VARIABLE STRUCTURE SYSTEMS THEORY BASED TRAINING STRATEGIES FOR COMPUTATIONALLY INTELLIGENT SYSTEMS

by

Mehmet Önder Efe

BS in Electronics and Telecommunications Eng., Istanbul Technical University, 1993

MS in Systems and Control Eng., Boğaziçi University, 1996

Submitted to Institute for the Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor
of
Philosophy

Boğaziçi University

2000

ACKNOWLEGMENTS

It is a great pleasure and honor for me to carry out my Ph.D. Thesis under the supervision of Prof. Okyay Kaynak, who has always opened the doors in front of me and never got tired doing so for his students. He has not only been my supervisor but also a colleague and a friend. Many thanks to him for sharing his so much time with me.

Thanks to Assoc. Prof. Levent Akın, by whom I have first been introduced to the realm of computational intelligence.

Thanks to Assoc. Prof. Feza Kerestecioğlu for his kind and helpful attitudes and incredibly constructive suggestions in evaluating the progress reports of my thesis.

Thanks to Prof. Yorgo İstefanopulos for his careful evaluation of this study.

I am indebted to Assoc. Prof. Seta Boğosyan from Istanbul Technical University. She has kindly accepted being a member of the jury and carefully evaluated what I have done.

Many thanks to Prof. Xinghuo Yu from Central Queensland University, Australia. His invaluable suggestions and collaboration played a substantial role in refinement of this study.

Many thanks to my brothers M. Özgür Efe and İ. Erdem Efe, who have continuously helped me throughout my postgraduate studies.

I would like to express my gratitude to Dr. Kemalettin Erbatur, Dr. Melikşah Ertuğrul, Serdar İplikçi, A. Buğra Koku, M. Murat Dündar, Alparslan Parlakçı who are the dedicated researchers of Mechatronics Research and Applications Center. Thanks to my friends Özgür Aksoy, Uğur Yıldıran, Hasip Bulut, O. Hasan Dağcı and A. Murat Fışkıran.

Thanks to my parents, who have always motivated me and shared the good and sad moments of my life with patience...

ABSTRACT

VARIABLE STRUCTURE SYSTEMS THEORY BASED TRAINING STRATEGIES FOR COMPUTATIONALLY INTELLIGENT SYSTEMS

Noise rejection, handling the plant-model mismatches and alleviation of structured or unstructured uncertainties constitute prime challenges that are frequently encountered in the practice of systems and control engineering. One way of reducing the adverse effects of the stated difficulties and obtaining a good tracking precision is to utilize the techniques of variable structure systems theory, which offers well formulated solutions particularly to problems containing uncertainty and imprecision.

In this thesis, variable structure systems theory based training strategies of computationally intelligent systems are discussed. Two approaches are developed for alleviating the above mentioned difficulties. Additionally, the learning rate selection problem is treated from the point of variable structure control.

In the first approach described, a dynamic parameter adaptation law is derived and the applicability of the algorithm is discussed. The analysis presented aims to extract the conditions for establishing equivalence between sliding mode control of the plant and sliding mode learning in the controller. The second method is based on the selection of an extended Lyapunov function, by the use of which the sensitivity of the cost measure to the adjustable parameters are minimized together with the half squared error measure. Lastly the selection of the learning rate for three different gradient based parameter tuning strategies are discussed. The objective of the learning rate selection is to drive the plant to a sliding mode while the output of the controller is driven to a similar regime.

The performances of the methods developed are assessed on the dynamic model of a two degrees of freedom direct drive SCARA robotic manipulator, whose dynamic equations are assumed to be unknown throughout the results presented. In the simulations, the alleviation of the adverse effects of observation noise and varying payload conditions are studied.

ÖZET

İŞLEMSEL AKIL İÇEREN SİSTEMLER İÇİN DEĞİŞKEN YAPILI SİSTEMLER KURAMINA DAYALI EĞİTİM STRATEJİLERİ

Gürültü bağışıklığı, sistem-model uyumsuzluklarının aşılması ve yapısal olan ya da olmayan belirsizliklerin bertaraf edilmesi sistem ve kontrol mühendisliği uygulamalarında sıkça karşılaşılan önemli sorunlardır. Bahsedilen güçlüklerin olumsuz etkilerinin azaltılmasının ve iyi bir izleme hassasiyetinin elde edilmesinin bir yolu özellikle kesinlik ve hassasiyet içermeyen problemler için iyi yapılandırılmış çözümler öneren değişken yapılı sistemler kuramının tekniklerinin kullanımıdır.

Bu çalışmada işlemsel akıl içeren sistemlerin eğitiminde değişken yapılı sistemler kuramına dayalı stratejiler ele alınmaktadır. Belirtilen güçlüklerin aşılması için iki yaklaşım geliştirilmiş ve öğrenme katsayısı seçimi problemi kayma kipli denetim açısından incelenmiştir.

İlk yaklaşımda dinamik bir parametre uyarlama kuralı türetilmekte ve algoritmanın kontrol mühendisliği açısından uygulanabilirliği tartışılmaktadır. Yapılan analiz, sistemin kayma kipli denetimi ile denetleyici içerisinde kayma kipli öğrenme arasındaki denklik koşullarının çıkarsanmasını amaçlamaktadır. İkinci yöntem genişletilmiş bir Lyapunov fonksiyonunun seçimi ile oluşturulmaktadır. Bu yöntemde karesel hata ölçütüne ek olarak maliyet ölçütünün uyarlanabilir parametrelere duyarlılığı da en aza indirilmektedir. Son olarak türeve dayalı üç değişik parametre uyarlama kuralı için öğrenme katsayısının seçimi üzerinde durulmaktadır. Buradaki seçimin amacı sistemi bir kayma kipine sürerken denetleyici çıkışını da benzer bir rejime zorlamaktır.

Geliştirilen yöntemlerin başarımı, iki hareket serbestisine sahip, doğrudan sürümlü bir SCARA robotunun dinamik modeli üzerinde değerlendirilmektedir ve sistem denklemlerinin bilinmediği varsayılmaktadır. Benzetimlerde gözlem gürültüsünün ve değişken yük koşullarının olumsuz etkileri de irdelenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

a_{ij}	Slope parameter of the bell shaped membership function
B_{ϕ}	Bound of the norm of the adjustable parameter vector
$B_{\dot{u}_A}$	Bound of the norm of the time derivative of the augmented input vector
B_{u_A}	Bound of the norm of the augmented input vector
B_{τ}	Bound of the norm of the output signal of the controller
$B_{\tau d}$	Bound of the norm of the desired output signal of the controller
$B_{\dot{\tau}_d}$	A scalar number that bounds the norm of the time derivative of the desired output signal of the controller
B_1	A positive constant that bounds the first term of the time solution of the parameter vector
B_2	A positive constant that bounds the second term of the time solution of the parameter vector
b_{ij}	Flatness parameter of the bell shaped membership function
$\underline{C}(\underline{\theta},\underline{\dot{\theta}})$	Vector of Coriolis terms
c_{ij}	Center of the membership function
D	Input gain matrix of the plant
$\underline{e}$	The state tracking error vector
$\underline{e}_c$	The difference between the desired output and the produced output of the computationally intelligent system
$\underline{e}_{\tau}$	Equivalent torque error
f^i	Linear defuzzification function for i^{th} rule in the rule base
$\underline{f}_c$	Coulomb friction terms
$\underline{f}_p$	Vector of state functions
$\underline{f}_{p_i}$	State function of subsystem i
G	Sliding surface slope matrix
h	Number of Gaussian radial basis function neurons in the hidden layer of a GRBFNN structure
h_k	Number of hidden neurons contained in the k^{th} layer of a FNN structure
I	Identity matrix

J	Cost function
K	Uncertainty bound
$M(\underline{\theta})$	State varying inertia matrix
M_p	Load mass
m	Number of inputs of the computationally intelligent system
n	Number of outputs of the computationally intelligent system
n_b	Noise bound constant
P	Total number of training pairs contained in the training data set
p_1,p_2,p_3	Constants of the manipulator dynamics
R	Number of rules contained in the rule base of a fuzzy inference system
r_i	Order of the subsystem i
$\underline{S}_a$	Vector of approximated sign functions
$\underline{s}_A$	Augmented switching manifold
s_c	Time varying sliding surface variable for controller
s_p	Time varying sliding surface variable for plant
T_1	Numerator of the weighted average in the input-output relation of standard fuzzy systems
T_2	Denominator of the weighted average the in input-output relation of standard fuzzy systems
$T_{\underline{\phi}}$	Threshold function
t	Time variable
t_h	Hitting time
U_j^i	Fuzzy set corresponding to the domain of i^{th} rule's j^{th} linguistic label
$\underline{u}$	Input vector of the computationally intelligent system
$\underline{u}_A$	Vector of augmented inputs
V_c	Lyapunov function defined by s_c
V_p	Lyapunov function defined by s_p
$\underline{v}_k$	Bias vector of the neurons belonging to k^{th} layer of a FNN structure
$\underline{w}$	Vector of firing strengths
$\underline{w}_n$	Vector of normalized firing strengths
Y	Matrix including the first order Sugeno model defuzzifier parameters
$\underline{y}$	Weight vector

α_{ij}	Variance parameter of the Gaussian radial basis function
δ	Parameter of the approximated sign function
Φ	State transition matrix
$\underline{\phi}$	Generic form of the adjustable parameter vector of a computationally intelligent system
Γ_k	Weight matrix between the $(k\text{-}1)^{th}$ and k^{th} layers of a FNN structure
κ	A strictly positive constant for ensuring Lyapunov stability
Λ	Learning rate quantity
λ_i	Parameter of the sliding surface of the i^{th} subsystem
μ_{ij}	Membership function for the i^{th} rule's j^{th} input variable
μ	A positive constant for two-term Lyapunov function based parameter tuning approach
$\underline{N}_\phi$	Vector of backpropagated error signal
$\tilde{\mathrm{N}}$	Normalization operator
$\underline{\theta}$	State vector of the plant under control
$\underline{\hat{\theta}}$	State vector observed at the output of the identifier
$\underline{\theta_d}$	Desired state vector of the plant under control
ρ	A positive constant for two-term Lyapunov function based parameter tuning approach
σ	Integration variable
$\underline{\tau}$	Output vector of the computationally intelligent system
$\underline{\tau_c}$	Corrective control component of the control vector
$\underline{\tau_d}$	Desired output vector of the computationally intelligent system
$\underline{\tau_{eq}}$	Equivalent control component of the control vector
$\underline{\Omega}$	Vector of nonlinear activation functions in the layers of a FNN structure
ξ	A positive definite diagonal matrix of dimensions nxn
Ψ	The mapping between the sliding surface of the plant (s_p) and the zero learning-error level of the controller (s_c)

ADALINE	Adaptive linear elements
ANFIS	Adaptive neuro-fuzzy inference systems
BPE	Bounded parameter evolution
CC	Computational complexity

EBP	Error backpropagation
FL	Fuzzy logic
FNN	Feedforward neural network
GRBFNN	Gaussian radial basis function neural network
HF	High frequency
LR	Learning rate
LM	Levenberg-Marquardt
RMP	Reaching mode performance
RNN	Recurrent neural network
SFS	Standard fuzzy systems
SMC	Sliding mode control
SMP	Sliding mode performance
VSC	Variable structure control
VSS	Variable structure systems

1. INTRODUCTION

Intelligence in the form of well-organized solutions to ill-posed problems has been the primary focus of much of the recent engineering research and the field of computational intelligence has thus emerged. Artificial neural networks and fuzzy inference systems constitute the core approaches of computational intelligence, whose architectures have extensively been used in the applications ranging from image/pattern recognition to identification and control of nonlinear systems [1-5]. What lies behind this wide application spectrum is the fact that architectures in the field of computational intelligence have the capability of perceiving the operating environment and tolerating the faults mostly stemming from ambiguities in the model of the problem in hand. The verbal power of artificial learning and the numeric power of high-speed connectionist approaches encourage the use of neuro-fuzzy techniques in solving computationally complex problems. Since the structures considered are universal approximators, an architecture with an appropriate learning strategy can realize any mapping with a predefined realization error bound [6-8]. The most substantial problem in the use of computationally intelligent architectures is the safety in training. When the extensive use of artificial neural networks and fuzzy inference systems in the domain of systems and control engineering are taken into consideration, the importance of training safety becomes more apparent. Safety in this context is intimately related to the time behavior of the adjustable parameter vector. If a constant solution for parameter vector exists, a convergent behavior is sought iteratively; however, the problem in hand may impose a dynamic solution entailing a time-varying behavior in the parameter vector. In the case of the latter, a major objective has to be the maintenance of the boundedness of the parameter vector. The achievement of either the parametric convergence or the bounded parameter evolution is dependent on the ability of compensating the adverse effects of the disturbances. This could be provided by an appropriate design of the training strategy.

In the theory of control engineering, one way of designing a robust and stable control system is to use the Variable Structure Systems (VSS) approach, which enables the designer to come up with a rigorous stability analysis. It is a well-known fact that a variable structure controller with a switching output will (under certain circumstances)

result in a sliding mode on a predefined subspace of the state space. This mode has useful invariance properties in the face of uncertainties in the plant model and therefore is a candidate for the tracking control of uncertain nonlinear systems. The theory is well developed, especially for single-input systems in controller canonical form. The philosophy of the control strategy is simple, being based on two goals. First, the system is forced towards a desired dynamics; second, the system is maintained on that differential geometry. In the literature, the behavior observed until the system enters into the predefined subspace of the state space is named the reaching mode, while the behavior on this subspace is called the sliding mode [9]. The control strategy borrows its name from the latter dynamic behavior, and is called Sliding Mode Control (SMC).

Earliest notion of SMC strategy was constructed on a second order system in the late 1960s by Emelyanov [10]. The work stipulated that a special line could be defined on the phase plane, such that any initial state vector can be driven towards the line and then be maintained on it, while forcing the error dynamics towards the origin. Since then, the theory has greatly been improved and the sliding line has taken the form of a multidimensional surface, called the sliding surface, around which a switching control action takes place.

In Variable Structure Control (VSC), the existence of observation noise constitutes a prime difficulty. This is due to the fact that pure sliding control requires very fast switching on the input, which cannot be provided by real actuators. In other words, the input depends on the sign of a measured variable, which is very close to zero. This makes the control signal extremely vulnerable to measurement noise and leads to unnecessarily large control signals. To alleviate these difficulties, several modifications to the original sliding mode control law have been proposed in the literature, some recent ones of which are based on the use of fuzzy logic [11-12] and artificial neural networks [13-14]. These methodologies provide an extensive freedom for control engineers to broaden their understanding of the problem, to deal with problems of uncertainty and imprecision.

During the last two decades, numerous contributions to VSS theory have been made. Some of them are as follows. Hung, et al. [9] has reviewed the control strategy for linear and nonlinear systems. In [9], the switching schemes, putting the differential equations into

canonical forms and generating simple SMC strategies are considered in detail. In [15] and [16], applications of SMC scheme to robotic manipulators are studied and the quality of the scheme is discussed from the point of robustness. One of the crucial points in SMC is the selection of the parameters of the sliding surface. Some studies devoted to the adaptive design of sliding surfaces have shown that the performance of control system can be refined by interfacing it with an adaptation mechanism, which regularly redesigns the sliding surface [16-17]. This eventually results in a robust control system. The performance of SMC scheme is proven to be satisfactory in the face of external disturbances and uncertainties in the system model representation. Another systematic examination of SMC approach is presented in [18]. In this reference, the practical aspects of SMC design are assessed for both continuous-time and discrete-time cases and a special consideration is given to the finite switching frequency, limited bandwidth actuators and parasitic dynamics. In [19-21], the design of discrete-time SMC is presented with particular emphasis on the system model uncertainties.

Studies demonstrating the high performance of the SMC scheme in handling the uncertainties and imprecision have motivated the use of SMC scheme in the training of computationally intelligent systems. The results presented in [22-25] have shown that the convergence properties of the gradient-based training strategies can be improved by utilizing the SMC scheme. The approach presented in these references is based on the construction of a dynamic model for the learning strategy and the design of a sliding motion in the parametric displacement space. Since the dynamics of a gradient-based strategy is kept under control, the method presented in [22-25] can be considered as an indirect use of VSS theory. However, some studies on the direct use of SMC strategy are reported in the literature [26-27]. The methods discussed in these references deal with the dynamic adaptation of the parameters of a flexible model such that the error on the output of the model tends to zero in finite time [26-27]. The first results discussed by Sira-Ramirez, et al. [26] are on the inverse dynamics identification of a Kapitsa pendulum by assuming the bounds of uncertainties constant. Yu, et al. [27] extend the results of [26] by introducing adaptive uncertainty bound dynamics, and their work focuses on the same example as the application. The major drawback in both of the approaches is the fact that the dynamic adaptation mechanism needs the error on the output of the model. If the model is to be used as a controller, this fact constitutes a difficulty because the use of the

approaches proposed in [26-27] for control applications requires the error on the applied control signal, which is unavailable. In [28], a dynamic learning scheme based on SMC is discussed for the training of Feedforward Neural Networks (FNN). The method has been used for the identification of a periodic time signal. The potential difficulty in implementing the algorithm is the fact that the time derivatives of the hidden neuron outputs are differentiated for evaluating the value of the switching function.

This thesis is organized as follows. The second section introduces the commonly used architectures of computational intelligence and their relevance to mechatronics. Adaptive Linear Elements (ADALINE), FNN, Gaussian Radial Basis Function Neural Networks (GRBFNN), Standard Fuzzy Systems (SFS) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) are considered as the intelligent structures.

The third section is devoted to the background knowledge on VSS theory for SMC. Standard procedure of designing a controller that drives the tracking error vector to a sliding surface is presented and the extent of the relevant a priori knowledge in the design is emphasized.

In the fourth section, a dynamic parameter tuning strategy is derived from the negative definiteness requirement of a single-term Lyapunov function. An analysis of the equivalence between the sliding mode control of the plant and the sliding mode learning inside the controller is presented. Three conditions for the establishment of the equivalence are elaborated. This is followed by the simulation results. Lastly in this section, a discussion is presented concerning the obtained results.

The derivation of a dynamic adaptation strategy from a two-term Lyapunov function is presented in the fifth section. The first term of the Lyapunov function is as that adopted in the fourth section. The second term in the Lyapunov function reflects the magnitude of the sensitivity of the cost measure with respect to the adjustable parameters. The simulation results and a discussion are presented for the performance evaluation of the method.

The sixth section is devoted to the learning rate selection problem in gradient-based training strategies. The formulation presented is based on an ADALINE controller and aims to evaluate a sequence of learning rate variable so that the plant under control enters into a sliding regime. The conventional constraints on the learning rate are thus removed. A discussion on the applicability of the algorithm, computational requirements and the assessment of the results obtained are presented at the end of the section.

Conclusions constitute the seventh section of the thesis.

2. COMPUTATIONAL INTELLIGENCE IN MECHATRONICS

The primary focus of the technologic investments in the first half of the 20th century was the development of accurate mechanical components. In the second half, the emergence of high-speed microprocessors enabled the technologists to integrate the mechanical systems with the software supported electronic hardware, the design and implementation of which necessitate the knowledge of mechanical, electronics and computer engineering backgrounds. The fusion of these disciplines was later on called *mechatronics* displaying a larger operational spectrum than the overall concern of its constituents when considered individually.

A number of definitions for mechatronics have been proposed in the literature, differing in the particular characteristics that the definition is intended to emphasize. The most commonly used one emphasizes the synergy:

> "Mechatronics is the synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacturing of products and processes. The embedded intelligence may vary from programmed behavior to self-organization and learning" [29].

It is rational to explain that the multifunctionality and versatility of the mechatronic products are fed by the increasing standards of the daily life. Similarly, the pursuit of increased productivity and improved product quality encourages the utilization of mechatronic devices in industry. Therefore there is a great need for methodologies that enable the designer to handle the complexity of real-world systems, the complexity of which limits a detailed understanding of the overall structure, and leads to a design based on imperfect models. Particularly, in the field of systems and control engineering, the design of controllers for sophisticated systems is a challenge because of the essentiality of a priori knowledge, which can be in the form of mathematical equations, observed data or physical laws. The chasm between the theory and practice is felt strongly in the cases where the mathematical equations describing the system dynamics are intractable or

incomplete, or the observed data are corrupted. The difficulties stated are the typical problems frequently encountered in the design and implementation of control schemes.

Although a considerable amount of progress has been made in the field of conventional systems theory since 1950s and the resulting outcomes have been used in many industrial applications, there still exists many difficulties. Depending on the requirements of the design problem, the conventional framework offers various approaches, most of which assume the availability of the governing equations of the system under investigation [30-31]. This is usually not the case in practice. Even when the form of the system dynamics is known, the uncertainties are assumed to satisfy certain conditions, which imply the availability of a nominal dynamics [32], or the availability of the statistical properties of the disturbances [33-34]. The existence of uncertainty and impreciseness has therefore made it obligatory in many cases to integrate the automatic control routines with some manual actions, which exploit the knowledge of the human expert.

The driving force for devising the strategies that replace the human expert with autonomous machines has led to the emergence of computationally intelligent systems, the use of which enables the representation and implementation of decision making procedures of the human brain in a systematic manner. More explicitly, the known microscopic models of the process under investigation are taught to an intelligent system, which gradually refreshes its content to reach a sufficiently detailed representation and keeps maintaining the satisfaction of the objectives of the design by adding the extracted information to the a priori knowledge.

The breakthrough in 1960s by Prof. Zadeh [35], who is the founder of Fuzzy Logic (FL), stimulated the community to seek for unconventional methodologies that are not subject to the constraints of the conventional approaches. At about the same time, Widrow and Hoff [36] demonstrated the use of Adaptive Linear Elements (ADALINE) in control. These two incidents were the stimulants of a new trend in control engineering and have led to a radical departure from the conventional control techniques. Therefore, the idea of coping with the uncertainty and impreciseness resulted in the birth of the field of

computational intelligence, the models of which have the structural flexibility, e.g. artificial neural networks or fuzzy inference systems.

Artificial neural networks are well known with their property of representing complex nonlinear mappings. Earlier works on the mapping properties of these architectures have shown that neural networks are universal approximators [6-8]. The mathematical power of intelligence is commonly attributed to the neural systems because of their massively interconnected, fault tolerant architecture. Various architectures of neural systems are studied in the literature. Feedforward and Recurrent Neural Networks (FNN, RNN), Gaussian Radial Basis Function Neural Networks (GRBFNN) [1,3], dynamic neural networks [37], and Runge-Kutta neural networks [38-40] constitute typical structurally different models.

