Control of Dynamical Systems using Neural Networks", *IEEE Transactions on Neural Networks*, Volume 1, no.1, pp. 4-27, 1990.

[28] P.J. Werbos, "Backpropagation through time: what it does and how we do it", *Proc. IEEE*, Volume 78, pp. 1550-1560, 1990.

[29] R.J. Williams, D. Zipser, "A learning algorithm for continually running fully recurrent neural networks", *Neural Computation*, Volume 1, pp. 270-280, 1989.

[30] M.M. Polycarpou, P.A. Ioannou, "Identification and Control of Nonlinear Systems using neural networks models: design and stability analysis", *Tech. Rep. 91-09-01, Univ. of Southern Cal., Los Angeles*, 1991.

[31] R.M. Sanner, J.-J.E. Slotine, "Gaussian neural networks for direct adaptive control", *IEEE Trans. Neural Networks*, Volume 3, no.6., pp. 837-863, 1992.

[32] F.-C. Chen, C.-C. Liu, "Adaptively controlling nonlinear continuous-time systems using multi-layer neural networks",*IEEE Trans. Automat. Control*, Volume 39, no.6, pp. 1306-1310, 1994.

[33] F.-C. Chen, H.-K. Khalil, "Adaptive control of a class of nonlinear discrete time systems using neural networks",*IEEE Trans. Automat. Control*, Volume 40, no.5, pp. 791-801, 1995.

[34] F.L. Lewis, K. Liu, A. Yesildirek, "Neural net robot controller with quaranteed tracking performance", *IEEE Trans. Neural Natworks*, Volume 6, no.3, pp. 703-715, 1995.

[35] N. Sadegh, "Nonlinear identification and control via neural networks", *Control Systems with Inexact Dynamic Models*, DSC-Volume 33, ASME Winter Annual Meeting, 1991.

[36] N. Sadegh, "A perceptron network for functional identification using radial gaussian networks", *IEEE Trans. Neural Networks*, Volume 4, no.6, pp. 982-988, 1991.

[37] G.A. Rovithakis, M.A. Christodoulou, "Adaptive Control of Unknown Plants using Dynamical Neural Networks", *IEEE Trans. Systems Man Cybernetics*, Volume 24, no.3, pp. 400-412, 1994.

[38] E.B. Kosmatopoulos, M. Polycarpou, M.A. Christodoulou, P.A. Ioannou, "High-order neural networks structures for identification of dynamical systems", *IEEE Trans. Neural Networks*, Volume 6, no.2, pp. 422-431, 1995.

[39] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc.Natl.Acad.Sci.*, Volume 81, pp. 3088-3092, 1984.

[40] M.A. Cohen, S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks", *IEEE Trans. on Systems, Man., and Cybernetics*, Volume SMC-13, pp. 815-826, 1983.

[41] P. Paretto, J.J. Niez, "Long term memory storage capacity of multiconnected neural networks", *Biol.Cybern.*, Volume 54, pp. 53-63, 1986.

[42] P. Baldi, "Neural networks, orientations of the hypercube and algebraic threshold functions", *IEEE Transactions on Information Theory*, Volume IT-34, pp. 523-530, 1988.

[43] A. Dempo, O. Faroimi, T. Kailath, "High-order absolutely stable neural networks", *IEEE Transactions on Ciruits and Systems*, Volume 38, no.1, 1991.

[44] Y. Kamp, M. Hasler, "Recursive Neural Networks for Associative Memory", *J.Wiley and Sons*, 1991.

[45] C.L. Giles, D. Chen, C.B. Miller, H.H. Chen, G.Z. Sun, Y.C. Lee, "Second-order recurrent neural networks for grammatical inference", *Proc. of Inter. Joint Conf. on Neural Networks*, IJCNN91, Volume 2, pp. 273-281, 1991.

[46] R. Liou, M.R. Azimi-Sadjadi, R. Dent, "Detection of dim targets in high cluttered backround using high order correlation neural networks", *Proc. of Iner. Joint Conf. on Neural Networks*, IJCNN91, Volume 1, pp. 701-706, 1991.