Fuzzy inference systems are the most popular constituent of the soft computing area since they are able to represent human expertise in the form of *IF antecedent THEN consequent* statements. In this domain, the system behavior is modeled through the use of linguistic descriptions. Although the earliest work by Prof. Zadeh on fuzzy systems [35] has not been paid as much attention as it deserved in the early 1960s, since then the methodology has become a well-developed framework. The typical architectures of fuzzy inference systems are those introduced by Wang [4-5], Takagi and Sugeno [41] and Jang, Sun and Mizutani [3]. In [4], a fuzzy system having Gaussian membership functions, product inference rule and weighted average defuzzifier is constructed and has become the standard method in most applications. Takagi and Sugeno change the defuzzification procedure where dynamic systems are used in the defuzzification stage. The potential advantage of the method is that under certain constraints, the stability of the system can be studied. Jang, et al. [3] propose an adaptive neuro-fuzzy inference system, in which a polynomial is used as the defuzzifier. This structure is commonly referred to as ANFIS in the related literature. The choice concerning the order of the polynomial and the variables to be used in the defuzzifier are left to the designer.

In this section, the architectures of artificial neural networks and fuzzy inference systems studied in the thesis are introduced and the data flow through the architecture is discussed for each structure. At the end of the section, a discussion on the concept of

learning in computational intelligence and connections to control engineering applications are established.

2.1. Adaptive Linear Elements

Adaptive linear elements are frequently used components of computationally intelligent systems. The structure of an ADALINE is illustrated in Figure 2.1 for an (m+1)-input case.

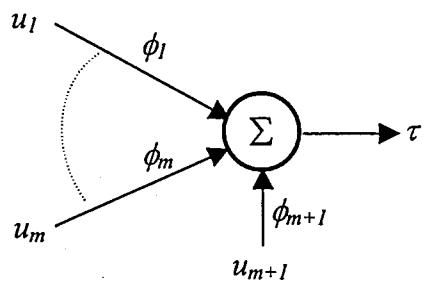

Figure 2.1. Structure of an Adaptive Linear Element

The parameter vector and the input vector of the ADALINE structure are defined as

$$\underline{\phi} = [\phi_1 \quad \phi_2 \quad \cdots \quad \phi_{m+1}]^T \tag{2.1}$$

$$\underline{u}_A = [u_1 \quad u_2 \quad \cdots \quad u_{m+1}]^T \tag{2.2}$$

The input-output relation of the structure can be defined as

$$\tau = \underline{\phi}^T \underline{u}_A \tag{2.3}$$

Without loss of generality, the u_{m+1} could be chosen as unity, which acts as a bias value, and $\underline{u}_A$ can be termed as the augmented input vector.

In the applications of computationally intelligent systems, ADALINE architecture is used either in the output layer of neural networks or in the defuzzification stage of fuzzy

inference systems. For this reason, it is important to keep the definitions in (2.1)-(2.2) and the relation in (2.3) in mind.

2.2. Feedforward Neural Networks

Feedforward neural networks constitute a class of neural network structures in which the data flow is from input to the output and no feedback connections are allowed. Because of the structural diversity of neural models, this discussion is devoted to the architecture and the mathematical representation of FNN structure, which is discussed from the point of control engineering.

The architecture of a FNN utilized in this study is illustrated in Figure 2.2, in which the neural network has three layers implying the sufficiency for realizing any continuous mapping to a desired degree of accuracy as long as the hidden layer contains sufficiently many neurons [6-8]. The number of neurons in the hidden layer is a design variable and is mostly determined either by trial and error or by empirical results.

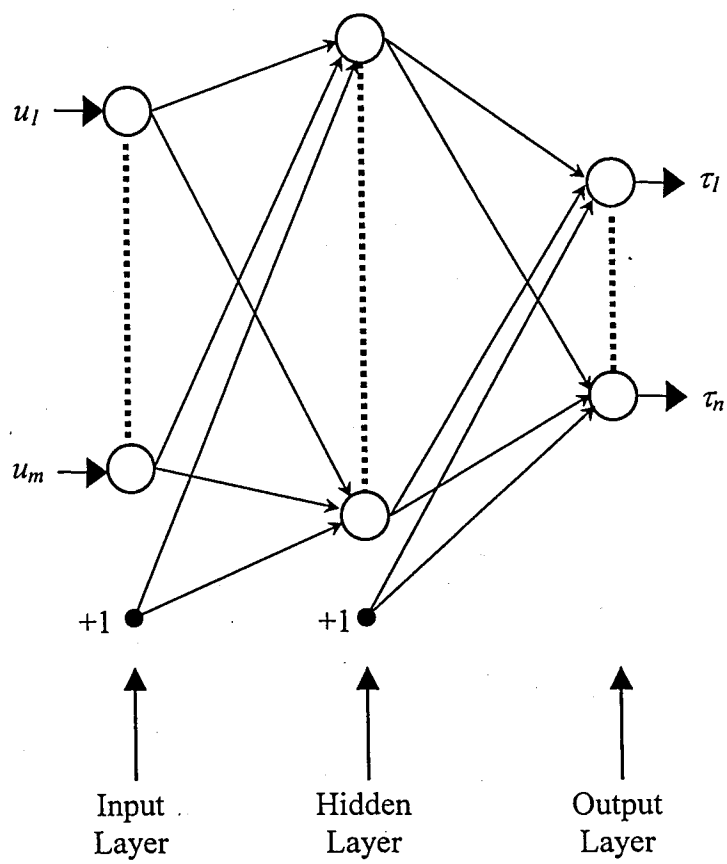

Figure 2.2. Structure of a Feedforward Neural Network

The overall realization performed by the structure depicted can be expressed in the vector form as described below.

$$\underline{\tau} = \underline{\Omega}\left(\Gamma_2 \underline{\Omega}\left(\Gamma_1 \underline{u} + \underline{v}_1\right) + \underline{v}_2\right) \tag{2.4}$$

In the above equation, if the number of hidden neurons is denoted by h_1, the matrix connecting the input vector and the hidden layer (Γ_1) is of dimension $h_1 \times m$, and that connecting the hidden layer output vector and the output vector (Γ_2) is of dimension $n \times h_1$. The vector of neuronal activation functions is denoted by $\underline{\Omega}$, which is a $h_1 \times 1$ vector function. Similarly, the dimensions of the bias vectors $\underline{v}_1$ and $\underline{v}_2$ can be obtained as $h_1 \times 1$ and $n \times 1$ respectively. It is straightforward to generalize the formulation for structures having more than one hidden layer.

It should be noted that the neurons in the output layer, in this thesis, employ linear activation functions (i.e. ADALINEs are used), nevertheless they could have a nonlinear behavior. The selection of the neuronal nonlinearity is also a design flexibility exploited by the designer. Although there are many variations, the widespread choice for the entries of the vector function $\underline{\Omega}$ is the hyperbolic tangent function, which is adopted throughout the thesis [1,42-44].

The control applications of artificial neural networks have extensively been analyzed in the existing literature [13-14,24,26-28,37,32-47]. The central question for the exploitation of the versatility of these systems for the fulfillment of a specific task is how the parameters of the structure are selected. Practically, the best parameter set is reached iteratively rather than by a direct assignment. The search for achieving the best parameter set i.e. the entries of the matrices Γ_1 and Γ_2 and those of $\underline{v}_1$ and $\underline{v}_2$ for the structure in Figure 2.2, a number of techniques are frequently utilized. These are based on the minimization of a quadratic cost function as

$$J = \frac{1}{2} \sum_{p=1}^{P} \left(\underline{\tau}_d^p - \underline{\tau}^p\right)^T \left(\underline{\tau}_d^p - \underline{\tau}^p\right) \tag{2.5}$$

The cost measure described above is differentiable with respect to each element of the entire parameter set. The operation of searching for the best parameter combination in a multidimensional space is called "learning". The terms "parameter tuning", "parameter adjustment" or "parameter adaptation" can be used for the same purpose interchangeably.

In (2.5), P is the total number of training pairs included in the training data set. For the on-line training strategies, the parameter adjustment mechanism processes the instant values of the observed quantities and the training data set includes only one pair for each time instant. The modified form of the cost function in (2.5) can be reformulated as

$$J = \frac{1}{2}(\underline{\tau}_d - \underline{\tau})^T(\underline{\tau}_d - \underline{\tau}) \tag{2.6}$$

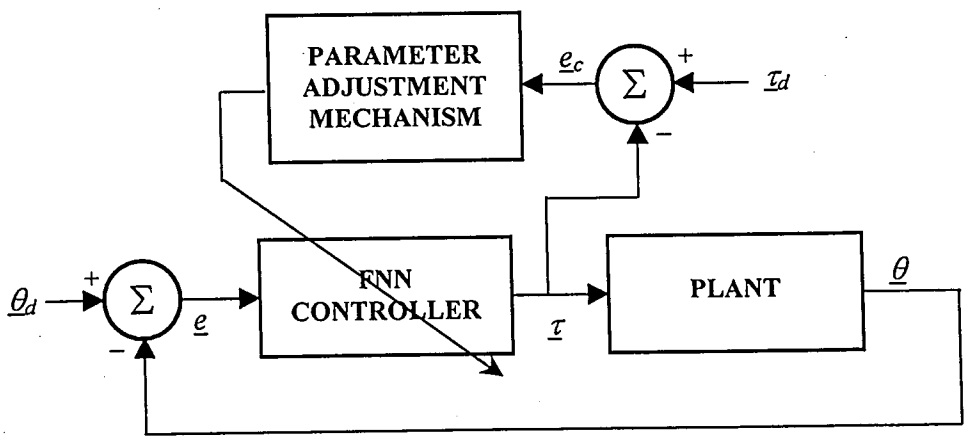

Figure 2.3. Structure of the feedback control system as imposed by the theory

The use of the FNN architecture depicted in Figure 2.2 as a controller is illustrated in Figure 2.3, in which the plant under control is in an ordinary feedback loop. The controller uses the discrepancy between the reference and observed state vectors and produces a control signal. In the outer loop, a parameter adjustment activity is performed to reduce the instantaneous value of the cost function in (2.6).

The parameter adjustment block in Figure 2.3 uses the error vector ($\underline{e}_c$) on the produced control signal and evaluates some parametric displacement values, which are to be added to the current values of the parameters. The difficulty in implementing such a

mechanism is the unavailability of the desired values of the controller outputs denoted by $\underline{\tau}_d$ [48]. Therefore the measure of the controller performance is practically constructed using the tracking error vector, which leads to the following cost function.

$$J = \frac{1}{2}\underline{e}^T \underline{e} \tag{2.7}$$

In Figure 2.4, the modified structure of the feedback control system is sketched, and is used throughout the thesis. It is to be noted that the adjustment mechanism block in Figure 2.3 may include the information about the plant, e.g. Jacobian or an identified model, or an estimator for the desired output of the controller [3,49-51]. In the fourth and fifth sections, an in-depth consideration is given to the training strategies for the architectures discussed in this section.

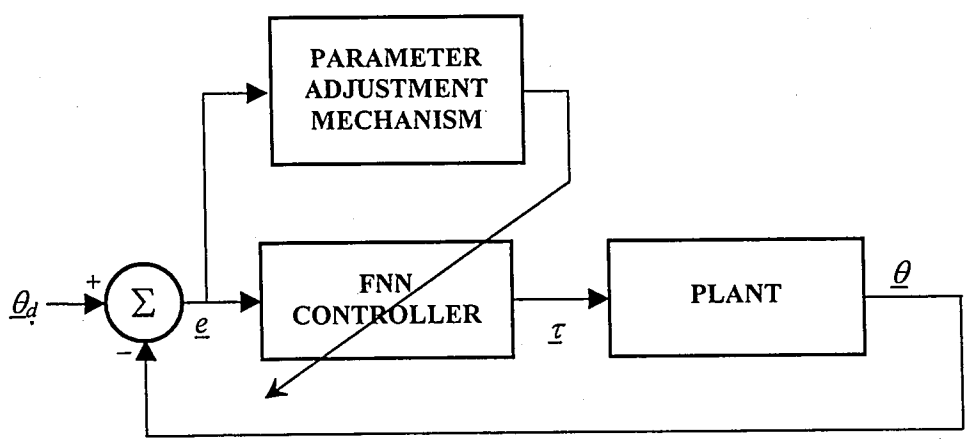

Figure 2.4. Structure of the feedback control system as imposed by the practice

2.3. Gaussian Radial Basis Function Neural Networks

The fundamental operation in most neural network architectures existing in the literature is the evaluation of a dot product of an input vector and a parameter vector, and then passing the evaluated quantity through a nonlinear activation function. The yield of the described process is the output of the neuron. However, in another class of neural networks the neuron output is evaluated by combining the values of some appropriately defined basis functions. The networks using basis functions constitute several number of

hidden neurons, the activation level of which depend on the distance between the input vector and a prototype vector [1-3]. The structure of a m-input, h-hidden neuron and single-output GRBFNN is illustrated in Figure 2.5.

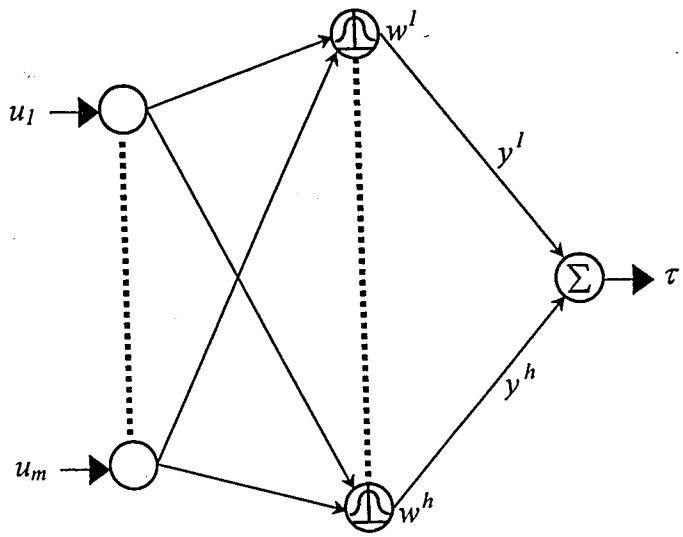

Figure 2.5. Structure of a GRBFNN

Gaussian radial basis function neural networks constitute a special class of the structures described above. A hidden neuron in GRBFNN uses a Gaussian nonlinearity as the activation function, namely

$$\mu_{ij}(u_j) = \exp\left\{-\left(\frac{u_j - c_{ij}}{\alpha_{ij}}\right)^2\right\} \tag{2.8}$$

In this definition, i indexes the neuron order in the hidden layer while j is for input vector ordering. The prototype vector is comprised of the c_{ij} variables, which characterize the centers of the Gaussian functions. The variable α_{ij} determines how the function (μ_{ij}) spreads over the domain of its input space (u_j). The output of the i^{th} neuron in the structure is denoted by w^i and is evaluated through the use of

$$w^i = \prod_{j=1}^{m} \mu_{ij}(u_j) \tag{2.9}$$

The overall output of the structure depicted is evaluated by a weighted sum of the responses of the neurons contained in the hidden layer and is described by

$$\tau = \sum_{i=1}^{h} y^i w^i = \underline{y}^T \underline{w} \tag{2.10}$$

In (2.10), y^i denotes the weight determining the effect of the i^{th} hidden neuron output on the overall network response.

What makes the use of GRBFNN attractive for control engineering applications is that the expert knowledge can be used in the initialization of the parameters of the basis functions. More explicitly, a hidden neuron is fired if the input vector is close to the prototype vector of this neuron. If the designer knows roughly what the GRBFNN structure must perform when the input vector is close to the prototype vector of each one of the hidden neurons, the entries of the $\underline{y}$ vector can be assigned by using this knowledge.

Obviously, by setting the parameters of the structure randomly and implementing a suitable learning algorithm can also lead to the achievement of the design specifications. But the utilization of the expert knowledge will shorten the time during which an undesired transient response is likely to arise.

The applications of GRBFNN for the identification and control purposes are considered in [38-39,52] and those for image/pattern recognition are discussed in [2].

2.4. Standard Fuzzy Systems

Contrary to what is done in the realm of predicate logic, representation of knowledge by fuzzy quantities can provide extensive degrees of freedom if the task to be achieved can better be expressed in words than in numbers. The concept of fuzzy logic in this sense can be viewed as a generalization of binary logic and refers to the manipulation of knowledge

with sets, whose boundaries are unsharp [53]. Therefore the paradigm offers a possibility of designing intelligent controllers operating in an environment, in which the conditions are inextricably intertwined, subject to uncertainties and impreciseness.

Understanding the information content of fuzzy logic systems is based on the subjective judgements, intuitions and the experience of an expert. From this point of view, a suitable way of expressing the expert knowledge is the use of *IF antecedent THEN consequent* rules, which can easily evaluate the necessary action to be executed for the current state of the system under investigation.

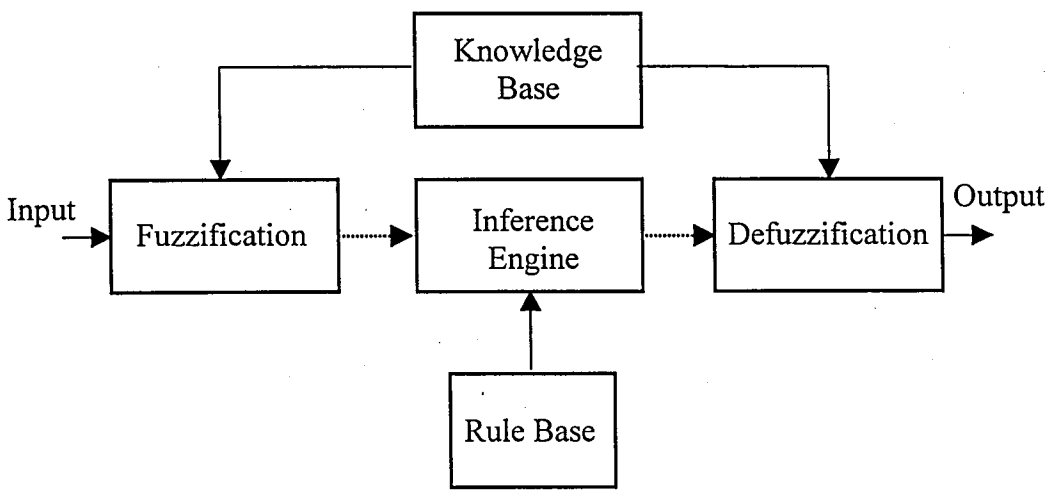

Figure 2.6. Structure of a fuzzy controller

Structurally, a fuzzy controller is comprised of five building blocks, namely, fuzzification, inference engine, knowledge base, rule base, and defuzzification as shown in Figure 2.6. Since the philosophy of the fuzzy models is based on the representation of knowledge in fuzzy domain, the variables of interest are graded first. This grading is performed through the evaluation of membership values of each input variable in terms of several class definitions. According to the definition of a membership function, how the degree of confidence changes over the domain of interest is characterized. This grading procedure is called fuzzification. In the knowledge base, the parameters of membership functions are stored. Rule base contains the cases likely to happen, and the corresponding actions for those cases through linguistic descriptions, i.e. the IF-THEN statements. The inference engine emulates the expert's decision making in interpreting and applying knowledge about how the best fulfillment of the task is achieved. Finally, the defuzzifier

converts the fuzzy decisions back onto the crisp domain [54]. As depicted in Figure 2.6, the outputs of the fuzzifier and the outputs of the inference engine carry fuzzy information.

The architecture used in this thesis is a Standard Fuzzy System (SFS) proposed by Wang [4-5] and illustrated in Figure 2.7. Among many other alternatives existing in the literature, this system uses algebraic product operator for the aggregation of the rule premises and bell-shaped membership functions described as

$$\mu_{ij}(u_j) = \frac{1}{1 + \left| \frac{u_j - c_{ij,}}{a_{ij}} \right|^{2b_{ij}}} \tag{2.11}$$

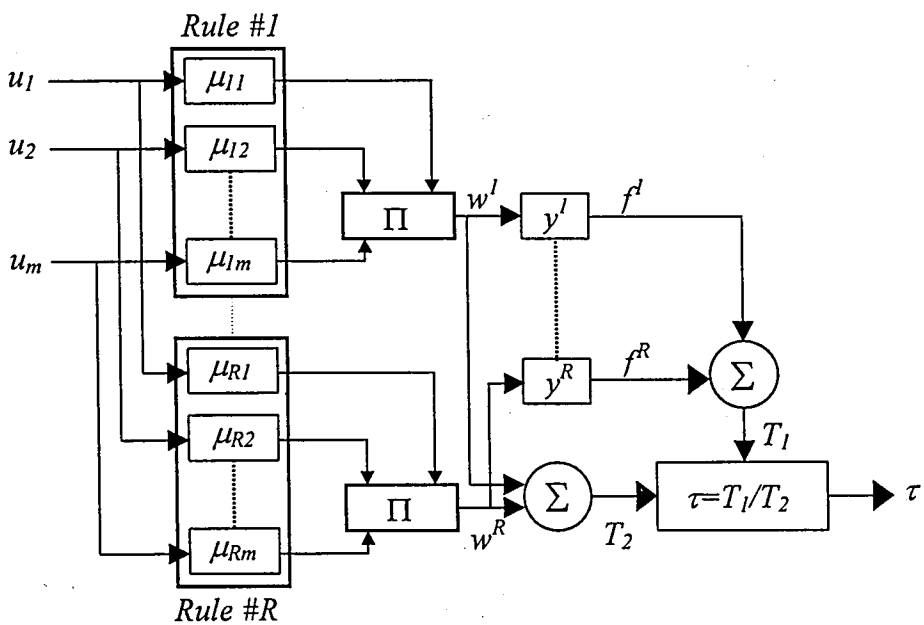

Figure 2.7. Structure of a Standard Fuzzy System

In (2.11), c_{ij} defines the center of the membership function; a_{ij} and b_{ij} characterize the slope and flatness of the function respectively. For the fuzzy system illustrated in Figure 2.6, the i^{th} rule in the rule base has the following structure.

IF u_1 *is* U_1^i *AND* u_2 *is* U_2^i *AND ... AND* u_m *is* U_m^i

THEN $f^i = y^i$

In the IF part of this representation, the lowercase variables denote the inputs and the uppercase variables stand for the fuzzy sets corresponding to the domain of each linguistic label. The THEN part is comprised of the prescribed decision in the form of a scalar number denoted by y^i.

The data flow through the architecture discussed entails the evaluation of the activation level (or the firing strength) of each rule contained in the rule base. For the i^{th} rule, this quantity is denoted by w^i and is given as

$$w^i = \prod_{j=1}^{m} \mu_{ij}(u_j) \tag{2.12}$$

in which the rule premises are aggregated with the algebraic product operator. Without loss of generality, one should notice the similarity between (2.9) and (2.12), which establishes a functional connection between the hidden neuron outputs in GRBFNN and aggregation of rule premises in SFS with product aggregation operation [3].

If the rule base is assumed to have R rules, the vector of firing strengths can be normalized and the i^{th} entry of the resulting vector can be described as

$$w_n^i = \frac{w^i}{\sum_{k=1}^{R} w^k} \tag{2.13}$$

The defuzzification strategy adopted is the weighted average given as

$$\tau = \frac{\sum_{i=1}^{R} f^i w^i}{\sum_{i=1}^{R} w^i} = \sum_{i=1}^{R} y^i w_n^i = \underline{y}^T \underline{w}_n \tag{2.14}$$

which determines the crisp output of the fuzzy system.

It should be noted that the output of the fuzzy system discussed is scalar. Since the aim is to use the structure in control applications, it is straightforward to design such a structure for each control input of the plant.

It has been discussed in the section 2.2 that adaptive fuzzy controllers in most applications suffer from the unavailability of the desired values of the control signal. Therefore, the feedback control structure depicted in Figure 2.4 is utilized in the applications of fuzzy control strategies, which adapt the parameters to reduce some norm of the tracking error.

2.5. Adaptive Neuro-Fuzzy Inference Systems

Autonomy is one of the most important characteristics required from an intelligent system. A basic requirement in this context is the ability to refresh and to refine the information content of the dynamics of the system. It therefore requires a careful consideration in the realm of engineering practice. From a systems and control engineering point of view, the designer is motivated by the time-varying nature of structural and environmental conditions to realize controllers that can accumulate the experience and improve the mapping precision [3,55]. Fuzzy methodologies are good in achieving the former and neural networks the latter. The integration of these methodologies that exploit the strength of each collectively and synergistically is therefore a natural way to follow. The early works in the direction of synthesizing hybrid neuro-fuzzy systems consider the neural networks having fuzzy weights and biases [56], uncertainty processing by different components of a network [57] and structurally hybridized architectures [58], all seeking a suitable integration of neural and fuzzy systems.

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) aim to utilize the verbal power of fuzzy systems and the numerical power due to neural networks by an appropriate combination. Such a combination also enables a biased assignment of the initial values of the parameters of the structure utilizing the expert knowledge on the system under investigation.

This thesis considers the ANFIS structure with first-order Sugeno model containing R rules as shown in Figure 2.8.

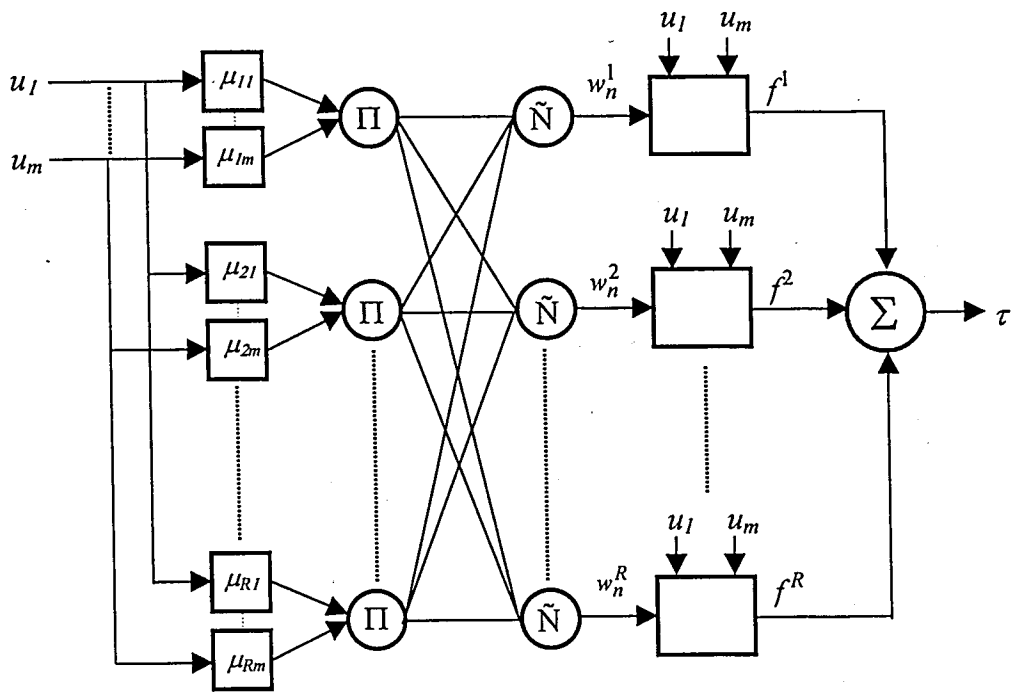

Figure 2.8. Structure of an Adaptive Neuro-Fuzzy Inference System

Bell-shaped membership functions as defined in (2.11) with product inference rule are used at the fuzzification level. The inference engine outputs the firing strengths for each rule, which is described in (2.12). The vector of the firing strengths is normalized through the use of (2.13), the function of which is depicted as circles with Ñ in Figure 2.8, and the resulting vector is defuzzified by utilizing the first order Sugeno model [5]. An m-input one-output ANFIS architecture is depicted in Figure 2.8, for which the rule structure is described below. When the input vector $\underline{u}$ is constrained to the fuzzy range characterized by the IF part of the i^{th} rule, the output is a linear function of the input variable.

$$IF \qquad u_1 \text{ is } U_1^i \text{ AND } u_2 \text{ is } U_2^i \text{ AND } \ldots \text{ AND } u_m \text{ is } U_m^i$$

$$THEN \qquad f^i = Y_{i,1}u_1 + \ldots + Y_{i,m}u_m + Y_{i,m+1}$$

The overall realization performed by the ANFIS structure is described in (2.15), in which Y is an Rx(m+1)-dimensional matrix used in the defuzzification stage and $\underline{u}_A$ is the vector of augmented inputs described explicitly in (2.2) with u_{m+1} being unity.

$$\tau = \frac{\sum_{i=1}^{R} f^i w^i}{\sum_{i=1}^{R} w^i} = \sum_{i=1}^{R} f^i w_n^i = \underline{w}_n^T Y \, \underline{u}_A \tag{2.15}$$

The use of ANFIS structure for control purposes is illustrated in Figure 2.4, where ANFIS takes the place of FNN controller. If the input vector is composed of the error and the rate of the error as discussed in [25,59-60], ANFIS structure can be viewed as an analogue of a collection of PD controllers operating on local regions of the phase space. A soft switching between the local regions takes place because of the fuzzy boundaries of the activation region of each local controller.

2.6. Learning, Computational Intelligence and Control Engineering

Although there is not a standard definition, the process of improving the future performance of a flexible structure by tuning the parameters can be described as learning. The approaches existing in the literature employ various techniques in achieving the desired parameter set, which is unknown and which require an iteratively evolving search mechanism. It is to be noted that the most common technique that can be used in performing a suitable search operation in a multidimensional parameter space is based on the use of an appropriately defined cost function. Alternatively, the search procedure can be implemented without using the derivative information; such as is done by the use of methods adapted from the evolutionary computation or random search methods [3].

Error Backpropagation (EBP) technique [61] and Levenberg-Marquardt (LM) optimization technique [62] are the frequently used techniques for adapting the parameters of a computationally intelligent architecture. Both of the approaches are based on the utilization of gradient information and necessitate the differentiability of the nonlinear activation functions existing in the architecture with respect to the parameter to be updated

[63], and include some heuristics for improved realization performance. These typically concern the selection of learning rate, momentum coefficient, and adaptive learning rate strategies in EBP or stepsize considerations in LM. However, the problem of convergence or that of maintaining the bounded parameter evolution is still there. More explicitly, the learning strategy is not protected against disturbances, which may excite the undesired internal modes of EBP or LM approaches. The multidimensionality of the problem is another difficulty in coming up with a thorough analysis distinguishing the useful training information and disturbance related useless information.

Apart from the algorithmic problems, the training of the computationally intelligent controllers entails the target outputs of the architecture. From the control engineering point of view, as illustrated in Figure 2.3, the unavailability of the target control sequence and the existence of an inexact or a simplified model of the plant constitute a difficulty. A suitable way of obtaining the equivalent error on the applied control inputs is to use another computationally intelligent system identifying the dynamics of the plant on-line as shown in Figure 2.9 [38-39,47,50,64-66]. However, this operation increases the computational burden because of the considerable amount of floating point operations to be performed at each control period. Therefore, obtaining a suitable transformation from the tracking error vector to the control signal error is impossible without making concessions from the point of computational complexity. In this respect, the structure depicted in Figure 2.4, which is used in practice, implies that the uncertainties stemming from an imperfect transformation are to be alleviated through an appropriate learning process, which utilize the tracking error vector.

Since the ultimate goal of the design is to achieve a good tracking precision, reducing the adverse effects of the disturbances and those of the approximate transformation between the tracking error vector and controller output requires that the adopted learning dynamics should be robustified. This steers the designer to seek for methods known in the conventional design framework. From this point of view, a learning strategy based on Variable Structure Systems (VSS) theory constitutes a good candidate for eliminating the adverse effects of disturbances. An in-depth analysis for the use of VSS theory in tuning the parameters of intelligent controllers is presented in this thesis.

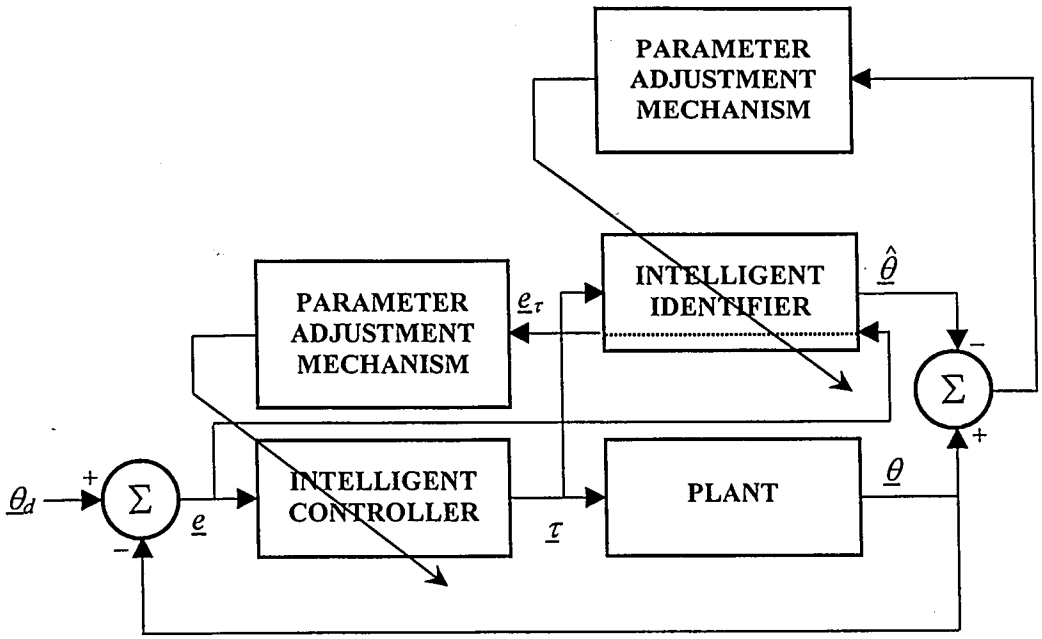

Figure 2.9. Use of an identifier for obtaining the equivalent torque error ($\underline{e}_\tau$)

3. AN INTRODUCTION TO VARIABLE STRUCTURE SYSTEMS THEORY FOR SLIDING MODE CONTROL

Design of a controller for a system, whose salient features are represented in the model, which the design is to be based upon, requires the alleviation of plant/model mismatches. These mismatches can be classified into two categories, namely, structured and unstructured. In the former, uncertainties are on the values of the parameters, while the latter concerns the unmodeled dynamics existing in the plant [67]. For the structured uncertainties, the framework of adaptive control offers well formulated solutions based on the estimation of slowly varying unknown parameters. The design of the controller can then be done using the estimates. However, an oversimplified or an imperfect model of the plant necessitates a very robust controller to maintain the tracking performance. Use of VSS theory is one particular approach for achieving this task [17,68].

In this section, an introduction to VSS theory for Sliding Mode Control (SMC) is discussed briefly. Although the theory scrutinizes various conditions that are likely to happen in practice, the discussion in this section is confined to what constitutes a basis for the subject of this thesis.

3.1. Variable Structure Control with Lyapunov Function Approach

Consider a nonlinear and non-autonomous system represented as

$$\theta_i^{(r_i)} = f_{p_i}(\underline{\theta}) + \sum_{j=1}^{n} d_{ij}\tau_j \qquad i=1,\ldots,n \tag{3.1}$$

The system under control is a multi-input multi-output one driven by the nx1-dimensional input vector $\underline{\tau}$. The state vector and the input vector of the system in (3.1) are defined as

$$\underline{\theta} = \left[\theta_1 \quad \dot{\theta}_1 \quad \ldots \quad \theta_1^{r_1-1} \quad \cdots \quad \theta_n \quad \dot{\theta}_n \quad \cdots \quad \theta_n^{r_n-1}\right]^T \tag{3.2}$$

$$\underline{\tau} = [\tau_1 \quad \tau_2 \quad \cdots \quad \tau_n]^T \tag{3.3}$$

respectively. The system in (3.1) can be rewritten as

$$\dot{\underline{\theta}} = \underline{f}_p(\underline{\theta}) + D\,\underline{\tau} \tag{3.4}$$

in which D denotes the input gain matrix. Without loss of generality, the vector of sliding surfaces is chosen as a linear function of the components of the tracking error vector described as

$$\underline{e} = [e_1 \quad e_2 \quad \cdots \quad e_n]^T = [\theta_1 - \theta_{1d} \quad \theta_2 - \theta_{2d} \quad \cdots \quad \theta_n - \theta_{nd}]^T \tag{3.5}$$

which is the discrepancy between the values of the measured state vector $\underline{\theta}$ and the desired state vector $\underline{\theta}_d$. The sliding surface $\underline{s}_p(\underline{e})$ is a nx1 vector and is defined as

$$\underline{s}_p(\underline{e}) = G\underline{e} = G(\underline{\theta} - \underline{\theta}_d) \tag{3.6}$$

The widespread selection of the matrix G is such that the i^{th} sliding surface function has the form

$$s_{p_i}(e_i) = \left(\frac{d}{dt} + \lambda_i\right)^{r_i - 1} e_i \tag{3.7}$$

in which, λ_i is a strictly positive constant. Let V_p be a candidate Lyapunov function given as

$$V_p(\underline{s}_p) = \frac{1}{2}\underline{s}_p^T \underline{s}_p \tag{3.8}$$

If the prescribed control signal satisfies

$$\dot{V}_p(\underline{s}_p) = -\underline{s}_p^T \xi \operatorname{sgn}(\underline{s}_p) \tag{3.9}$$

the negative definiteness of the time derivative of the Lyapunov function in (3.8) is ensured. In (3.9), ξ is a positive definite diagonal matrix of dimension nxn. If the time derivative of the Lyapunov function in (3.8) is evaluated, the quantity obtained must be equal to the expression in (3.9). This can be stated as

$$\underline{s}_p^T \dot{\underline{s}}_p = -\underline{s}_p^T \xi \operatorname{sgn}(\underline{s}_p) \tag{3.10}$$

From (3.4) and (3.6), one can write the time derivative for $\underline{s}_p$ as

$$\dot{\underline{s}}_p = -G\dot{\underline{\theta}}_d + G\left(\underline{f}_p(\underline{\theta}) + D\underline{\tau}\right) \tag{3.11}$$

Substituting (3.11) into (3.10), the control signal can be constructed as

$$\underline{\tau} = \underline{\tau}_{eq} + \underline{\tau}_c \tag{3.12}$$

in which the first term is the equivalent control term and the second term is the corrective control term. Both terms are explicitly given as in (3.13) and (3.14) respectively. For the existence of the mentioned components, the matrix GD must not be rank deficient.

$$\underline{\tau}_{eq} = -(GD)^{-1}\left(G\underline{f}_p(\underline{\theta}) - G\dot{\underline{\theta}}_d\right) \tag{3.13}$$

$$\underline{\tau}_c = -(GD)^{-1}\xi \operatorname{sgn}(\underline{s}_p) \tag{3.14}$$

In the literature, equivalent control is considered as the low frequency (average) component of the control signal. Because of the discontinuity on the sliding surface, the corrective term brings a high rate component [69-70].

If $\underline{e}(0)=\underline{0}$, the tracking problem can be considered as keeping $\underline{e}$ on the sliding surface, however, for nonzero initial conditions, the strategy must enforce the state trajectories towards the sliding surface, which is ensured by imposing the equality in (3.9). For the case of nonzero initial conditions, the phase until the error vector hits the sliding surface is called the reaching mode, the dynamic characteristics of the system during which is determined by the control strategy adopted. Application of the control input formulated in (3.12) imposes the dynamics describes as

$$\dot{s}_p = -\xi \operatorname{sgn}(s_p) \tag{3.15}$$

which clearly enforce the error vector towards the sliding surface. Once the sliding surface is reached, the equality in (3.7) becomes equal to zero; and this enforces the error vector to move towards the origin.

Lastly in this section, it is beneficial to mention the problems associated with the SMC strategy in devising variable structure controllers. The first problem stems from the discontinuity of the control signal about the sliding surface. After the reaching phase, the constructed form of the control signal enforces the system states to lie on the sliding surface, along which a discontinuous control action is of interest. This fact introduces high frequency components into the prescribed form of the control signal, the application of which may excite the undesired high frequency dynamics of the plant under control and can lead to unpredictable instabilities. The problem of having such high frequency components in the control signal is referred to as chattering in the related literature [9,18,68]. For the alleviation of the chattering phenomenon, various techniques have been reported in the literature, which postulate the form of the control signal for the plants, whose governing equations are linear in the control term [71-78].

Secondly, if the observed state variables are noisy, the control signal is adversely affected by this parasitic dynamics. Since the form of control signal entails the sign of a measured quantity, which is very close to zero, the problem can be solved by introducing a thin boundary layer, in which the control signal is smoothed out [67].

3.2. Visualization of Sliding Mode Control for Second Order Systems

For a single-input single-output second order system (n=1, r_1=2) of the form (3.1), the error and the rate of error define the sliding surface, which passes through the origin with slope equal to $-\lambda_1$. The surface for this case is a line in fact, and is explicitly given in (3.16) and depicted in Figure 3.1.

$$s_p(e) = \dot{e} + \lambda_1 e \tag{3.16}$$

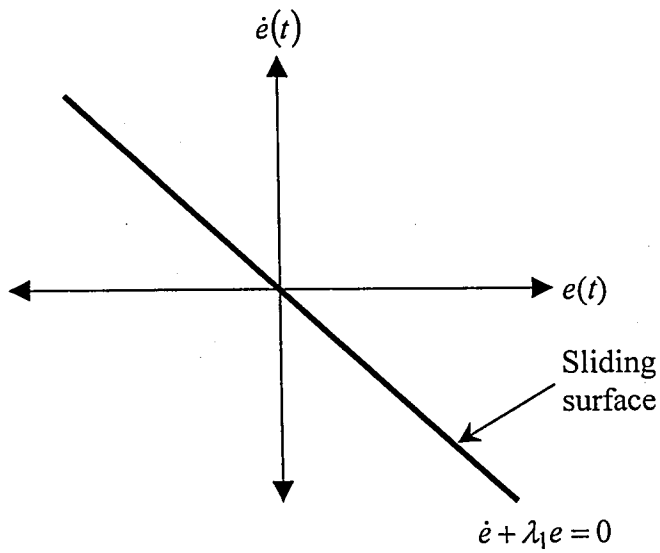

Figure 3.1. Sliding surface for a second order system

It should be noted here that the form of the sliding surface does not have to assume a linear function of its arguments. The selection shown in Figure 3.1 ensures that once the tracking error reaches the sliding surface, it remains on it forever and converges to origin. Additionally, although the surface shown in Figure 3.1 is a locus in the phase space, it is a dynamics characterizing the behavior of the error [68].

3.3. A Discussion on Computational Intelligence and Sliding Mode Control Relevance

Sliding mode control is a specific approach for designing a controller, whose most popular form and the problems associated with it have been discussed in the preceding

discussion. In the literature, methods of computationally intelligent systems have widely been used for compensating the problems arising in SMC applications. These can be classified into two groups.

In the first group, the standard procedure of SMC design is followed and the parameters of the design or the uncertain terms in the plant dynamics are constructed by the use of computationally intelligent systems [12,77-82].

In the second group, a computationally intelligent architecture is used as the controller and the parameters of the controller are adjusted such that the plant enters into a sliding mode [78-79,83].

A third possible and novel class of combining the methods of computational intelligence and SMC is to drive the system to a sliding mode while enforcing a similar dynamics on the parameters of the computationally intelligent system acting as the controller. In the next two sections, an analysis and design aiming such a combination is discussed.

4. PARAMETER TUNING BASED ON A SINGLE-TERM LYAPUNOV FUNCTION

In this section, a novel method for driving the dynamics of a single-input, second order nonlinear system to a sliding mode is discussed. The approach is based on SMC methodology, i.e., the system under control is driven towards a sliding regime by tuning the parameters of the controller. In this tuning loop, the parameters of the controller are adjusted such that a zero learning-error level is reached in one-dimensional phase space defined on the output of the controller. For the purpose of tuning the controller parameters, it is shown that there exists a relation between the sliding line defined for the plant under control and the zero learning-error level for the controller. The relation between these two quantities is analyzed and the conditions for observing an equivalent sliding regime on the controller parameters are discussed.

4.1. Definitions and the Formulation of the Problem Using ADALINE Structure

Consider the three-input one-output ADALINE structure, which is to be used as the controller, depicted in Figure 2.1. The adjustable parameter vector and the input vector of the structure are described as

$$\underline{\phi} = [\phi_1 \quad \phi_2 \quad \phi_3]^T \tag{4.1}$$

$$\underline{u}_A = [e \quad \dot{e} \quad 1]^T = [u_1 \quad u_2 \quad 1]^T \tag{4.2}$$

In (4.2), the symbol e denotes the tracking error, which is the discrepancy between the response of the system under control and the reference signal ($e = \theta - \theta_d$). The input output relation of the controller is given as

$$\tau = \phi_1 u_1 + \phi_2 u_2 + \phi_3 \tag{4.3}$$

The structure of the control system is an ordinary feedback loop as illustrated in Figure 4.1. The definitions of the sliding surface $s_p(e,\dot{e})$ and that of zero learning-error level $s_c(\tau,\tau_d)$, which are seen in this figure, are given as

$$s_p(e,\dot{e}) = \dot{e} + \lambda e \tag{4.4}$$

where, λ is the slope of the sliding surface and

$$s_c(\tau,\tau_d) = \tau - \tau_d \tag{4.5}$$

where, τ_d is the desired output of the controller and is unknown.