[47] N.E. Cotter, "The Stone-Weierstrass theorem and its application to neural network", *IEEE Trans. on Neural Networks*, Volume 1, no.4, pp. 290-295, 1990.

[48] G.C. Goodwin, K.S. Sin, "Adaptive filtering, Prediction and Control", *Prentice Hall*, New Jersey, 1984.

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

[50] P.A. Ioannou, A. Datta, "Robust Adaptive Control: a unified approach", *Proceedings of the IEEE*, 1991.

[51] K.S. Narendra, A.M. Annaswamy, "Stable Adaptive Systems", Englewood Cliffs, NJ, *Prentice-Hall*, 1989.

[52] M.M. Polycarpou, P.A. Ioannou, "On the existence and uniqueness of solutions in adaptive control systems", *IEEE Trans. on Automatic Control*, Volume 30, no.3, pp. 474-479, 1994.

[53] P.Ioannou, B.Fidan, "Adaptive Control Tutorial", *SIAM:Advances in Design and Control Series*, 2006.

[54] Y.Boutalis, D.Theodoridis, M.Christodoulou, "A new Neuro FDS definition for indirect adaptive control of unknown nonlinear systems using a method of Parameter Hopping", *IEEE Transactions on Neural Networks*, 20(4):609-625, 2009.

[55] D.Theodoridis, Y.Boutalis, M.Christodoulou, "Direct Adaptive Control of Unknown Nonlinear Systems Using a New Neuro-Fuzzy Method together with a Novel Approach of Parameter Hopping", *Kybernetica*, 45(3):349-386, 2009.

[56] D.Theodoridis, Y.Boutalis, M.Christodoulou, "Indirect Adaptive Control of Unknown multi variable nonlinear systems with parametric and dynamic uncertainties using a new neuro-fuzzy system description", *International Journal of Neural Systems(IJNS)*, Volume 20, no.2, pp. 129-148, 2010.

[57] D.Theodoridis, Y.Boutalis, M.Christodoulou, "A new Direct Adaptive Regulator with Robustness Analysis of Systems in Brunovsky Form", *International Journal of Neural Systems (IJNS)*, Volume 20, no.4, pp. 319-339, 2010.

[58] D.Theodoridis, Y.Boutalis, M.Christodoulou, "Direct Adaptive Regulator of Unknown Nonlinear Systems with Analysis of the Model Order Problem", *Journal of Zhejiang University* - Science C (Computers and Electronics), Volume 12, no.1, pp. 1-16, 2011.

[59] D.Theodoridis, Y.Boutalis, M.Christodoulou, "Dynamical Recurrent Neuro-Fuzzy Identification Schemes Employing Switching Parameter Hopping", *International Journal of Neural Systems*, Volume 22, no.2, pp. 16, 2012.

[60] D.Theodoridis, Y.Boutalis, M.Christodoulou, "Direct Adaptive Neuro-Fuzzy Trajectory Tracking of uncertain Nonlinear Systems", *International Journal of Adaptive Control and Signal Processing*, Special Issue: New Results on Neuro-Fuzzy Adaptive Control Systems, Volume 26, No.7, pp. 660-688, 2012.

[61] John A. Aseltine, "Transform method in Linear System Analysis", 1958.

[62] H. Adeli, H. Jiang, "Dynamic Fuzzy wavelet neural network model for structural system identification", *Journal of Structural Engineering*, 132(1),pp. 102-111, 2006.

[63] Ieroham S.Baruch, Rafael Beltran Lopez, Jose-Luis Olivares Guzman, Jose Martin Flores, "A fuzzy-neural multi-model for nonlinear systems identification and control", *Fuzzy Sets and Systems*, Volume 159, Issue 20, pp. 2650-2667, 2008.

[64] Mohamed Chemachema, Khaled Belarbi, "Robust Direct Adaptive Controller for a class of nonlinear systems basedon neural networks and fuzzy logic systems", *International Journal of Artificial Intelligence Tools*, Volume 16, Issue 03, 2007.

[65] Yixin Diao, K.M. Passino, "Adaptive neural/fuzzy control for inerpolated nonlinear systems", *IEEE Transactions on Fuzzy Systems*, Volume 10, Issue 5, pp. 583-595, 2002.

[66] Yang Gao, Meng Joo Er., "Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems", *IEEE Transactions on Fuzzy Systems*, Volume 11, Issue 4,pp. 462-477, 2003.

[67] Chia-Feng Juang, Chin-Teng Lin, "An online self-constructing neural fuzzy inference network and its applications", *IEEE Transactions on Fuzzy Systems*, Volume 6, Issue 1, pp. 12-32, 1998.

[68] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, Volume 8, Issu 3, pp. 338-353, 1965.

[69] S.-G. Kong, B. Kosko, "Adaptive Fuzzy Systems ofr backing up a truck-and-trailer", *IEEE Transactions on Neural Networks*, Volume 3, Issue 2,pp. 211-223, 1992.

[70] Chin-Teng Lin, C.S.G. Lee, "Reinforcement structure/parameter learning for neural network-based fuzzy logic control systems", *IEEE Transactiona on Neural Networks*, Volume 2 ,Issue 1,pp. 46-63, 1994.

[71] Chia-Feng Juang, "A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms", *IEEE Transactions on Neural Networks*, 2002.

[72] Chin-Hung Lee, Ching-Cheng Teng, "Identification and Control of dynamic systems using recurrent fuzzy neural networks", *IEEE Transactions on Fuzzy Systems*, Volume 8, Issue 4, pp. 349-366, 2000.

[73] George A. Rovithakis, Manolis A. Christodoulou, "Adaptive Control with Recurrent High-Order Neural Networks, Theory and In dustrial Applications", *Advances in Industrial Control*, Springer, 2000.

[74] P.A. Mastorocostas, J.B. Theocharis, "A recurrent fuzzy-neural model for dynamic system identification", *IEEE Transactions on Systems, Man and Cybernetics*, Volume 32, Issue 2, pp. 176-190, 2002.

[75] Jie Zhang, A.J. Morris, "Recurrent neuro-fuzzy networks for nonlinear process modeling", *IEEE Transactions on Neural Networks*, Volume 10, Issue 2, pp. 313-326, 1999.

[76] J.B. Theocharis, "A high-order recurrent neuro-fuzzy system with internal dynamics: Application to the adaptive noise cancellation", *Fuzzy Sets and Systems*, Volume 157, Issue 4,pp. 471-500, 2006.

[77] J.T. Spooner, K.M. Passino, "Stable adaptive control using fuzzy systems and neural networks", *IEEE Transactions on Fuzzy Systems*, Volume 4, Issue 2, pp. 339-359, 1996.

[78] Li-Xin Wang, "Adaptive Fuzzy Systems and Control - Design and Stability Analysis", *Englewood Cliffs, NJ:PTR Prentice Hall*, 1994.

[79] A.Yesildirek, F.L. Lewis, "Feedback Linearization using neural networks", *Automatica*, Volume 31, Issue 11,pp. 1659-1664, 1995.

[80] S.S. Sastry, A. Isidory, "Adaptive Control of Linearizable Systems", *IEEE Transactions on Automatic Control*, Volume 34, Issue 11,pp. 1123-1131, 1989.

[81] R.M. Sanner, J.-J.E. Slotine, "Gaussian Networks for Direct Adaptive Control", *IEEE Transactions on Neural Networks*, Volume 3,Issue 6,pp. 837-863, 1992.

[82] G.A. Rovithakis, C. Bechlioulis, "Adaptive Control with guaranteed transient and steady state tracking error bounds for strict feedback systems", *Automatica*, Volume 45, Issue 2,pp. 532-538, 2009.

[83] Jay A.Farrell, Marios M.Polycarpou, "Adaptive Approximation Based Control: Unifying Neural,Fuzzy and Traditional Adaptive Approximation Approaches", *Wiley-Interscience*, 2006.

[84] P.A. Ioannou, J. Sun, "Robust Adaptive Control", *PTR Prentice Hall*, 1995.