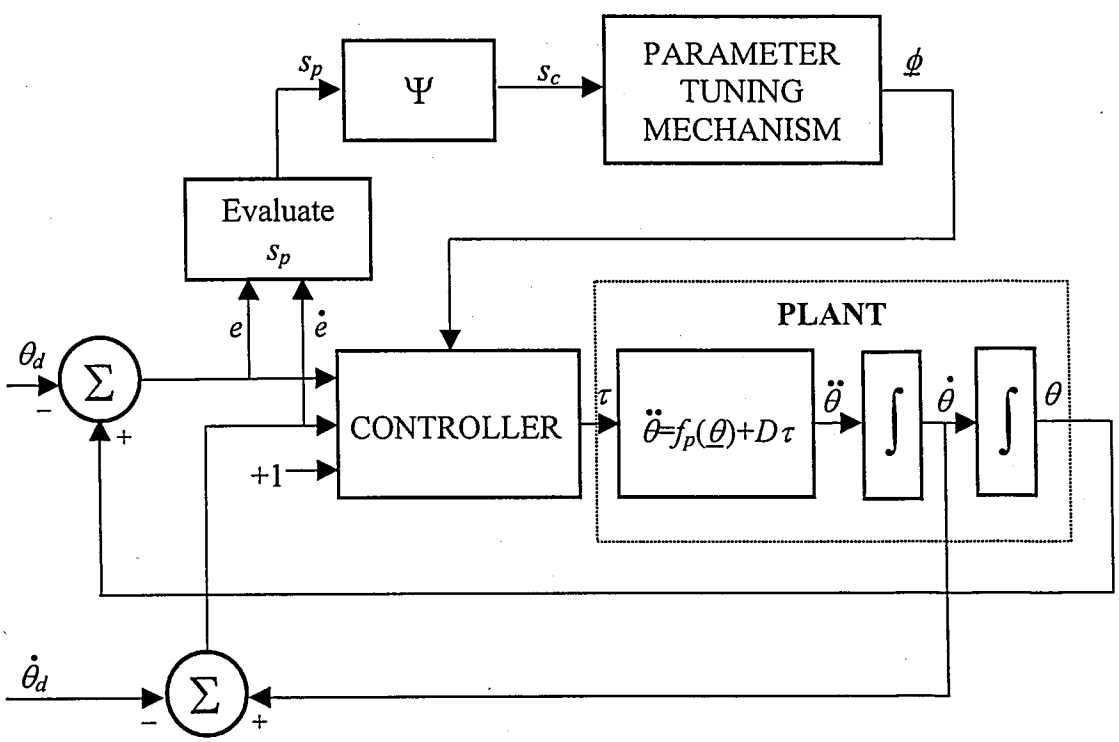

Figure 4.1. Structure of the control system

In order not to be in conflict with the physical reality, the designer must impose the following inequalities, the truth of which states that the parameters of the controller (ϕ), the

time derivative of the input signal ($\dot{\underline{u}}_A$) and the time derivative of the desired output of the controller ($\dot{\tau}_d$) remain bounded.

$$\left\|\underline{\phi}\right\| = \sqrt{\left(\sum_{i=1}^{m+1} \phi_i^2\right)} \leq B_\phi \tag{4.6}$$

$$\left\|\dot{\underline{u}}_A\right\| \leq B_{\dot{u}_A} \tag{4.7}$$

$$\left\|\dot{\tau}_d\right\| \leq B_{\dot{\tau}_d} \tag{4.8}$$

***Theorem* 4.1.** For a multi-input single-output flexible structure, whose output is a linear function of the adjustable parameters, the adaptation mechanism described as

$$\dot{\underline{\phi}} = -\frac{\underline{u}_A}{\underline{u}_A^T \underline{u}_A} K \mathrm{sgn}(s_c) \tag{4.9}$$

enforces the parameters to values resulting in zero learning-error level in one-dimensional phase space, whose argument is defined by (4.5). In the adaptation rule described above, K is a sufficiently large constant satisfying the inequality given as

$$K > B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d} \tag{4.10}$$

The adaptation mechanism in (4.9) drives an arbitrary initial value of s_c to zero in finite time denoted by t_h satisfying the inequality

$$t_h \leq \frac{|s_c(0)|}{K - (B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d})} \tag{4.11}$$

Proof: Consider the Lyapunov function candidate as

$$V_c = \frac{1}{2} s_c^2 \tag{4.12}$$

In order to reach the zero learning-error level (s_c=0), the time derivative of (4.12) must be negative definite, which is given as

$$\begin{aligned}
\dot{V}_c &= \dot{s}_c s_c \\
&= (\dot{\tau} - \dot{\tau}_d) s_c \\
&= \left(\underline{\dot{\phi}}^T \underline{u}_A + \underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d \right) s_c \\
&= -K \mathrm{sgn}(s_c) s_c + \left(\underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d \right) s_c \\
&= -K |s_c| + \left(\underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d \right) s_c \\
&\le \left(-K + B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d} \right) |s_c|
\end{aligned} \tag{4.13}$$

Since $\underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d \le B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d}$ holds always true, it is apparent that the condition in (4.10) ensures the negative definiteness of the time derivative of the selected Lyapunov function. If one evaluates $\dot{s}_c$ with the aid of (4.9), one obtains

$$\dot{s}_c = -K \mathrm{sgn}(s_c) + \underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d \tag{4.14}$$

The solution to the differential equation in (4.14) can be given as

$$s_c(t)-s_c(0)=-Kt\,\mathrm{sgn}(s_c(0))+\int_0^t\left(\underline{\phi}^T(\sigma)\dot{\underline{u}}_A(\sigma)-\dot{\tau}_d(\sigma)\right)d\sigma \tag{4.15}$$

At $t = t_h$, $s_c(t_h) = 0$;

$$-s_c(0)=-Kt_h\,\mathrm{sgn}(s_c(0))+\int_0^{t_h}\left(\underline{\phi}^T(\sigma)\dot{\underline{u}}_A(\sigma)-\dot{\tau}_d(\sigma)\right)d\sigma \tag{4.16}$$

By multiplying both sides of (4.16) by $-\mathrm{sgn}(s_c(0))$, one obtains

$$\begin{aligned}|s_c(0)| &= Kt_h-\left(\int_0^{t_h}\left(\underline{\phi}^T(\sigma)\dot{\underline{u}}_A(\sigma)-\dot{\tau}_d(\sigma)\right)d\sigma\right)\mathrm{sgn}(s_c(0))\\ &\geq Kt_h-\left(B_\phi B_{\dot{u}_A}+B_{\dot{\tau}_d}\right)t_h\end{aligned} \tag{4.17}$$

which implies hitting in finite time as described by the inequality in (4.11). □

***Theorem* 4.2.** If the system enters the sliding mode s_c=0 and remains in it thereafter, then the parameters of the flexible controller, $\underline{\phi}$, will evolve boundedly.

Proof: In the sliding mode, s_c=0 and $\dot{s}_c$=0. Based on this, the following derivation can be made.

$$\dot{s}_c=\dot{\tau}-\dot{\tau}_d \tag{4.18}$$

Substituting (4.3) into (4.18),

$$\dot{s}_c=\dot{\underline{\phi}}^T\underline{u}_A+\underline{\phi}^T\dot{\underline{u}}_A-\dot{\tau}_d=0 \tag{4.19}$$

which implies the following relations.

$$\underline{u}_A^T \dot{\underline{\phi}} = -\dot{\underline{u}}_A^T \underline{\phi} + \dot{\tau}_d$$

$$= -\frac{\underline{u}_A^T \underline{u}_A}{\underline{u}_A^T \underline{u}_A} \dot{\underline{u}}_A^T \underline{\phi} + \frac{\underline{u}_A^T \underline{u}_A}{\underline{u}_A^T \underline{u}_A} \dot{\tau}_d \tag{4.20}$$

$$= \underline{u}_A^T \left(-\frac{\underline{u}_A}{\underline{u}_A^T \underline{u}_A} \dot{\underline{u}}_A^T \underline{\phi} + \frac{\underline{u}_A}{\underline{u}_A^T \underline{u}_A} \dot{\tau}_d \right)$$

The above equality can be rewritten as follows.

$$\underline{u}_A^T \left(\dot{\underline{\phi}} + \frac{\underline{u}_A}{\underline{u}_A^T \underline{u}_A} \dot{\underline{u}}_A^T \underline{\phi} - \frac{\underline{u}_A}{\underline{u}_A^T \underline{u}_A} \dot{\tau}_d \right) = 0 \tag{4.21}$$

Since the entries of the vector $\underline{u}_A$ cannot be linearly dependent for all time [67], the equality in (4.21) imposes the following differential equation form in the sliding mode.

$$\dot{\underline{\phi}} = -\frac{\underline{u}_A \dot{\underline{u}}_A^T}{\underline{u}_A^T \underline{u}_A} \underline{\phi} + \frac{\underline{u}_A}{\underline{u}_A^T \underline{u}_A} \dot{\tau}_d \tag{4.22}$$

The solution to the above equation is

$$\underline{\phi}(t) = \Phi(t,0)\underline{\phi}(0) + \int_0^t \Phi(t,\sigma) \frac{\underline{u}_A(\sigma)}{\underline{u}_A(\sigma)^T \underline{u}_A(\sigma)} \dot{\tau}_d(\sigma) d\sigma \tag{4.23}$$

where, Φ denotes the state transition matrix corresponding to the differential equation in (4.22), namely,

$$\Phi(t,0) = \exp\left\{ -\int_0^t \frac{\underline{u}_A(\sigma) \dot{\underline{u}}_A(\sigma)^T}{\underline{u}_A(\sigma)^T \underline{u}_A(\sigma)} d\sigma \right\} \tag{4.24}$$

Because of (4.2), one can write $1 \leq \| \underline{u}_A \| \leq B_{u_A}$. For the first term in (4.23), following relations can be induced.

$$
\begin{aligned}
\|\Phi(t,0)\| &= \left\| \exp\left\{ -\int_0^t \frac{\underline{u}_A(\sigma)\dot{\underline{u}}_A(\sigma)^T}{\underline{u}_A(\sigma)^T \underline{u}_A(\sigma)} d\sigma \right\} \right\| \\
&= \left\| \exp\left\{ -\int_0^t \frac{\underline{u}_A(\sigma)}{\underline{u}_A(\sigma)^T \underline{u}_A(\sigma)} d\underline{u}_A(\sigma)^T \right\} \right\| \\
&\leq \left\| \exp\left\{ \left| -\int_0^t \frac{\underline{u}_A(\sigma)}{\underline{u}_A(\sigma)^T \underline{u}_A(\sigma)} d\underline{u}_A(\sigma)^T \right| \right\} \right\| \\
&\leq \left\| \exp\left\{ \int_0^t \frac{|\underline{u}_A(\sigma)|}{\underline{u}_A(\sigma)^T \underline{u}_A(\sigma)} |d\underline{u}_A(\sigma)^T| \right\} \right\| \\
&< \left\| \exp\left\{ \int_0^t |\underline{u}_A(\sigma)| |d\underline{u}_A(\sigma)^T| \right\} \right\| \\
&< \left\| \exp\left\{ B_{u_A} \int_0^t |d\underline{u}_A(\sigma)^T| \right\} \right\| \\
&= \left\| \exp\left\{ B_{u_A} \left(\underline{u}(t)^T - \underline{u}(0)^T \right) \right\} \right\| \\
&\leq B_1
\end{aligned}
\tag{4.25}
$$

where, B_1 is some positive constant. For the second term in (4.23), the analysis proceeds as given below.

$$\left\| \int_0^t \Phi(t,\sigma) \frac{\underline{u}_A(\sigma)}{\underline{u}_A(\sigma)^T \underline{u}_A(\sigma)} \dot{\tau}_d(\sigma) d\sigma \right\| < B_1 \left\| \int_0^t \frac{\underline{u}_A(\sigma)}{\underline{u}_A(\sigma)^T \underline{u}_A(\sigma)} \dot{\tau}_d(\sigma) d\sigma \right\|$$

$$< B_1 \left\| \int_0^t \underline{u}_A(\sigma) \dot{\tau}_d(\sigma) d\sigma \right\|$$

$$< B_1 B_{u_A} \left\| \int_0^t \dot{\tau}_d(\sigma) d\sigma \right\| \tag{4.26}$$

$$< B_1 B_{u_A} \| \tau_d(t) - \tau_d(0) \| \leq B_2$$

where, B_2 is some positive constant. Since the two components of the solution in (4.23) evolve boundedly, the sum of them will trivially be bounded as

$$\| \underline{\phi}(t) \| < B_1 + B_2 \tag{4.27}$$

□.

Note that in (4.6) the parameters of the flexible controller ($\underline{\phi}$) are required to be bounded. However, Theorem 4.2 states that once the system enters the sliding mode s_c=0, the boundedness of $\underline{\phi}$ is guaranteed. That is to say that (4.6) is automatically satisfied.

In the view of the analysis presented, the parameters of the controller are adjusted as

$$\dot{\phi}_1 = -\frac{e}{e^2 + \dot{e}^2 + 1} K \text{sgn}(s_c) \tag{4.28}$$

$$\dot{\phi}_2 = -\frac{\dot{e}}{e^2 + \dot{e}^2 + 1} K \text{sgn}(s_c) \tag{4.29}$$

$$\dot{\phi}_3 = -\frac{1}{e^2 + \dot{e}^2 + 1} K \text{sgn}(s_c) \tag{4.30}$$

The main problem in applying the design presented is the unavailability of the desired value of the control signal (τ_d). If this quantity is not available, one cannot construct s_c and the approach cannot be used for control purposes. In section 4.2, the relation between the s_p of (4.4) and s_c of (4.5) is analyzed.

4.2. Analysis of the Equivalence between Sliding Mode Control and Sliding Mode Learning

Consider the sliding line s_p and the zero-learning-error level s_c described by (4.4) and (4.5) respectively. The relation between these two quantities is assumed to be as

$$s_c = \Psi(s_p) \tag{4.31}$$

Qualitatively, if the value of s_p tends to zero, this means that s_c goes to zero. Theoretically, the system achieves perfect tracking because the controller produces the desired control inputs or vice versa. Conversely, as the value of s_p increases in magnitude, indicating that the error vector is getting away from the origin, the same sort of a divergent behavior in s_c is observed or vice versa. In this section, three conditions that Ψ must satisfy are discussed.

4.2.1. Region Condition

It should be clear that as the control input approaches the desired value for the current conditions, the state tracking error vector of the plant is driven towards the sliding manifold. In other words, the desired control signal drives the state tracking error to the sliding manifold. These two statements are clarified below.

$$\text{As } \tau \to \tau_d, s_p \to 0 \Leftrightarrow \text{As } s_p \to 0, \tau \to \tau_d \tag{4.32}$$

By utilizing s_p and s_c, the two equivalent statements and their consequences can be rewritten as

$$\lim_{s_c \to 0} s_p = 0 \Rightarrow \{\dot{e} \to -\lambda e \Rightarrow \begin{cases} e \to 0 \\ \dot{e} \to 0 \end{cases} \tag{4.33}$$

and

$$\lim_{s_p \to 0} s_c = 0 \Rightarrow \{\tau \to \tau_d \tag{4.34}$$

The two statements above require the following condition on Ψ.

$$\Psi(0) = 0 \tag{4.35}$$

Furthermore, as indicated in Figure 4.2, the relation Ψ must use the first and the third quadrants of the $\Psi(s_p)$ vs. s_p coordinate system.

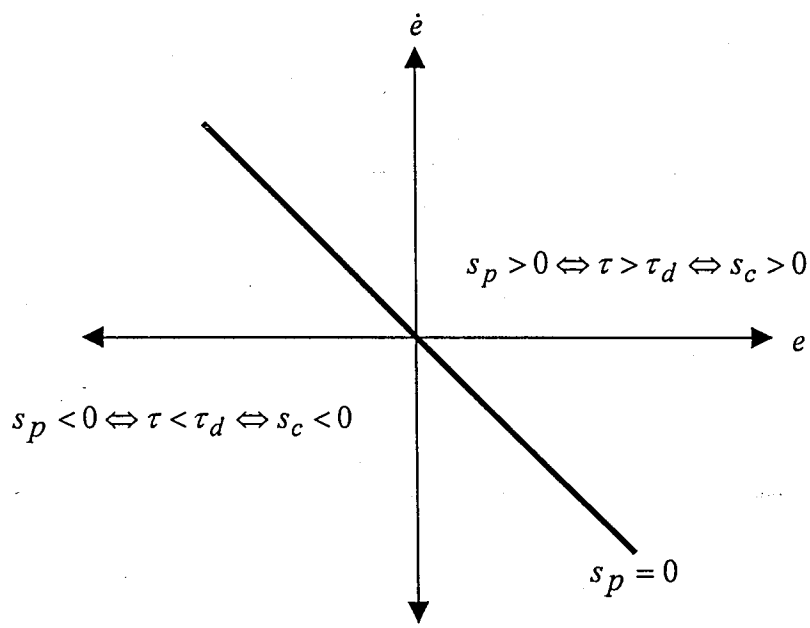

Figure 4.2. Signs of s_p and s_c on different sides of s_p=0 line

$$\Psi(s_p) = \begin{cases} \text{positive} & s_p > 0 \\ \text{zero} & s_p = 0 \\ \text{negative} & s_p < 0 \end{cases} \tag{4.36}$$

4.2.2. Compatibility Condition

In order to measure the tracking performance of the control system; define the Lyapunov function as

$$V_p = \frac{1}{2} s_p^2 \tag{4.37}$$

The measure of the realization performance of the controller (V_c) has already been defined in (4.12). In Figure 4.3, two sets are illustrated. If one selects a Ψ relation such that $\dot{V}_c < 0$ and $\dot{V}_p < 0$ are achieved simultaneously, then this selection can be considered as a suitable candidate. Since Ψ candidates from the regions other than the shaded set causes the violation of at least one of the design objectives, one has to find a Ψ relation from the intersection set.

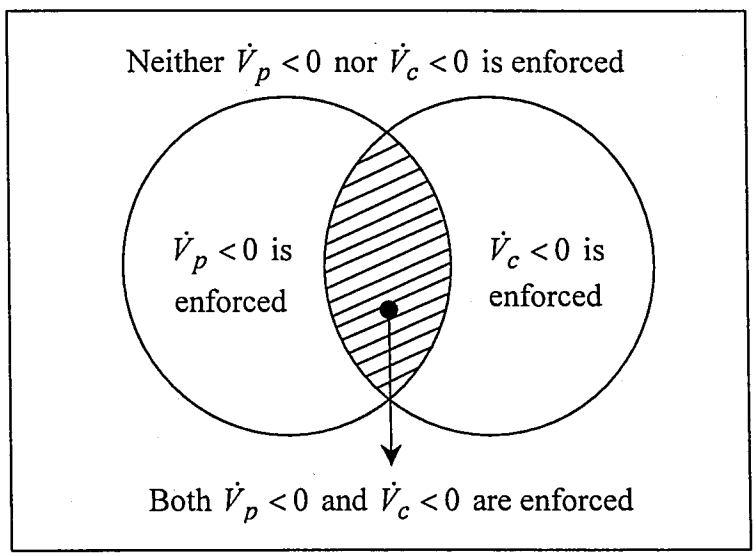

Figure 4.3. Sets of possible four cases

4.2.3. Invertibility Condition

As depicted in Figure 4.4, if the family of lines described by $s_p = \zeta$ ($\zeta > 0$) are drawn for varying values of ζ, the tracking error vector will fall into one of these subsets of the phase space at each instant of time. However, each one of the members of this family

corresponds to a different situation entailing different s_c values. Furthermore, as ζ increases in magnitude, the value of the relation Ψ must increase in magnitude, because of the increasing distance to the sliding line. Therefore the relation Ψ must be invertible. In other words, there must be a unique $s_p \in \Re$ for $\forall\ s_c \in \Re$.

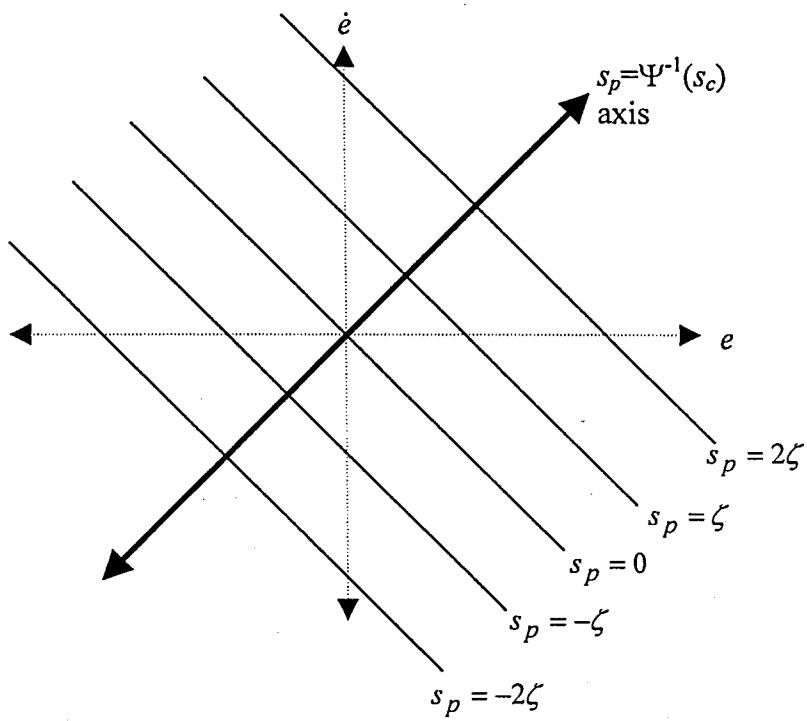

Figure 4.4. The family of lines formed as the value of s_p varies

These three conditions clearly stipulate that the Ψ relation must be such that the horizontal axes of the two subplots shown in Figure 4.5 must be mapped onto each other for achieving negative time derivatives for V_p and V_c.

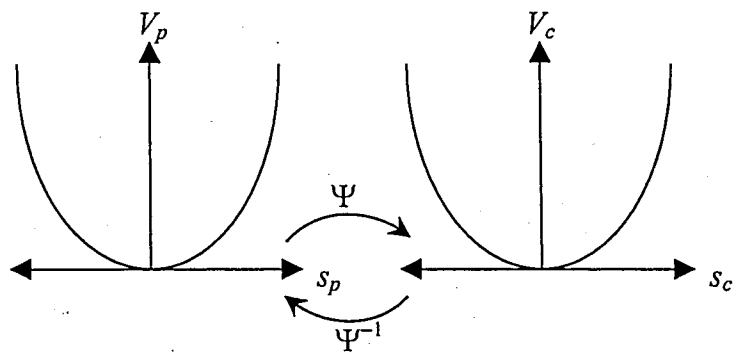

Figure 4.5. The relation Ψ performs a mapping between two horizontal axes shown

***Theorem* 4.3.** All monotonically increasing continuous functions passing through the origin satisfy the region, compatibility and invertibility conditions, and can serve as the Ψ relation for the establishment of an equivalency between the sliding mode control of the plant and the sliding mode learning inside the controller.