[85] I. Kanellakopoulos,P.V. Kokotovic, A.S. Morse, "systematic design of adaptive controllers for feedback linearizable systems", *IEEE Transactions on Automatic Control*, Volume 36, Issue 11,pp. 1241-1253, 1991.

[86] E.B. Kosmatopoulos, M.A. Christodoulou, P.A. Ioannou, "Dynamical Neural Networks that ensure Exponential Identification Error Convergence", *Neural Networks*, Volume 10, Issue 2, pp. 299-314, 1997.

[87] K.S. Narendra, L.Parthasarathy, "Identification and Control of Dynamical Systems using Neural Networks", *IEEE Transactions on Neural Networks*, Volume 1, Issue 1,pp. 4-27, 1990.

[88] K.S. Narendra, L.Parthasarathy, "Gradient Methods for the optimization of Dynamical Systems containing Neural Networks", *IEEE Transactions on Neural Networks*, Volume 2, Issue 2, pp. 252-262, 1991.

[89] M.M. Polycarpou, "Stable Adaptive Control Scheme for Nonlinear Systems", *IEEE Transactions on Automatic Control*, Volume 41, Issue 3, pp. 447-451, 1996.

[90] F.L. Lewis, A. Yesildirek, K.Liu, "Multilayer neural-net Robot Controller with guaranteed tracking performance", *IEEE Transactions on Neural Networks*, Volume 7, Issue 2, pp. 388-399, 1996.

[91] S.s. Ge, J. Zhang, "Neural-Network Control of Nonaffine Nonlinear System with zero dynamics by state and output feedback", *IEEE Transactions on Neural Networks*, Volume 14, Issue 4, pp. 900-918, 2003.

[92] A.J. Calise, R.T. Rysdyk, "Nonlinear Adaptive flight Control using Neural Networks", *IEEE Transactions on Control Systems*, Volume 18, Issue 6, pp. 14-25, 1998.

[93] K.B. Cho, B.H. Wang, "Radial Basis Function based Adaptive Fuzzy systems and their Applications to system identification and prediction", *Fuzzy Sets and Systems*, Volume 83, Issue 3, pp. 325-339, 1996.

[94] M. Hojati, S. Gazor, "Hybrid Adaptive Fuzzy Identification and Control of nonlinear Systems", *IEEE Transactions on Fuzzy Systems*, Volume 10, Issue 2, pp. 198-210, 2002.

[95] P.A. Phan, T. Gale, "Two-Mode Adaptive Fuzzy Control with Approximation Error Estimator", *IEEE Transactions on Fuzzy Systems*, Volume 15, Issue 5, pp. 943-955, 2007.

[96] P.A. Phan, T. Gale, "Direct Adaptive Fuzzy Control with a self-constructing Algorithm", *Fuzzy Sets and Systems*, Volume 159, Issue 8, pp. 871-899, 2008.

[97] H. Du, N. Zhang, "Application of evolving Takagi-Sugeno fuzzy model to nonlinear system identification", *Applied Soft Computing*, Volume 8, Issue 1, pp. 676-686, 2008.

[98] F. Abdollahi, H.A. Talebi, R.V. Patel, "A stable neural network-based observer with application to flexible-joint manipulators", *IEEE Transactions on Neural Networks*, Volume 17, Issue 1, pp. 118-129, 2006.

[99] E.B. Kosmatopoulos, M.A. Christodoulou, "The Boltzmann g-RHONN: A learning machine for estimating unknown probability distributions", *Neural Networks*, Volume 7, Issue 2, pp. 271-278, 1994.

[100] K.M. Passino, S. Yurkovic, "Fuzzy Control", *Addison-Wesley*, 1998.

[101] B. Kosko, H.M. Kim, "Neural Fuzzy motion estimation and compensation", *IEEE Transactions on Signal Processing*, Volume 45, Issue 10, pp. 2515-2532, 1997.

[102] H. Adeli, S.L. Hung, "An adaptive conjugate gradient learning algorithm for efficient training of neural networks", *Applied Mathematics and Computation*, Volume 62, Issue 1, pp. 81-102, 1194.