Proof: It must be noted that the partial derivative $\partial\Psi^{-1}(s_c)/\partial s_c$ is positive due to the monotonically increasing behavior of Ψ, and $\Psi^{-1}(s_c)\text{sgn}(s_c) \geq 0$ since Ψ is defined on the first and the third quadrants of s_c vs. $\Psi(s_c)$ coordinate system.

Stability in the Lyapunov sense requires the negative definiteness of the time derivative of the Lyapunov function in (4.37), whose argument can be rewritten by using the invertibility condition as

$$s_p = \Psi^{-1}(s_c) \tag{4.38}$$

Utilizing (4.38) leads to the time derivative in (4.39). In order to proceed comfortably, one should remember that the time derivative of s_c is defined in (4.19) and the parameter update rule is described in (4.9). Furthermore, the inequality $\underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d \leq B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d}$ holds always true as the right-hand side is composed of the bound values, which are strictly positive quantities.

$$\begin{aligned}
\dot{V}_p &= \dot{s}_p s_p \\
&= \left(\dot{\Psi}^{-1}(s_c)\right)\Psi^{-1}(s_c) \\
&= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\dot{s}_c \Psi^{-1}(s_c) \\
&= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(\underline{\dot{\phi}}^T \underline{u}_A + \underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d\right)\Psi^{-1}(s_c)
\end{aligned} \tag{4.39}$$

$$= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(-K\mathrm{sgn}(s_c) + \underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d\right)\Psi^{-1}(s_c)$$

$$= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(\Psi^{-1}(s_c)\left(\underline{\phi}^T \underline{\dot{u}}_A - \dot{\tau}_d\right) - K\left(\Psi^{-1}(s_c)\mathrm{sgn}(s_c)\right)\right)$$

$$\leq \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(\left|\Psi^{-1}(s_c)\right|\left(B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d}\right) - K\left(\Psi^{-1}(s_c)\mathrm{sgn}(s_c)\right)\right)$$

$$= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left(\left|\Psi^{-1}(s_c)\right|\left(B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d}\right) - K\left|\Psi^{-1}(s_c)\right|\right)$$

$$= \frac{\partial \Psi^{-1}(s_c)}{\partial s_c}\left|\Psi^{-1}(s_c)\right|\left(-K + B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d}\right)$$

Apparently from (4.39), choosing the bound parameter as given in (4.10) enforces the value of s_c to zero level, or equivalently, s_p to zero. It is straightforward to prove that a hitting occurs in finite time (See Proof 4.1). □

4.3. Simulation Results for the Computationally Intelligent Architectures

4.3.1. Dynamic Model of the Plant

In this thesis, the two degrees of freedom direct drive SCARA robotic manipulator, which is illustrated in Figure 4.6, is used as the test bed. Since the dynamics of such a mechatronic system is modeled by nonlinear and coupled differential equations, precise output tracking becomes a difficult objective due to the strong interdependency between the variables involved. Additionally, the ambiguities on the friction related dynamics in the plant model and the existence of noise on the measured quantities make the design much more complicated. Therefore the control methodology adopted must be capable of handling the difficulties stated.

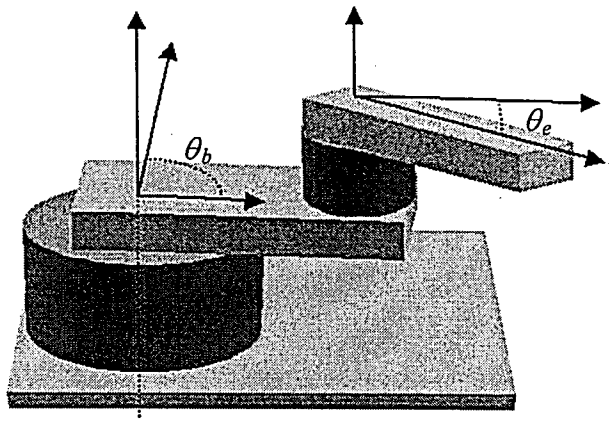

Figure 4.6. Physical structure of the manipulator

The general form of the robot dynamics is described as

$$M(\underline{\theta})\ddot{\underline{\theta}} + \underline{C}(\underline{\theta},\dot{\underline{\theta}}) = \underline{\tau} - \underline{f}_c \tag{4.40}$$

where, $M(\underline{\theta})$, $\underline{C}(\underline{\theta},\dot{\underline{\theta}})$, $\underline{\tau}$ and $\underline{f}_c$ stand for the state varying inertia matrix, the vector of Coriolis terms, the applied torque inputs and the Coulomb friction terms respectively. The plant parameters are given in Table 4.1 in standard m-kg-s units.

If the angular positions and angular velocities are described as the state variables of the system, four coupled and first order differential equations can define the model. In (4.41) and (4.42), the terms seen in (4.40) are given explicitly.

$$M(\underline{\theta}) = \begin{bmatrix} p_1 + 2p_3\cos(\theta_e) & p_2 + p_3\cos(\theta_e) \\ p_2 + p_3\cos(\theta_e) & p_2 \end{bmatrix} \tag{4.41}$$

$$V(\underline{\theta},\dot{\underline{\theta}}) = \begin{bmatrix} -\dot{\theta}_e(2\dot{\theta}_b + \dot{\theta}_e)p_3\sin(\theta_e) \\ \dot{\theta}_b^2 p_3\sin(\theta_e) \end{bmatrix} \tag{4.42}$$

In the above equations, p_1 = 3.31655+0.18648M_p, p_2 = 0.1168+0.0576M_p and p_3 = 0.16295+0.08616M_p. Here M_p denotes the payload mass. The details of the plant model can be found in Direct Drive Manipulator R&D Package User Guide [84].

Table 4.1. Manipulator parameters

Motor 1 Rotor Inertia	0.2670	Payload Mass (M_p)	2.0000
Arm 1 Inertia	0.3340	Arm 1 Length	0.3590
Motor 2 Rotor Inertia	0.0075	Arm 2 Length	0.2400
Motor 2 Stator Inertia	0.0400	Arm 1 CG Distance	0.1360
Arm 2 Inertia	0.0630	Arm 2 CG Distance	0.1020
Motor 1 Mass	73.000	Axis 1 Friction	4.9000
Arm 1 Mass	9.7800	Axis 2 Friction	1.6700
Motor 2 Mass	14.000	Torque Limit 1	245.00
Arm 2 Mass	4.4500	Torque Limit 2	39.200

4.3.2. Simulations for ADALINE Controller

In the simulations presented throughout the thesis, the reference trajectory depicted in Figure 4.7 is used. The links of the manipulator are named as base and elbow, the relevant variables of which are the vectors $\underline{\theta_1}$ and $\underline{\theta_2}$ respectively. In the results given in various figures, the parameters relevant to these links are indicated by 'b' or 'e' subscripts that follow the variable names. Initially, the links have been moved to $\theta_b=\pi/50$, and $\theta_e=-\pi/50$ radians respectively. Apart from these, what apply to all tests performed are the following.

First, the payload is varied during the simulations as illustrated in Figure 4.8. The payload variation graph implies that the motion starts with no payload. At time t=2 sec., a payload of 2 kg is grasped and released at time t=5 sec. The same variation is repeated at time t=9 sec. and t=12 sec. After the time t=15 sec., the manipulator is kept motionless.

The second common factor in the simulations is the noise sequence corrupting the observed state variables. Each state variable is corrupted by a different noise sequence, having zero mean and Gaussian distributed with variance approximately equal to $0.3333\text{x}10^{-6}$. The peak magnitude of the signal is within $\pm 10^{-3}$ with probability very close

to unity. The simulations have the stepsize 2.5msec, and the final time has been set to 20 sec.

The third parameter common in all results is the slope of the sliding line, which has been set to minus unity (λ=1).

Fourthly, the processing of the known bounds of the noise sequence is common in all results. Since the value of the bound denoted by n_b can be known in most applications, by appropriately utilizing this knowledge, the parameter adjustment activity excited solely by the noise sequence can be eliminated to some extent. For this purpose, the threshold function given as

$$T_{\underline{\phi}}(s_p)=\left(1+\exp\left(-10^5\left(|\Psi(s_p)|-n_b\right)\right)\right)^{-1} \tag{4.43}$$

is used in the parameter tuning mechanism and in the results discussed, n_b has been set to $2\text{x}10^{-3}$. Furthermore, in order to reduce the adverse effects of chattering phenomenon [70], the sign function in the parameter update rule has been replaced with a smooth function given as

$$\text{sgn}(\Psi(s_p))\approx\frac{\Psi(s_p)}{|\Psi(s_p)|+\delta} \tag{4.44}$$

where δ has been selected as 0.05. Together with these, the modified form of the update law can be described as

$$\dot{\underline{\phi}}=-\frac{\underline{u}_A}{\underline{u}_A^T\underline{u}_A}K\frac{\Psi(s_p)}{|\Psi(s_p)|+\delta}T_{\underline{\phi}}(s_p) \tag{4.45}$$

Lastly, the uncertainty bounds denoted by K_b and K_e have been set to 10^4 and 10^3 for the simulation results presented in this section, and Ψ relation has been selected as $\Psi(s_p)=s_p$.

The state tracking errors depicted in Figure 4.9 demonstrate that the prescribed sliding mode is achieved. Since the initial errors are nonzero, after a short reaching phase the components of the error vector come close to zero quickly, i. e. an accurate tracking is achieved with a simple controller architecture.

Figure 4.10 illustrates the applied torque signals, the sharp changes on which can be interpreted as follows. Since the controller has only three adjustable parameters, the lack of structural redundancy shows its effect as an increase in the task load on each adjustable parameter, and this necessitates a very fast tuning mechanism. Apparently, this makes the tuning strategy very sensitive to disturbances. A way to alleviate this drawback is to increase the number of adjustable parameters and to adopt a controller structure, which constructs the global behavior by combining the information from several subspaces of the phase space. The examples of such controller structures will be considered in the following sections.

In Figure 4.11, the phase plane behavior for both links is illustrated. In conjunction with the discussion above, the performance of the ADALINE controller during the reaching phase is good but it results in several number of hittings until the sliding mode takes place.

To understand the time behavior of the Lyapunov function defined in (4.37), one should refer to Figure 4.12. In order to make the transient behavior distinguishable, the horizontal axes of the subplots are selected as logarithmic. Parallel to the discussion on Figure 4.10 and Figure 4.11, during the early phases of the simulation, some fluctuations are observed on the Lyapunov function of (4.37) for both links.

Furthermore, the bounded parametric evolution claim of the derivation has been confirmed as depicted in Figure 4.13. One should notice here that the expressions in (4.28) and (4.29) have the error vector components in the numerator and the evaluated time derivative is therefore small in magnitude compared to that in (4.30), in which the corresponding term is equal to unity. This causes the parameter ϕ_3 to respond quicker than the first two parameters.

Briefly, the disturbance rejection ability is gained through the use of proposed learning law and accurate tracking is achieved with the structure used.

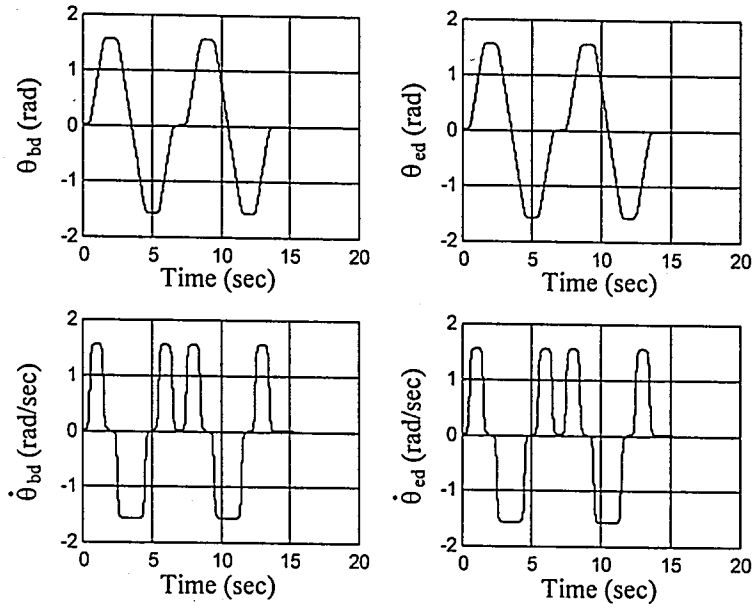

Figure 4.7. Reference state trajectories

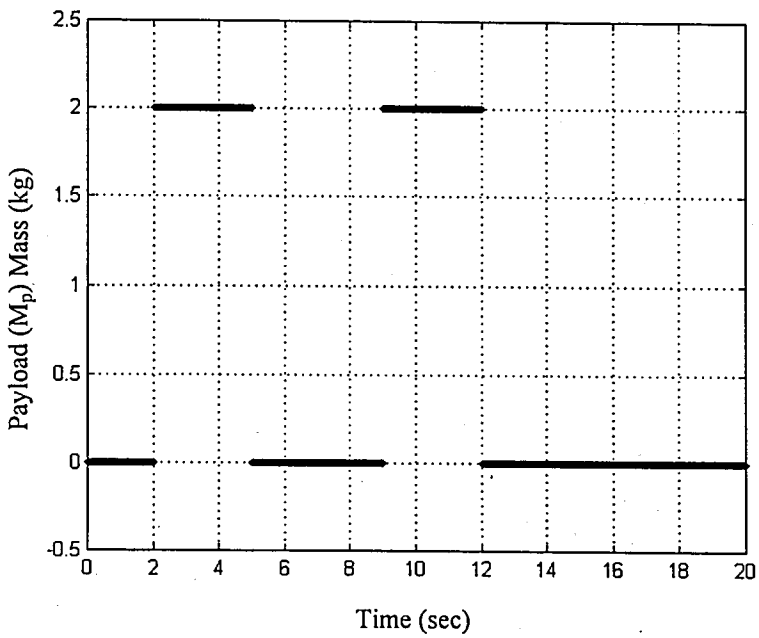

Figure 4.8. Time behavior of the payload mass

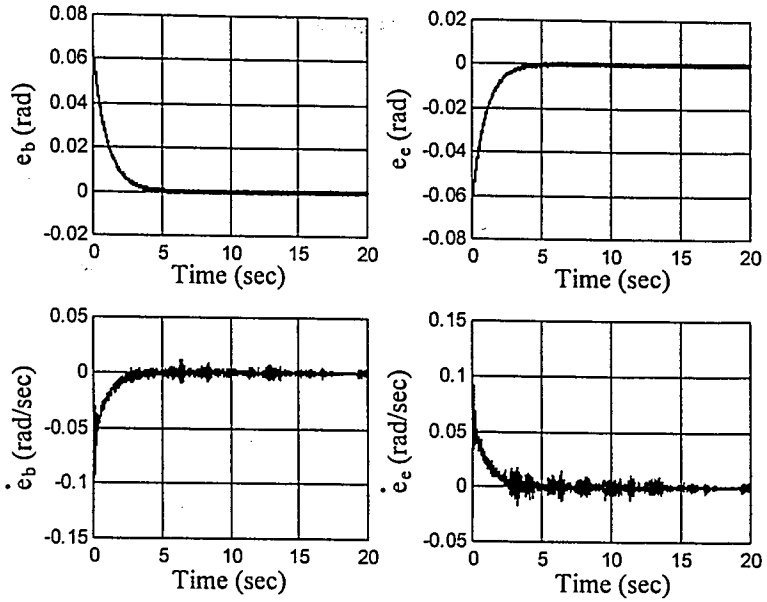

Figure 4.9. State tracking errors for ADALINE controller using (4.45)

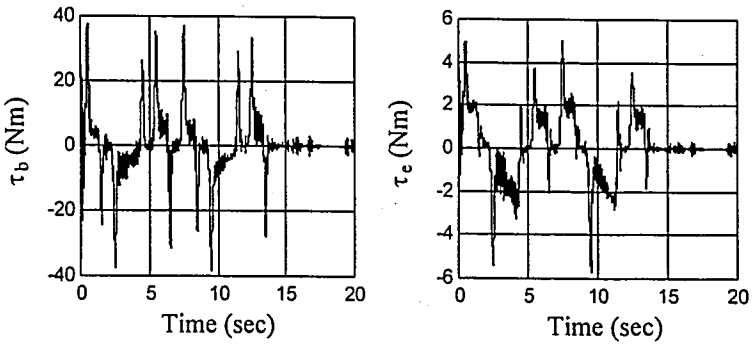

Figure 4.10. Applied torque signals for ADALINE controller using (4.45)

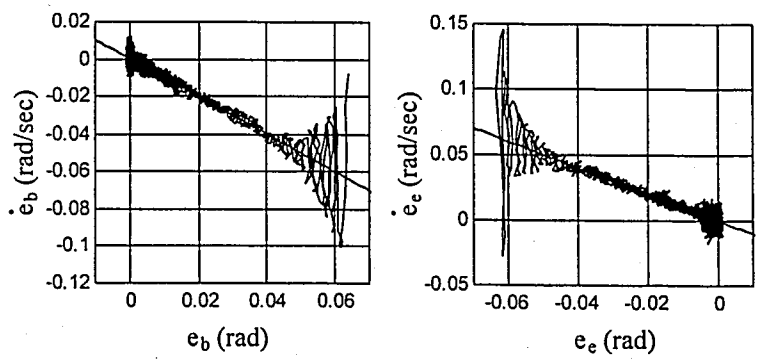

Figure 4.11. Trajectories in the phase plane for ADALINE controller using (4.45)

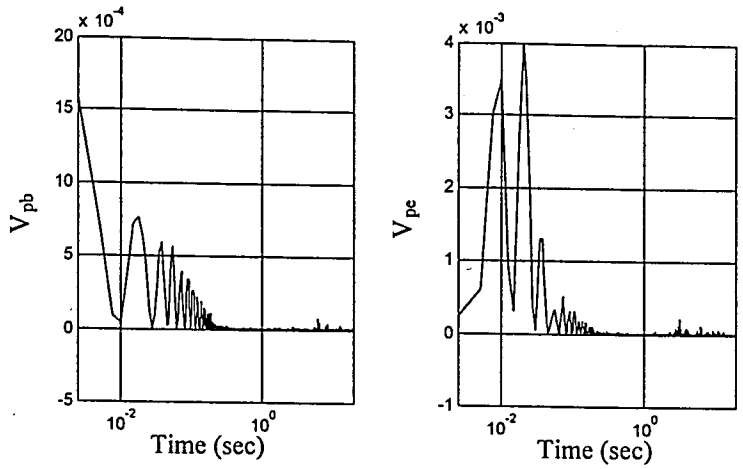

Figure 4.12. Time behavior of the Lyapunov functions for ADALINE controller using (4.45)

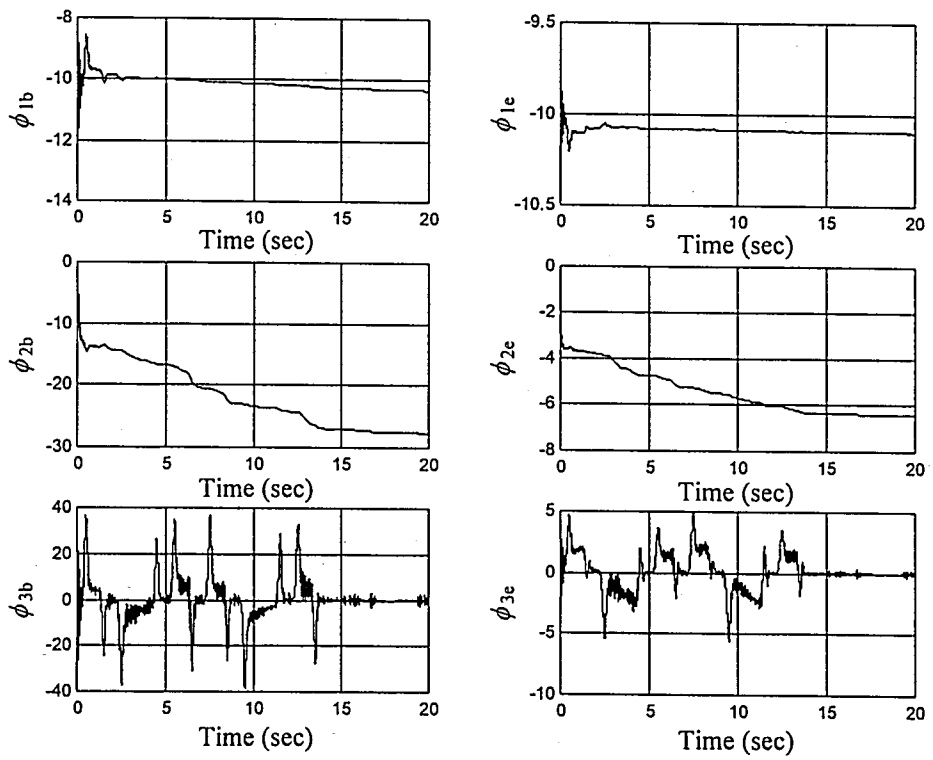

Figure 4.13. Evolution of the parameters of the ADALINE controllers using (4.45)

4.3.3. Formulation and Simulations for GRBFNN Controller

In using a GRBFNN structure as the controller, the c_{ij} and σ_{ij} parameters of the Gaussian functions defined in (2.8) are kept constant. The view of these functions over the relevant input domain is depicted in Figure 4.14. According to the adjustment law in (4.46), the entries of the $\underline{y}$ vector in (2.10) are adjusted as follows.

$$\dot{\underline{y}} = -\frac{\underline{w}}{\underline{w}^T \underline{w}} K \frac{\Psi(s_p)}{|\Psi(s_p)| + \delta} T_{\underline{\phi}}(s_p) \tag{4.46}$$

Initially, the entries of the $\underline{y}$ vector have been set to zero and 9 hidden neurons are used.