[103] E.B. Kosmatopoulos, M.M. Polycarpou, M.A. Christodoulou, "High-Order Neural Network structures for identification of dynamical systems", *IEEE Transactions on Neural Networks*, Volume 6, Issue 2, pp. 422-431, 1995.

[104] G.A. Rovithakis, I. Chalkiadakis, M.E. Zervakis, "High-Order neural network structure selection for function approximation applications using genetic algorithms", *IEEE Transactions on Systems, Man and Cybernetics*, Volume 34, Issue 1, pp. 150-158, 2004.

[105] M. Sugeno, T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling", *IEEE Transactions on Fuzzy Systems*, Volume 1, Issue 1, 1993.

[106] T. Takagi, M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control", *IEEE Transactions on Systems, Man and Cybernetics*, 1985.

[107] L.X. Wang, J.M. Mendel, "Fuzzy Basis Functions, Universal Approximation and Orthogonal Least-Squares Learning", *IEEE Transactions on Neural Systems*, Volume 3, Issue 5, pp. 807-814, 1992.

[108] S.S. Kim, K.C. Kwak, "Development of Quantum-based Adaptive Neuro-Fuzzy Networks", *IEEE Transactions on Systems, Man and Cybernetics*, Volume 40, Issue 1, pp. 91-100, 2010.

[109] D. Kukolj, E. Levi, "Identification of complex systems based on neural and Takagi-Sugeno fuzzy model", *IEEE Transactions on Systems, Man and Cybernetics*, Volume 34, Issue 1, pp. 272-282, 2004.

[110] R.P. Li, M. Mukaidono, "A new approach to rule learning Based on Fusion of Fuzzy Logic and Neural Networks", *IEEE Transactions on Information and Systems*, Vol. E78-D, No. 11, pp. 1509-1514, 1995.

[111] E.H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers", *International journal of Man-Machine Studies*, Volume 8, Issue 6, pp. 669-678, 1976.

[112] K.S. Narendra, L.S. Valavani, "Direct and Indirect Model Reference Adaptive Control", *Automatica*, Volume 15, Issue 6, pp. 653-664, 1979.

[113] K.J. Astrom, "Theory and Appclications of Adaptive Control - A survey", *Automatica*, Volume 19, Issue 5, pp. 471-486, 1983.

[114] P.V. Kokotovic, M. Arcak, "Constructive Nonlinear Control: A historical perspective", *Automatica*, Volume 37, Issue 5, pp. 637-662, 2001.

[115] R. Ortega, T. Yu, "Robustness of Adaptive Controllers - A survey", *Automatica*, Volume 25, Issue 5, pp. 651-677, 1989.

[116] M. Krstic, I. Kanellakopoulos, P.V. Kokotovic, "Nonlinear and Adaptive Control Design"m *Wiley*, 1995.

[117] H. Abid, M. Chtourou, A. Toumi, "Fuzzy Indirect Adaptive Control Scheme ofr Nonlinear Systems Based on Lyapunov Approach and Sliding Mode", *International Journal of Computational Cognition*, Volume 5, No. 1, 2007.

[118] C.-C. Kung, T.-H. Chen, "Observer-based Indirect Adaptive Fuzzy sliding mode Control with stable variable filters for unknown nonlinear dynamical systems", *Fuzzy Sets and Systems*, volume 155, Issue 2, pp. 292-308, 2005.

[119] H.N. Nounou, K.M. Passino, "Stable auto-tuning of adaptive fuzzy/neural controllers for nonlinear discrete-time systems", *IEEE Transactions on Fuzzy Systems*, Volume 12, Issue 1, pp. 70-83, 2004.

[120] R. Ordonez , K.M. Passino, "Stable multi-input multi-output adaptive fuzzy/neural Control", *IEEE Transactions on Fuzzy Systems*, Volume 7, Issue 3, pp. 345-353, 1989.

[121] L.A. Zadeh, "Probability Measures of Fuzzy Events", *Journal of Mathematical Analysis and Applications*, Volume 23, No. 2, 1968.

[122] L.A. Zadeh, "Similarity relations and Fuzzy ordering", *Information Sciences*, Volume 3, Issue 2, pp. 177-200, 1971.

[123] L.A. Zadeh, "Fuzzy Sets and their Applications to cognitive and decision processes", *Oxford, England: Academic Press*, 1975.

[124] M. Sugeno, "Industrial Application of Fuzzy Control", *Elsevier Science Inc.*, New York, NY, USA, 1985.

[125] L. Yu, Y.-Q. Zhang, "Evolutionary fuzzy neural networks for hybrid financial prediction", *IEEE Transactions on Systems, Man and Cybernetics*, Volume 35, Issue 2, pp. 244-249, 2005.

[126] G. Feng, G. Chen, "Adaptive Control of discrete-time Chaotic Systems: A fuzzy control approach", *Chaos, Solitions and Fractals*, Volume 23, Issue 2, pp. 459-467, 2005.

[127] S. Labiod, M.S. Boucherit, T.M. Guerra, "Adaptive Fuzzy Control of a class of MIMO nonlinear Systems", *Fuzzy Sets and Systems*, Volume 151, Issue 1, pp. 59-77, 2005.

[128] T. Chai, X. Liu, "New Approaches on H(infinity) Control of T-S fuzzy systems with interval time-varying delay", *Fuzzy Sets and Systems*, Volume 160, Issue 12, pp. 1669-1688, 2009.

[129] E.B. Kosmatopoulos, P.A. Ioannou, "A switching Adaptive Controller for feedback linearizable systems", *IEEE Transactions on Automatic Control*, Volume 44, Issue 4, pp.742-750, 1999.

[130] D.E. Miller, E.J. Davison, "An adaptive controller which provides an arbitrarily good transient and steady-state response", *IEEE Transactions on Automatic Control*, Volume 36, Issue 1, pp. 68-81, 1991.

[131] S.P. Moustakidis, G.A. Rovithakis, J.B. Theocharis, "An Adaptive Neuro-Fuzzy Tracking Control for multi-input nonlinear dynamic systems", *Automatica*, Volume 44, Issue 5, pp. 1418-1425, 2008.

[132] J.T. Spooner, K.M. Passino, "Decentralized Adaptive Control of nonlinear systems using radial basis neural networks", *IEEE Transactions on Automatic Control*, Volume 44, Issue 11, pp. 2050-2057, 1999.

[133] J.T. Spooner, K.M. Passino, R. Ordonez, "Indirect Adaptive Fuzzy Control for a class of discrete-time systems", *Proceedings of the American Control Conference*, Volume 5, pp. 3311-3315, 1997.

[134] J. Lam, S. Xu, "Exponential H(infinity) filter design for uncertain Takagi-Sugeno fuzzy systems with time delay", *Engineering applications of Artificial Intelligence*, Volume 17, Issue 6, pp. 645-659, 2004.

[135] F.L. Lewis, W.K. Tim, L.-Z. Wang, Z.X. Li, "Deadzone Compensation in motion control systems using adaptive fuzzy logic control", *IEEE Transactions on Control, Systems, Technology*, Volume 7, Issue 6, pp. 731-742, 1999.

[136] A.U. Levin, K.S. Narendra, "Control of nonlinear dynamical systems using neural networks. Observability, Identification and Control", *IEEE Transactions on Neural Networks*, Volume 7, Issue 1, pp. 30-42, 1996.

[137] J.J. Rubio, W. Yu, "Nonlinear System Identification with recurrent neural networks and dead-zone Kalman Filter algorithm", *Neurocomputing*, Volume 70, Issue 13-15, pp. 2460-2466, 2007.

[138] F.-M. Yu, H.-Y. Chung, S.-Y. Chen, "Fuzzy sliding mode controller design for uncertain time-delayed systems with nonlinear input", *Fuzzy Sets and Systems*, Volume 140, Issue 2, pp. 359- 374, 2003.

[139] E.N. Sanchez, M.A. Bernal, "Adaptive Recurrent neural control for nonlinear system tracking", *IEEE Transactions on Systems, Man and Cybernetics*, Volume 30, Issue 6, pp. 886-889, 2000.

I hereby affirm that I composed this work independently and used no other than the specified sources and tools and that I marked all quotes as such.