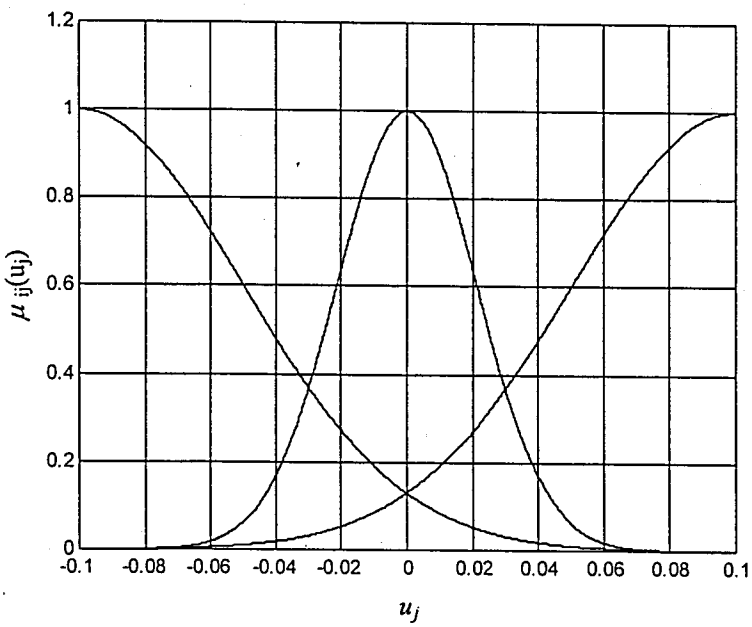

Figure 4.14. Appearance of the Gaussian functions covering the input space

The state tracking errors obtained with the GRBFNN controller are illustrated in Figure 4.15. Since the structure used has nine adjustable parameters being responsible for different subspaces of the phase space characterized by the radial basis functions, the tracking performance is slightly better than the ADALINE controller (See Figure 4.9).

A similar improvement is also observed in the applied torque signals, which are depicted in Figure 4.16. The increase in the structural redundancy introduces better handling of the local regions. When the applied elbow torque for ADALINE controller in Figure 4.10 and that for GRBFNN controller in Figure 4.16 is compared, a considerable improvement in the sense of torque smoothness is seen.

The behavior in the phase space depicted in Figure 4.17 clarifies the improvement introduced by the use of GRBFNN controller too. It must be noted that the activity during the reaching phase does not require as much hittings as in the case of ADALINE controller.

The time behavior of the Lyapunov functions for GRBFNN controller is illustrated in Figure 4.18, which confirm the results of the analysis.

The parameter evolution graphs for the base and the elbow link controllers are given in Figures 4.19 and 4.20 respectively. One can easily see that a parameter drift problem is of interest for the base link controller, but the parameters of the elbow link controller evolve reasonably.

The results obtained confirm the prominent features as claimed at the analysis level except the parameter drift problem.

A last remark on the use of GRBFNN controller must be on the boundaries of the operating region. Since the value of a Gaussian function decreases as the distance between the argument and the center increases, the coverage of the input space, i.e. the phase space in this case, deserves a careful consideration. Unless the designer ensures that the error vector lies within input universe of discourse illustrated in Figure 4.14 for all the time, the adjustable parameters of the structure will not be excited at all. Since it is difficult to ensure such a situation, the designer must either enlarge the input space reasonably or enforce other conditions, which, in fact, are the driving forces for using fuzzy controllers having open ended membership functions.

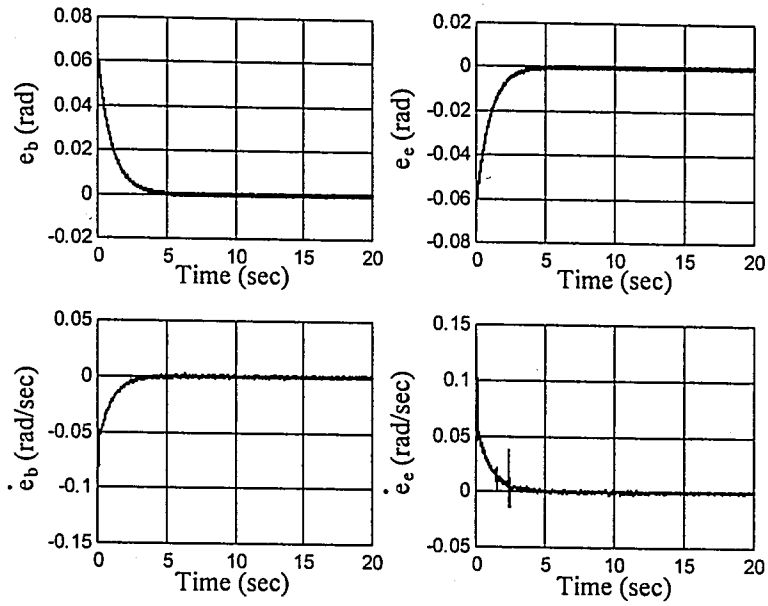

Figure 4.15. State tracking errors for GRBFNN controller using (4.46)

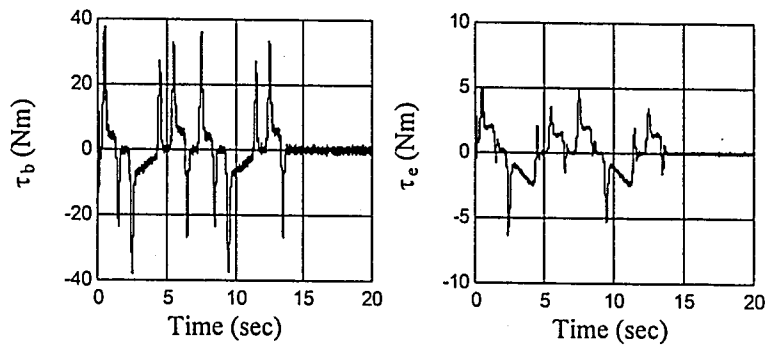

Figure 4.16. Applied torque signals for GRBFNN controller using (4.46)

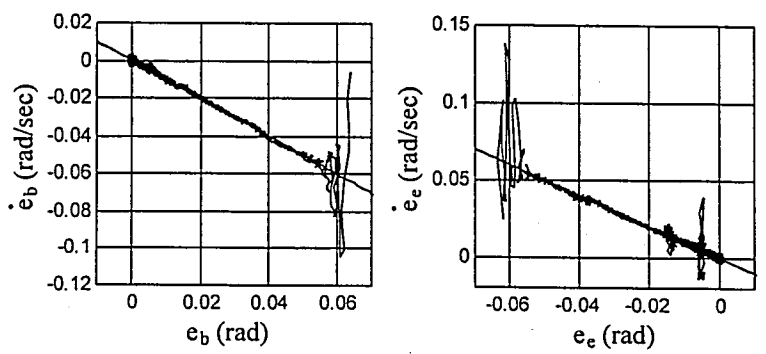

Figure 4.17. Trajectories in the phase plane for GRBFNN controller using (4.46)

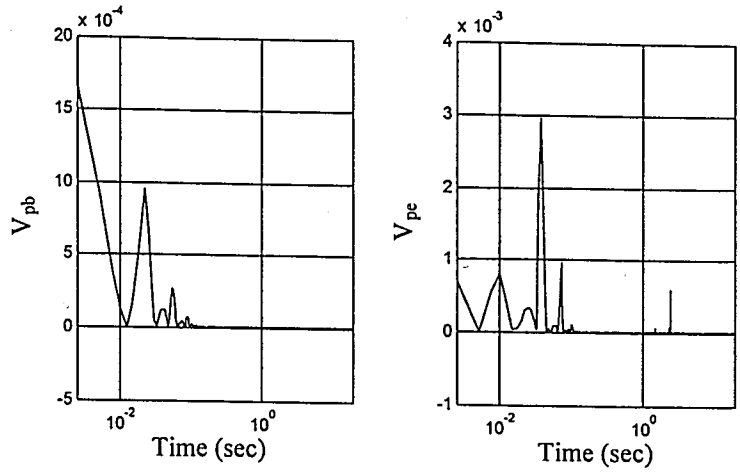

Figure 4.18. Time behavior of the Lyapunov functions for GRBFNN controller using (4.46)

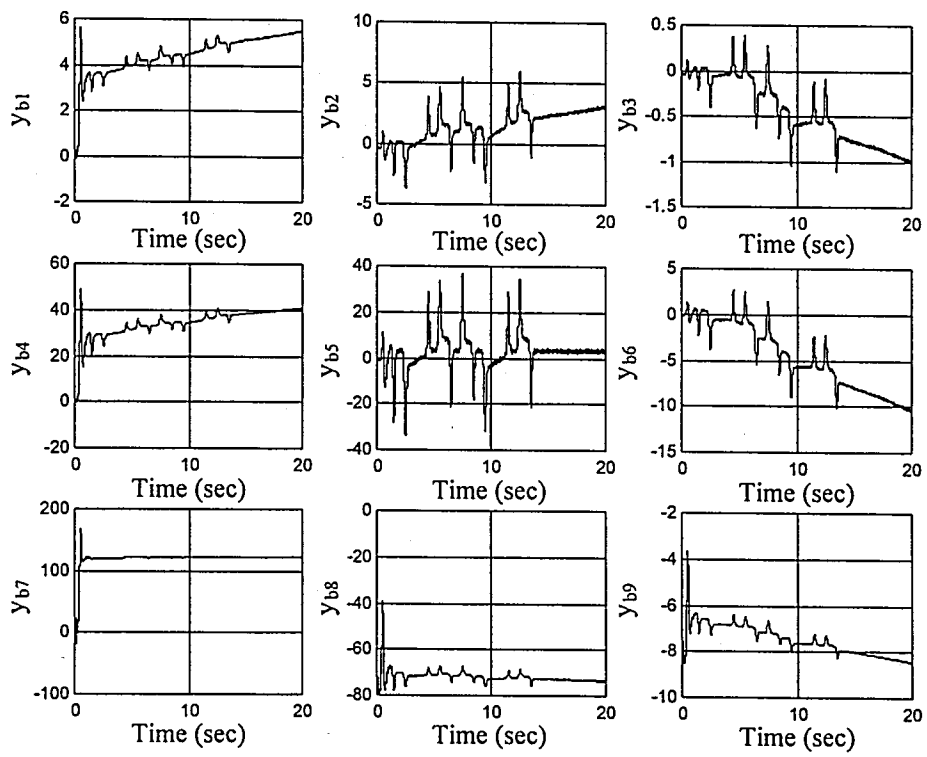

Figure 4.19. Evolution of the parameters of the base link GRBFNN controller using (4.46)

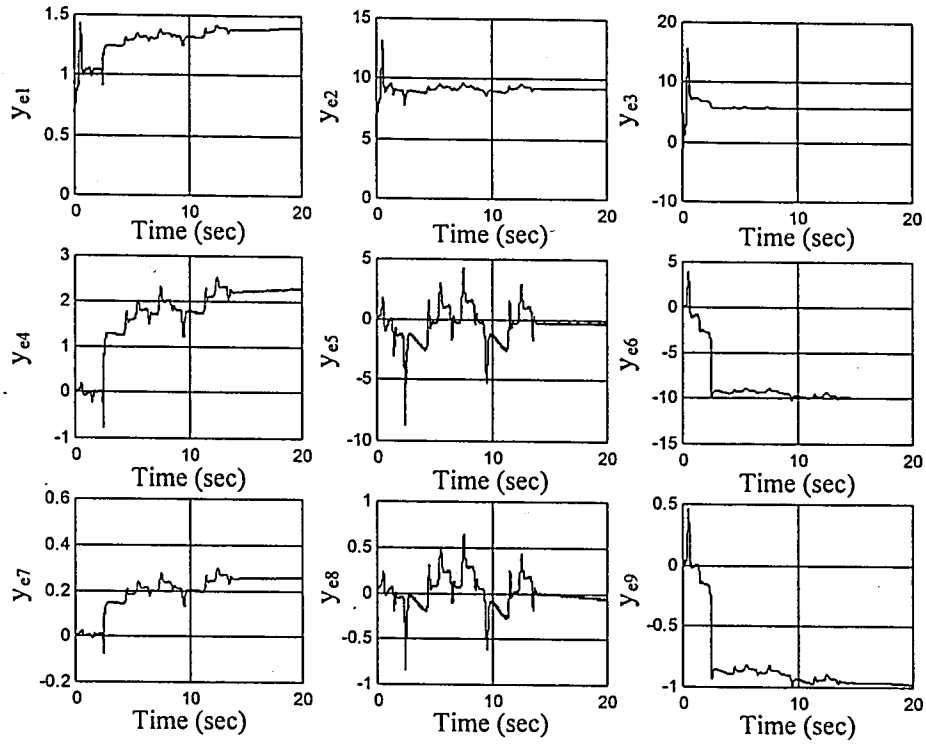

Figure 4.20. Evolution of the parameters of the elbow link GRBFNN controller using (4.46)

4.3.4. Formulation and Simulations for SFS Controller

The results presented in this part concern the use of a SFS structure depicted in Figure 2.7 as the controller. The c_{ij} and σ_{ij} parameters of the bell-shaped membership functions defined in (2.11) are not adapted but kept as constant. The coverage of the input universe of discourse is as depicted in Figure 4.21. According to the adjustment law in (4.45), the entries of the $\underline{y}$ vector in (2.14) are adjusted as follows.

$$\dot{\underline{y}} = -\frac{\underline{w}_n}{\underline{w}_n^T \underline{w}_n} K \frac{\Psi(s_p)}{|\Psi(s_p)| + \delta} T_{\underline{\phi}}(s_p) \tag{4.47}$$

Initially, the entries of the $\underline{y}$ vector have been set to zero. The results obtained for this structure are depicted in Figures 4.22 through 4.27.

The state tracking errors obtained using SFS controller is illustrated in Figure 4.22, from which it can clearly be seen that the behavior for all four entries is better than those of ADALINE and GRBFNN controllers, which are depicted in Figures 4.9 and 4.15 respectively.

The applied torque signals illustrated in Figure 4.23 do not differ very much from those produced by GRBFNN controllers but a significant smoothness comparably to the case of ADALINE type controller is apparent. From this point of view, the performance of the SFS controller is very good.

Similarly, the behavior in the phase space for both links is illustrated in Figure 4.24, from which it is clearly seen that the number of hittings during the reaching phase is not so excessive as in the case of ADALINE and GRBFNN controllers (See Figures 4.11 and 4.17). Therefore the performance during both the reaching mode and the sliding mode is very good.

In Figure 4.25, the time behavior of the Lyapunov function of (4.37) is depicted for both links. Unsurprisingly, the superior quality of the behavior observed in the phase plane is verified here. The fluctuations during the reaching phase are dampened out very quickly and a sliding mode takes place. During this mode the cost measures for both links are almost equal to zero, i.e. the tracking error components are in the vicinity of the origin.

Figures 4.26 and 4.27 illustrate the evolution of the parameters of the base and the elbow link controllers respectively. As in the case of GRBFNN controller, a parameter drift problem is still there and is because of the noise driven parameter tuning activity around the origin.

An overall assessment of the SFS controller should state the very good performance during the reaching and the sliding modes and the smoothness of the torque signals as the advantages. On the other hand, a slowly evolving parameter drift is the only problem associated with the structure.

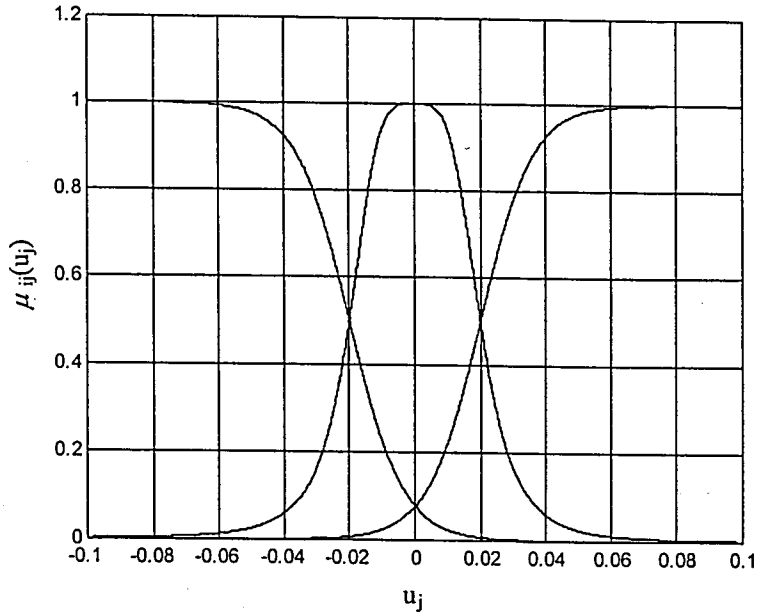

Figure 4.21. Appearance of the bell-shaped membership functions covering the input space

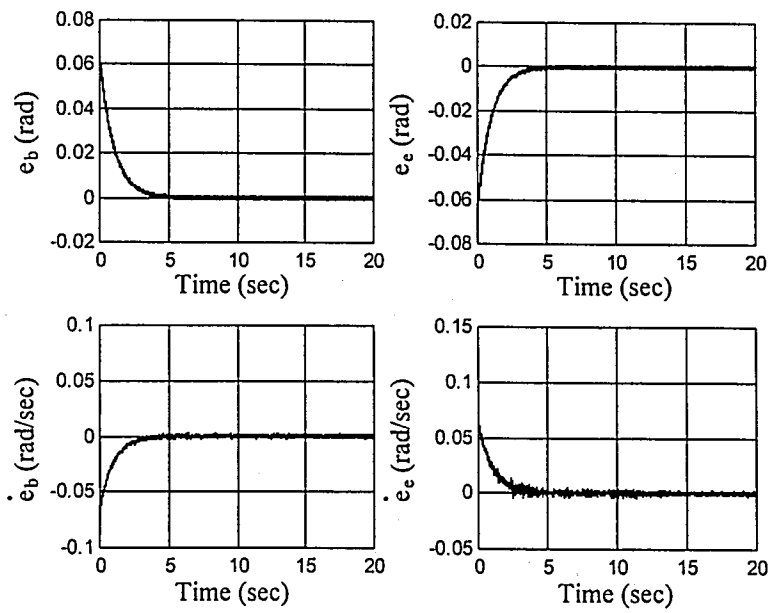

Figure 4.22. State tracking errors for SFS controller using (4.47)

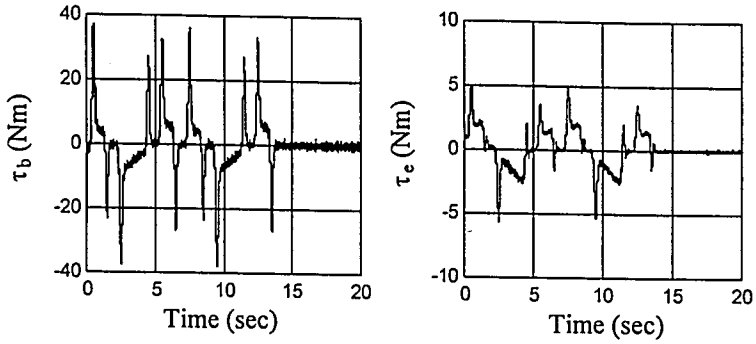

Figure 4.23. Applied torque signals for SFS controller using (4.47)

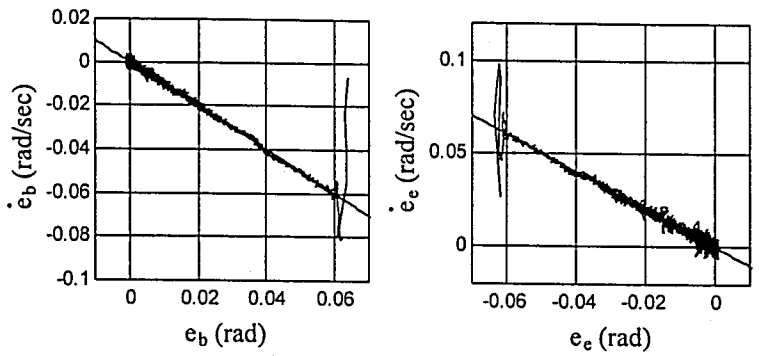

Figure 4.24. Trajectories in the phase plane for SFS controller using (4.47)

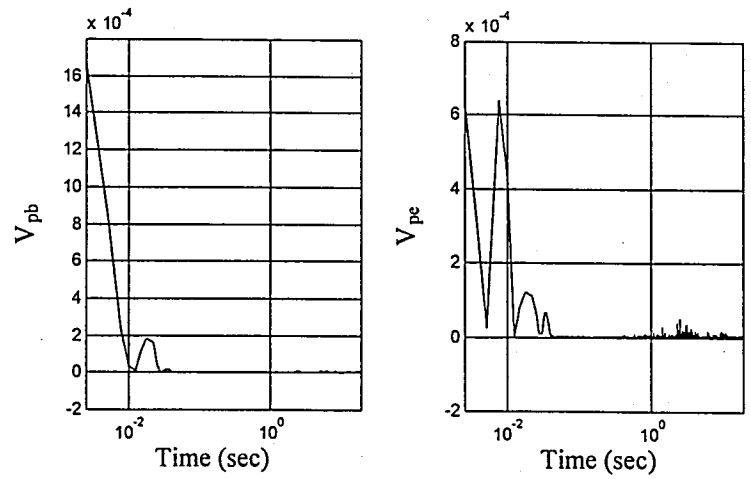

Figure 4.25. Time behavior of the Lyapunov functions for SFS controller using (4.47)

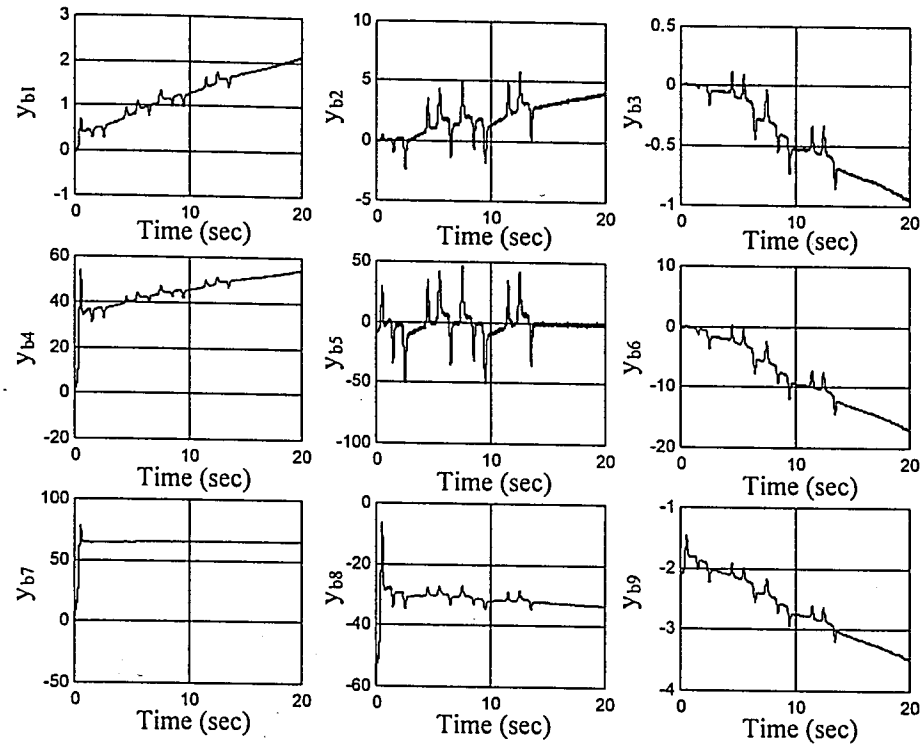

Figure 4.26. Evolution of the parameters of the base link SFS controller using (4.47)

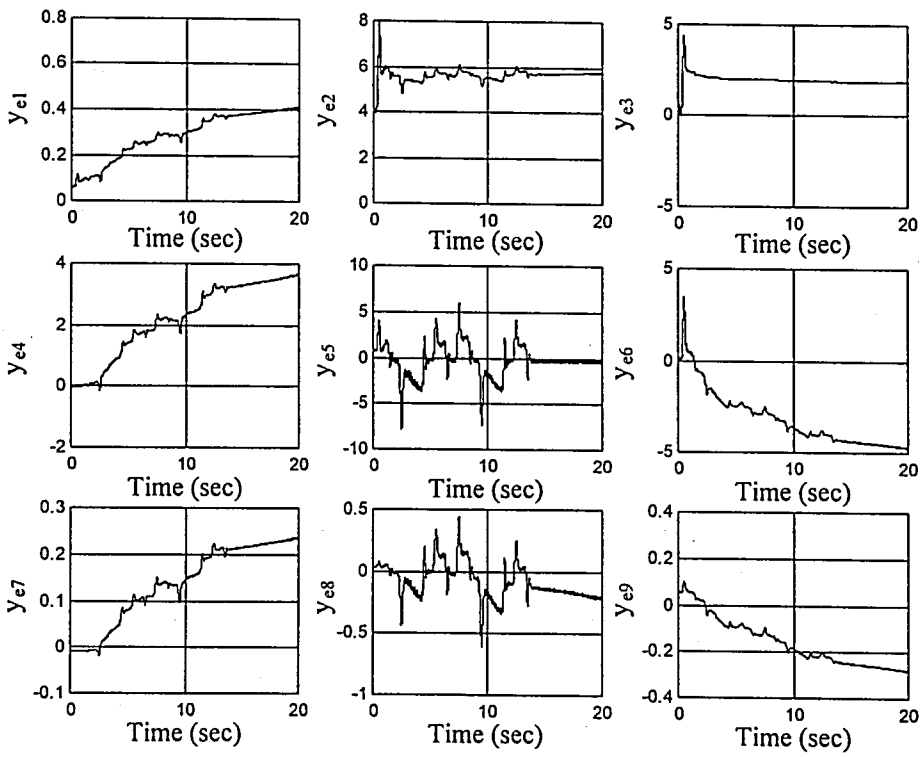

Figure 4.27. Evolution of the parameters of the elbow link SFS controller using (4.47)

4.3.5. Formulation and Simulations for ANFIS Controller

As in the previous part considering the use of SFS structure, the parameters of the bell-shaped membership functions for the ANFIS structure have been selected as the same constant values with zero initial values for the entries of the Y matrix of (2.15). The entries of the Y matrix in (2.15) are adjusted using

$$\dot{Y} = -\frac{\underline{w}_n \underline{u}_A^T}{\underline{w}_n^T \underline{w}_n \underline{u}_A^T \underline{u}_A} K \frac{\Psi(s_p)}{|\Psi(s_p)| + \delta} T_{\underline{\phi}}(s_p) \tag{4.48}$$

Since the approach entails 54 adjustable parameters (27 for each link), the time behavior graphs for the parameters are not given for the case of ANFIS strategy but the reader should keep the slowly evolving parameter drift problem in mind.

The results obtained are illustrated in Figures 4.28 through 4.31. The state tracking errors indicate the good tracking performance obtained in the face of strong external disturbances. However, it must be noted that the behavior obtained with the SFS controller is almost the same as that depicted in Figure 4.28.

Figures 4.29, 4.30 and 4.31 illustrate the applied torque signals, trajectories in the phase plane and the time behavior of the Lyapunov function of (4.37) for ANFIS controller. A fair comparison between these figures and those obtained with SFS controller leads to the fact that the two structures do not differ in terms of the performance measures discussed before. The designer must therefore pay attention to the cost of implementing the two structures. As mentioned above, the defuzzification stage of ANFIS controller possesses 27 adjustable parameters for each controller, while this number for the SFS controller is nine for the application discussed in the thesis. Therefore, the computational complexity now acts as the factor influencing the choice.

In brief, the structural redundancy of the ANFIS architecture having R rules in the rule base is more than that of a SFS architecture having the same number of rules in the rule base or a GRBFNN structure having R neurons in the hidden layer. This fact stipulates

that the ANFIS controller needs a structural optimization to reduce the unnecessary redundancy contained in the defuzzification stage.

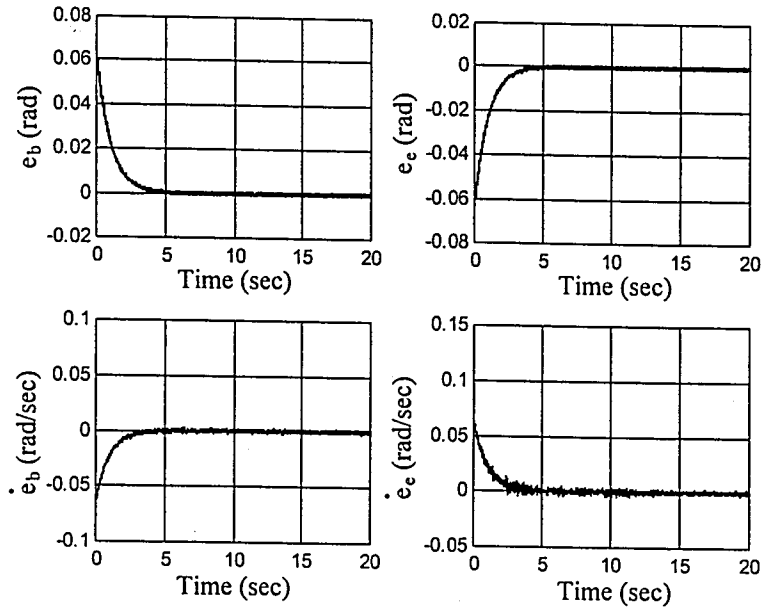

Figure 4.28. State tracking errors for ANFIS controller using (4.48)

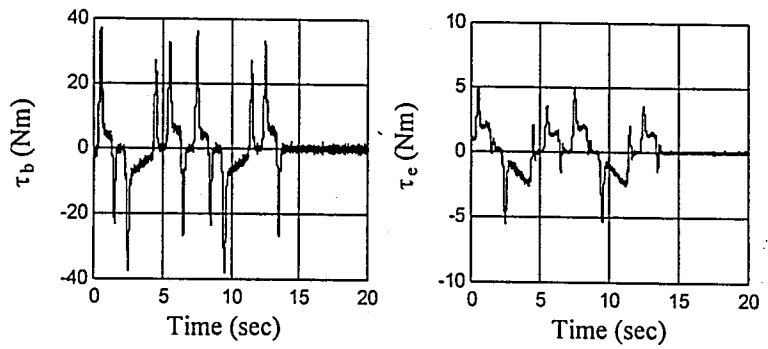

Figure 4.29. Applied torque signals for ANFIS controller using (4.48)

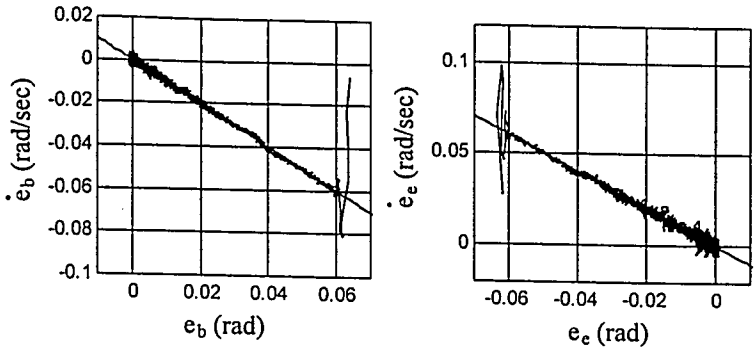

Figure 4.30. Trajectories in the phase plane for ANFIS controller using (4.48)

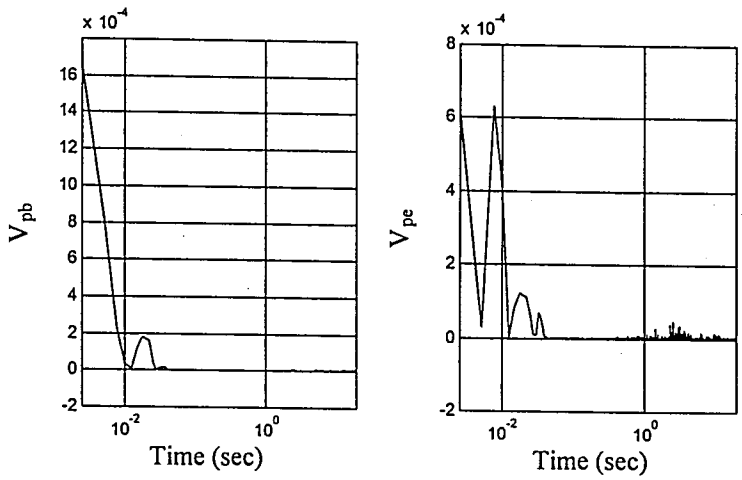

Figure 4.31. Time behavior of the Lyapunov functions for ANFIS controller using (4.48)

4.4. A Discussion on the Results

The simulation studies presented in this section consider the application of a dynamic parameter tuning strategy to computationally intelligent architectures. The method discussed has originally been proposed for applications in which the target output of the computationally intelligent structure is available [26-27]. However, in control engineering applications, unavailability of the target control signal makes it impossible to implement such algorithms directly. For this purpose, the relations between the sliding surface s_p and the zero learning-error level of s_c are analyzed, and it is demonstrated that a suitable relation between them can make it possible to use the algorithm for control applications [85].

The application of the algorithm for FNN structure deserves a careful interpretation. If an open form of Figure 2.2 is redrawn as illustrated in Figure 4.32, it can clearly be seen that the quantities entering into the logistic functions are realizing the necessary controller outputs, i.e. the torque values, which are identical to each other in the average sense. More explicitly, each one of the linear combiners in the hidden layer corresponds to an ADALINE structure, whose output tends to the necessary torque value as imposed by the learning strategy. However, the hidden layer performs a nonlinear transformation, which is because of the logistic function, and a final linear combination takes place for the evaluation of the output of the FNN controller.

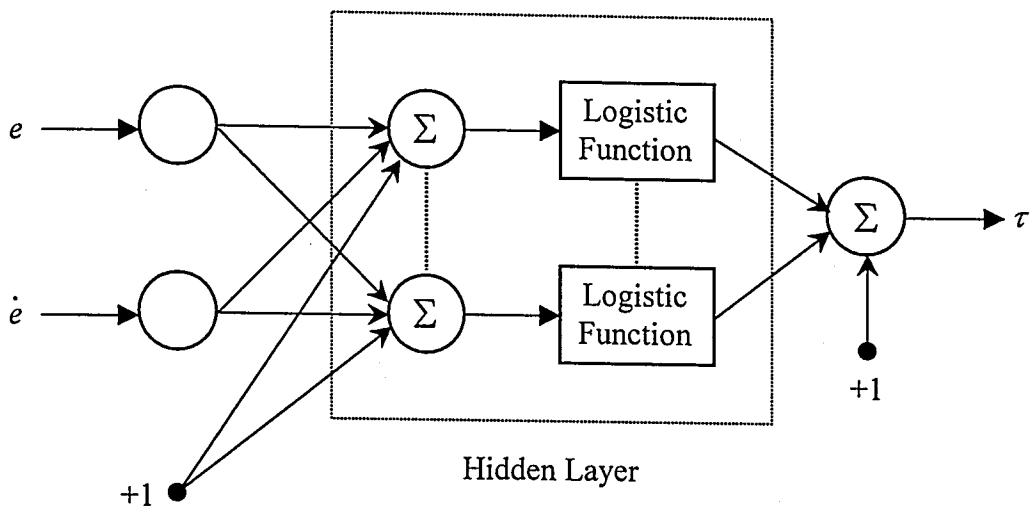

Figure 4.32. An open form of the FNN structure for an architectural interpretation of the parameter tuning algorithm

The problem in applying the learning strategy for FNN structure stems from the fact that at an intermediate stage, the approximate values of the necessary controller output is produced. But the hidden layer transforms this value to some other value, using which the structure is expected to produce necessary torque values, which are already available at the output of the linear combiners. Therefore, with the proposed learning strategy, the structural redundancy of FNN architecture causes a meaningless *architecture-learning strategy* pair.

Furthermore, since the logistic functions introduce nonlinearity, the linearity requirement of Theorem 4.1 on the adjustable parameters is violated and one cannot

guarantee that the structure produces the necessary torque values at the output node. Therefore, in the simulation studies presented in this section, FNN structure has not been implemented.

What should be emphasized as the last remarks on the results are the good tracking performance and the robustness against the disturbances. In the view of what have been observed, the approach is a good candidate for tracking control of nonlinear systems, whose responses are robustified by an intelligent controller.

5. PARAMETER TUNING BASED ON A TWO-TERM LYAPUNOV FUNCTION

In this section, an analysis of the sliding mode creation problem, which is based on a two-term Lyapunov function, is given. The approach presented is elaborated specifically for second order systems having a single input; nevertheless, the extension to higher order systems is possible. The proposed form of the update dynamics constructs the time derivative of the parameter vector, the use of which results in the observation of a sliding dynamics on the phase plane. For the purpose of tuning the controller parameters, the analysis presented in the preceding section, which considers the relation between the sliding line defined for the plant under control (s_p) and the zero learning-error level for the controller (s_c), is utilized.

5.1. Definitions and the Formulation of the Problem Using ADALINE Structure

As discussed in the section 4.1, the parameter vector, the vector of augmented inputs and the input-output relation of a three input one output ADALINE structure are given in (4.1), (4.2) and (4.3) respectively. Defining the sliding surface of the plant and the zero learning-error level of the controller as in (4.4) and (4.5), one can construct the following cost function.

$$J = \frac{1}{2}(\tau - \tau_d)^2 = \frac{1}{2}s_c^2 \tag{5.1}$$

Based on the cost measure in (5.1), an augmented switching manifold can be designed as

$$\underline{s}_A = \begin{bmatrix} s_c \\ \dfrac{\partial J}{\partial \underline{\phi}} \end{bmatrix} \tag{5.2}$$

The interpretation of the selection in (5.2) is of special importance because of the following facts. If a parameter adjustment mechanism results in a motion taking place in a

subspace characterized by $\underline{s}_A = \underline{0}$, the controller achieves the zero learning-error level due to the enforcement of $s_c = 0$. However, the contribution of the second component is not so trivial and is related to the gradient descent technique [61]. In the conventional gradient descent technique, the prescribed form of the parameter vector is

$$\dot{\underline{\phi}} = -\gamma \frac{\partial J}{\partial \underline{\phi}} \tag{5.3}$$

in which the parameter γ is called the learning rate and is chosen from the interval (0,1).

Apparently, the rule in (5.3) uses the gradient of the cost in (5.1) and extracts the direction information, the optimality of which is questionable. As is clearly seen in (5.3), the second term of (5.2) determines the rate of change in the parameter vector $\underline{\phi}$. The rationale behind the addition of the second term now becomes clearer. If the switching manifold is augmented with this term, the motion on $\underline{s}_A = 0$ subspace will still entail $s_c = 0$, which is the goal of the design. The importance of $\partial\tau/\partial\underline{\phi}$ term is seen if the rule in (5.3) is rewritten as

$$\dot{\underline{\phi}} = -\gamma\, s_c \frac{\partial s_c}{\partial \underline{\phi}} = -\gamma\, s_c \frac{\partial \tau}{\partial \underline{\phi}} \tag{5.4}$$

The information contained in $\partial\tau/\partial\underline{\phi}$ term is a measure of a parameter's effect on the output, i.e. the parameters having high values in the corresponding entries of the vector $\partial\tau/\partial\underline{\phi}$ are dominantly determining the output of the controller and deserve adjustment priority.

In view of this, the motion in the phase plane can be organized such that a shorter path to $\underline{s}_A = 0$ subspace could be extracted by an appropriate selection of the parameter adjustment law alternative to that in (5.3). For this purpose, consider the Lyapunov function candidate as

$$V_c = \frac{1}{2} \underline{s}_A^T P \underline{s}_A \tag{5.5}$$

where, P is defined as follows.

$$P = \begin{bmatrix} \mu & 0_{1xL} \\ 0_{Lx1} & \rho I_{LxL} \end{bmatrix} \tag{5.6}$$

where, L is the number of adjustable parameters and, μ and ρ are positive constants. Based on the selection in (5.6), the open form of the Lyapunov function in (5.5) can be written as

$$V_c = \mu J + \rho \frac{1}{2} \left\| \frac{\partial J}{\partial \underline{\phi}} \right\|^2 \tag{5.7}$$

in which, the selection of the weight parameters μ and ρ must be done by comparing the magnitudes of the time-varying two terms of (5.7). The definition of the norm in (5.7) is as in (4.6). Aside from the bound conditions given in (4.6) through (4.8), the following assumptions are needed for the derivation.

$$\|\tau\| \le B_\tau \tag{5.8}$$

$$\|\tau_d\| \le B_{\tau_d} \tag{5.9}$$

$$\|\underline{u}_A\| \le B_{u_A} \tag{5.10}$$

***Theorem* 5.1.** For a controller structure, in which the output is a linear function of the adjustable parameters, the adaptation of the controller parameters as described in (5.11) ensures the negative definiteness of the time derivative of the Lyapunov function candidate in (5.5).

$$\dot{\underline{\phi}} = -K \left(\mu I + \rho \frac{\partial^2 J}{\partial \underline{\phi}\, \partial \underline{\phi}^T} \right)^{-1} \operatorname{sgn}\left(\frac{\partial J}{\partial \underline{\phi}} \right) \tag{5.11}$$

where, K is a sufficiently large constant satisfying

$$K > \left(\mu B_{\phi} + \rho B_{u_A}\right) B_{\dot{u}_A} \tag{5.12}$$

Proof: Evaluating the time derivative of the Lyapunov function in (5.7) yields the following.

$$\dot{V}_c = \mu\left(\frac{\partial J}{\partial \underline{\phi}}\right)^T \dot{\underline{\phi}} + \rho\left(\frac{\partial J}{\partial \underline{\phi}}\right)^T \frac{\partial^2 J}{\partial \underline{\phi}\, \partial \underline{\phi}^T} \dot{\underline{\phi}} + \mu\left(\frac{\partial J}{\partial \underline{u}_A}\right)^T \dot{\underline{u}}_A + \rho\left(\frac{\partial J}{\partial \underline{\phi}}\right)^T \frac{\partial^2 J}{\partial \underline{\phi}\, \partial \underline{u}_A^T} \dot{\underline{u}}_A$$

$$= \left(\frac{\partial J}{\partial \underline{\phi}}\right)^T \left(\mu I + \rho \frac{\partial^2 J}{\partial \underline{\phi}\, \partial \underline{\phi}^T}\right) \dot{\underline{\phi}} + \mu\left(\frac{\partial J}{\partial \underline{u}_A}\right)^T \dot{\underline{u}}_A + \rho\left(\frac{\partial J}{\partial \underline{\phi}}\right)^T \frac{\partial^2 J}{\partial \underline{\phi}\, \partial \underline{u}_A^T} \dot{\underline{u}}_A \tag{5.13}$$

Since $\tau = \underline{\phi}^T \underline{u}_A$, following terms can be calculated by using (5.1).

$$\left(\frac{\partial J}{\partial \underline{\phi}}\right)^T = s_c \underline{u}_A^T \tag{5.14}$$

$$\left(\frac{\partial J}{\partial \underline{u}_A}\right)^T = s_c \underline{\phi}^T \tag{5.15}$$

$$\frac{\partial^2 J}{\partial \underline{\phi}\, \partial \underline{\phi}^T} = \underline{u}_A \underline{u}_A^T \tag{5.16}$$

$$\frac{\partial^2 J}{\partial \underline{\phi}\, \partial \underline{u}_A^T} = I \tag{5.17}$$

The time derivative in (5.13) can be rearranged as

$$\dot{V}_c = s_c \underline{u}_A^T \left(\mu I + \rho \underline{u}_A \underline{u}_A^T\right) \dot{\underline{\phi}} + \mu s_c \underline{\phi}^T \dot{\underline{u}}_A + \rho s_c \underline{u}_A^T \dot{\underline{u}}_A$$

$$\leq s_c \underline{u}_A^T \left(\mu I + \rho \underline{u}_A \underline{u}_A^T\right) \dot{\underline{\phi}} + \left(\mu B_\phi + \rho B_{u_A}\right) B_{\dot{u}_A} |s_c| \tag{5.18}$$

If the parameter adjustment rule in (5.11) is substituted into the last inequality of (5.18), the expression below is obtained.

$$\dot{V}_c \leq -K s_c \underline{u}_A^T \operatorname{sgn}(s_c \underline{u}_A) + \left(\mu B_\phi + \rho B_{u_A}\right) B_{\dot{u}_A} |s_c|$$

$$= -K |s_c| \underline{u}_A^T \operatorname{sgn}(\underline{u}_A) + \left(\mu B_\phi + \rho B_{u_A}\right) B_{\dot{u}_A} |s_c|$$

$$= -|s_c| \left(K \underline{u}_A^T \operatorname{sgn}(\underline{u}_A) - \left(\mu B_\phi + \rho B_{u_A}\right) B_{\dot{u}_A}\right) \tag{5.19}$$

$$\leq -|s_c| \left(K - \left(\mu B_\phi + \rho B_{u_A}\right) B_{\dot{u}_A}\right)$$

The last inequality in (5.19) follows from the expression given as

$$\underline{u}_A^T \operatorname{sgn}(\underline{u}_A) = u_{m+1} \operatorname{sgn}(u_{m+1}) + \sum_{j=1}^{m} u_j \operatorname{sgn}(u_j)$$

$$= 1 + \sum_{j=1}^{m} u_j \operatorname{sgn}(u_j) \tag{5.20}$$

$$= 1 + \sum_{j=1}^{m} |u_j| \geq 1$$

The selection of the parameter K as in (5.12) ensures the negative definiteness of the time derivative of the Lyapunov function in (5.5) and proves Theorem 5.1 □.

Estimating an upper value for the hitting time (t_h) is not as a mathematically tractable problem as that in the case of single-term Lyapunov function based approach. For this purpose, following differential equation can be written.

$$\dot{\underline{s}}_A = \frac{d}{dt}\begin{bmatrix} s_c \\ \dfrac{\partial J}{\partial \underline{\phi}} \end{bmatrix} = \dot{s}_c \begin{bmatrix} 1 \\ \underline{u}_A \end{bmatrix} + s_c \begin{bmatrix} 0 \\ \dot{\underline{u}}_A \end{bmatrix} \tag{5.21}$$

On the switching manifold s_c is equal to zero, i.e. the second term in (5.21) disappears. Therefore, one has to analyze dynamics introduced by $\dot{s}_c$, the solution to which at time t_h is given as

$$\frac{|s_c(0)|}{K} = \int_0^{t_h} \left(\operatorname{sgn}\left(\underline{u}_A^T\right)\left(\mu I + \rho \underline{u}_A \underline{u}_A^T\right)^{-1^T} u_A + \underline{\phi}^T \dot{\underline{u}}_A - \dot{\tau}_d \right) d\sigma \tag{5.22}$$

Since the available knowledge on the time derivative of the input vector and the desired output are limited to the bounds, it is difficult to give an analytic estimate for the hitting time. Therefore, the question of whether there exists a positive t_h value satisfying the above equation is an open question.

5.2. Simulation Results for the Computationally Intelligent Architectures

5.2.1. Simulations for ADALINE Controller

In the simulations presented, the dynamic model of the plant introduced in the section 4.3.1 has been used. The reference trajectory used is as depicted in Figure 4.7 and the time behavior of the payload mass is as illustrated in Figure 4.8.

The initial conditions of the robot, the noise sequence, the simulation stepsize, the simulation endtime, the sliding line parameter, the noise bound parameter, and the approximated sign function given in (4.44) are the same as those discussed in section 4.3.2. The parameters μ and ρ are both selected as unity for all simulations presented in this

section. Furthermore, the uncertainty bounds K_b and K_e have been set to 10^4 and 10^3 respectively for ADALINE controller.

The form of the update law used in the simulations for ADALINE controller is described as

$$\dot{\underline{\phi}} = -K\left(\mu I + \rho \underline{u}_A \underline{u}_A^T\right)^{-1} \underline{S}_a T_{\underline{\phi}}(s_p) \tag{5.23}$$

in which, the i^{th} entry of the term $\underline{S}_a$, which is an $(m+1)\times 1$ vector, and the scalar term $T_{\underline{\phi}}(s_p)$ are defined in (5.24) and (4.43) respectively.

$$\underline{S}_{a_i} = \frac{\Psi(s_p)\underline{u}_{A_i}}{\left|\Psi(s_p)\underline{u}_{A_i}\right| + \delta} \tag{5.24}$$

In Figure 5.1, the state tracking errors are illustrated. The nonzero initial errors quickly converge to zero by the applied torque signal depicted in Figure 5.2. If the results that obtained with ADALINE controller using the single-term Lyapunov function based tuning strategy are compared with those given in Figures 5.1 through 5.4, one cannot see any major difference. However, the method discussed in this section extends the cost measure with the sensitivity terms and the magnitude of fluctuations in the parameters gets relatively small.

The illustrated results stipulate that the proposed form of the update law is an appropriate alternative to the conventional training strategies.

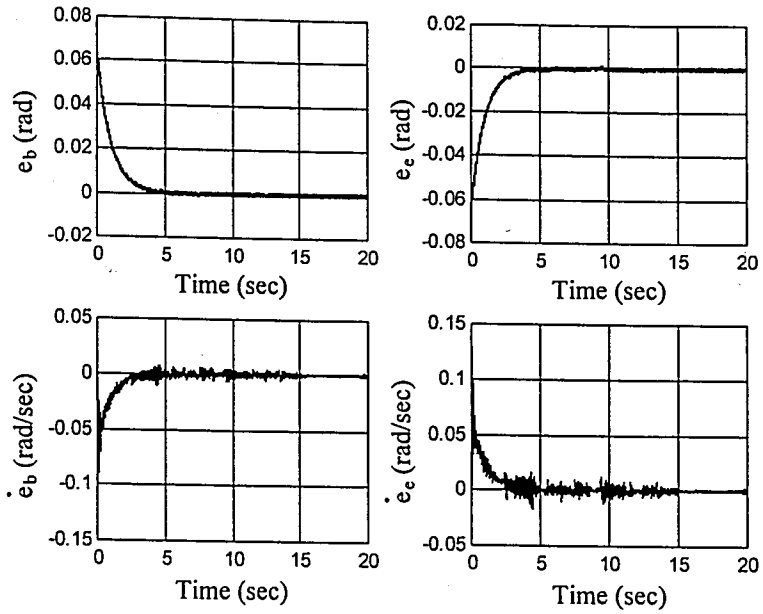

Figure 5.1. State tracking errors for ADALINE controller using (5.23)

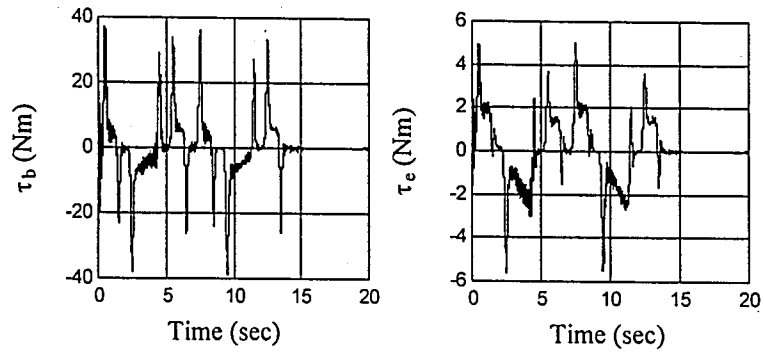

Figure 5.2. Applied torque signals for ADALINE controller using (5.23)

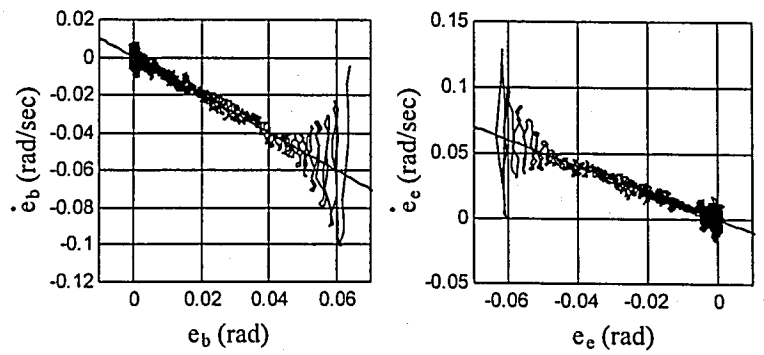

Figure 5.3. Trajectories in the phase plane for ADALINE controller using (5.23)

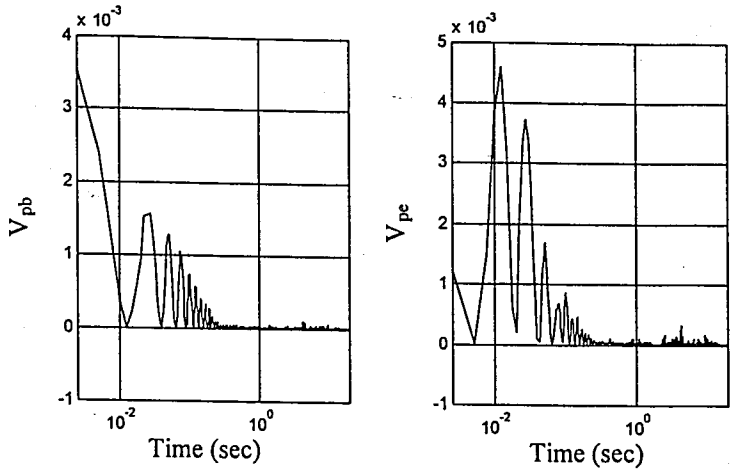

Figure 5.4. Time behavior of the Lyapunov functions for ADALINE controller using (5.23)

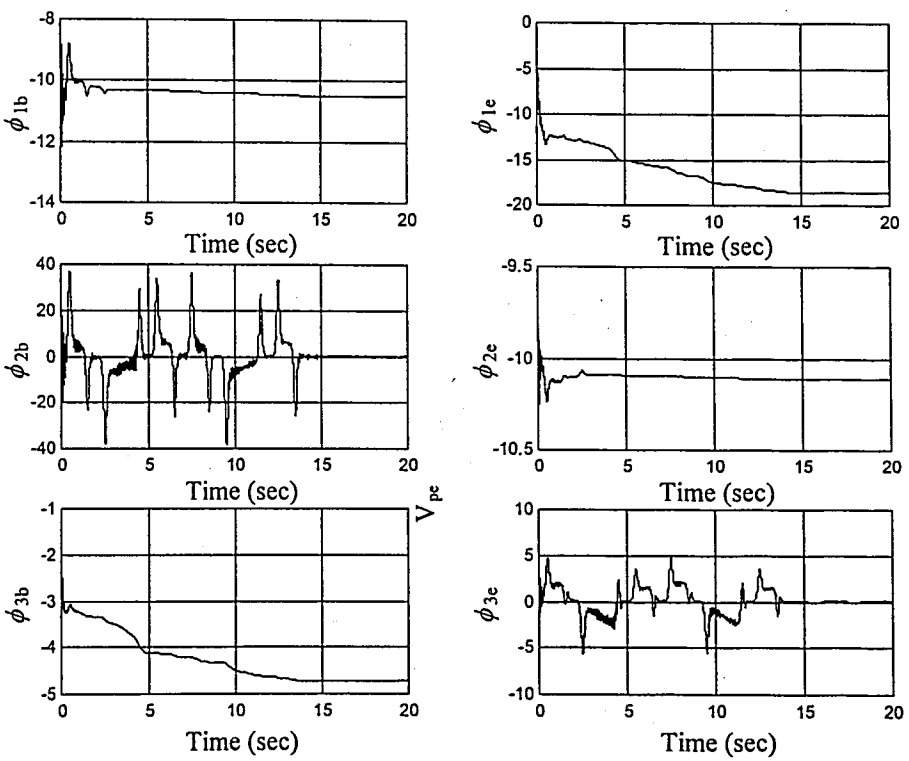

Figure 5.5. Evolution of the parameters of the ADALINE controller using (5.23)

5.2.2. Formulation and Simulations for GRBFNN Controller

The application of the algorithm for tuning the $\underline{y}$ parameters of a GRBFNN structure necessitates the following update rule. The vector $\underline{S}_a$ in (5.25) is defined in (5.24), in which the application for GRBFNN structure requires the replacement of $\underline{u}_{Ai}$ term with w^i of (2.9).

$$\dot{\underline{y}} = -K\left(\mu I + \rho \underline{w}\underline{w}^T\right)^{-1} \underline{S}_a T_{\underline{\phi}}(s_p) \tag{5.25}$$

where, $\underline{w}$ vector contains the hidden layer neuron outputs. In the simulations, the c_{ij} and σ_{ij} parameters of the Gaussian functions are kept constant. Figure 4.14 illustrates how these functions cover the relevant input space. Furthermore, K_b and K_e parameters have been chosen as 10^4 and 10^3 respectively.

The simulation results for GRBFNN controller are presented in Figures 5.6 through 5.11. It should be clear from Figure 5.6 that unlike the positional errors, at the early phases of the simulation; the velocity errors carry instantaneous spikes, whose effect is also apparent in Figures 5.8 and 5.9. On the other hand, the tuning strategy entails the minimization of the norm of the sensitivity vector because of the second term in the Lyapunov function of (5.7). The achievement of this together with a good tracking ability is characterized by the weighting coefficients μ and ρ. As the two terms of (5.7) are of close orders in magnitude, one can set $\mu = \rho = 1$, and the decision lies in between the minimization of the first and the second terms of (5.7); and the strategy makes concessions when required.

If the parameter evolution graphs in the Figures 4.19 and 5.10 are compared, one can directly infer that the parameter drift problem is alleviated by the use of the method discussed in this section.

Finally, the applied torque signal graph in Figure 5.7 clarifies that the smoothness of the control signals is improved considerably when compared to the result in Figure 4.16, which is based on the single-term Lyapunov function based tuning law given in (4.46).

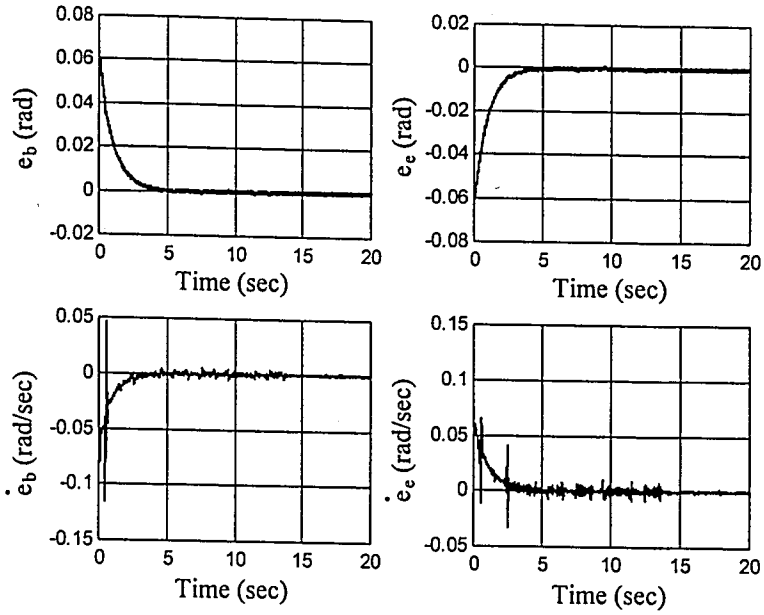

Figure 5.6. State tracking errors for GRBFNN controller using (5.25)

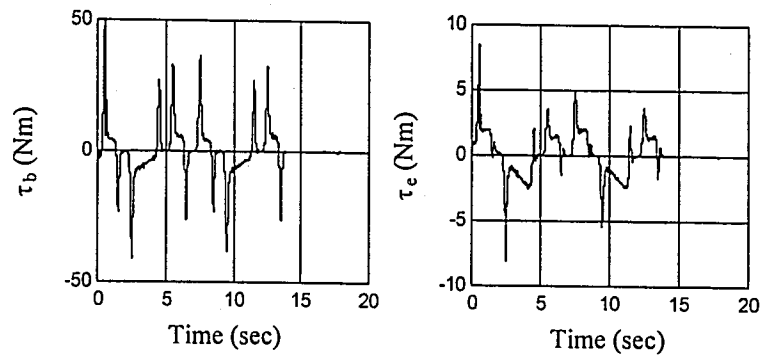

Figure 5.7. Applied torque signals for GRBFNN controller using (5.25)

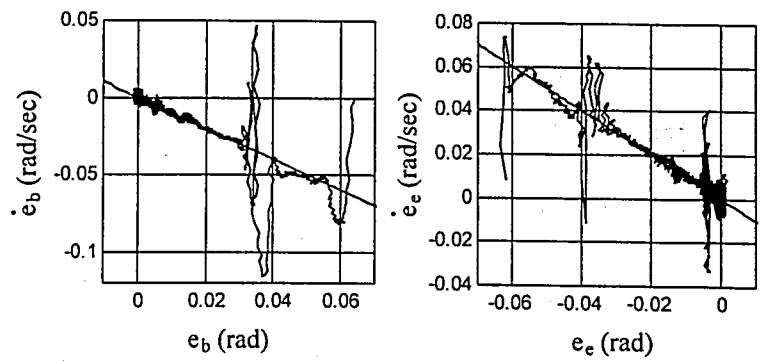

Figure 5.8. Trajectories in the phase plane for GRBFNN controller using (5.25)

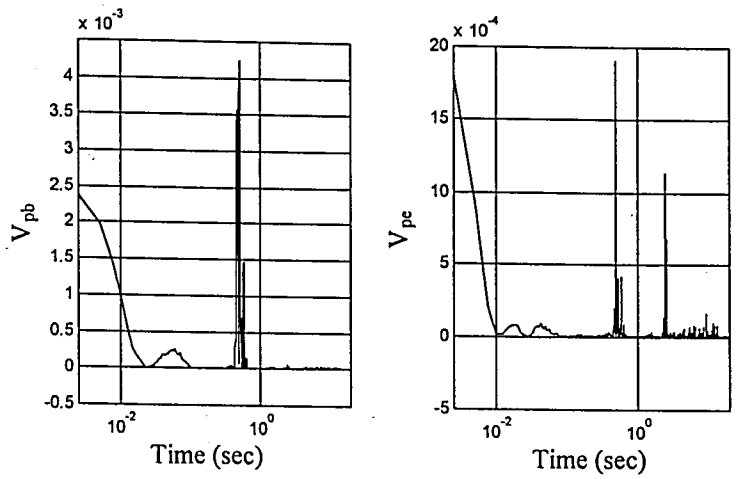

Figure 5.9. Time behavior of the Lyapunov functions for GRBFNN controller using (5.25)

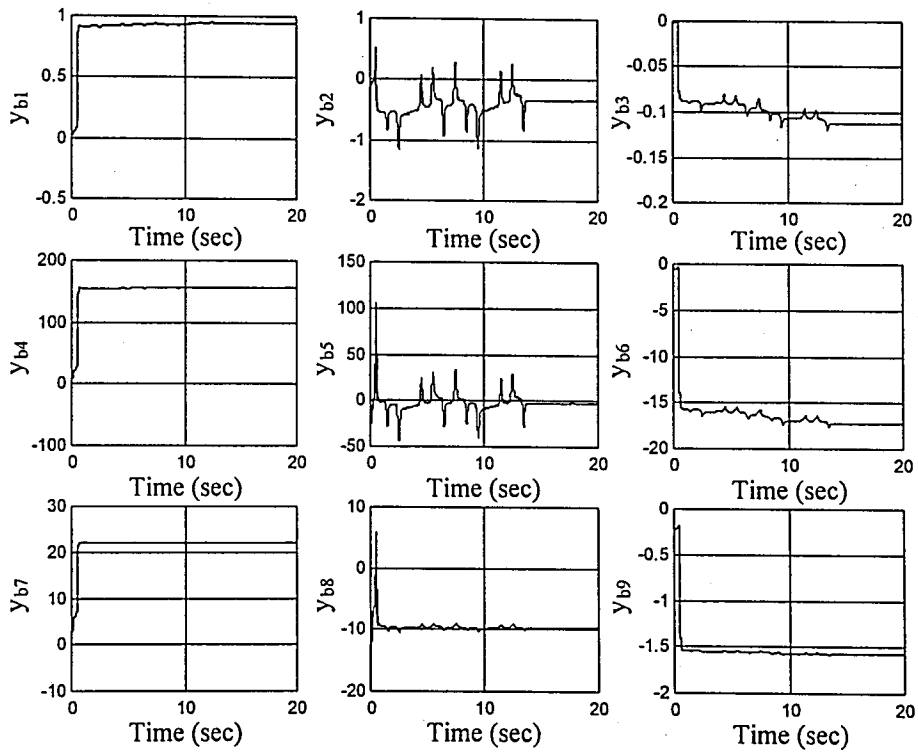

Figure 5.10. Evolution of the parameters of the base link GRBFNN controller using (5.25)

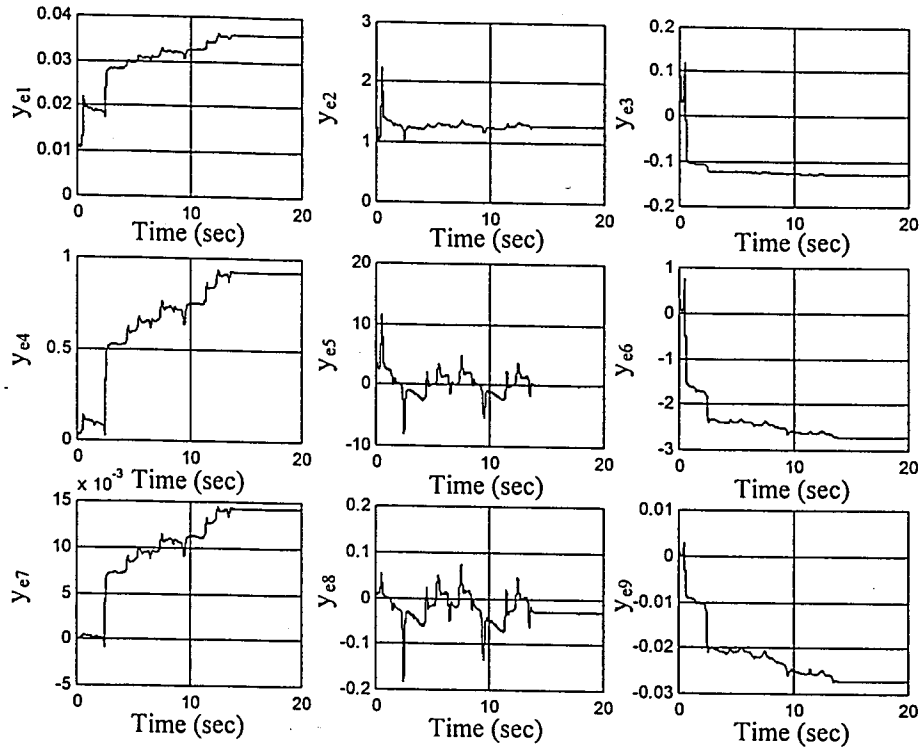

Figure 5.11. Evolution of the parameters of the elbow link GRBFNN controller using (5.25)

5.2.3. Formulation and Simulations for SFS Controller

In using the SFS structure as the controller for the robotic manipulator, the parameters of the defuzzifier are adjusted and the parameters of the membership functions are kept constant. For this purpose, the update rule can be expressed as

$$\dot{\underline{y}} = -K\left(\mu I + \rho \underline{w}_n \underline{w}_n^T\right)^{-1} \underline{S}_a T_{\underline{\phi}}\left(s_p\right) \tag{5.26}$$

The definition of the vector $\underline{S}_a$ in (5.26) is given in (5.24), the use of which for SFS structure requires the replacement of $\underline{u}_{Ai}$ term with w_n^i of (2.13).

In the simulations, the uncertainty bound parameters denoted by K_b and K_e have been chosen as 10^4 and $2\text{x}10^3$ respectively. The results obtained with SFS controller are depicted in Figures 5.12 through 5.17.

In Figure 5.12, the differences between the measured state variables and the desired state variables are illustrated. The performance in terms of the velocity tracking is better than those observed in the cases of ADALINE and GRBFNN controllers, which are depicted in Figures 5.1 and 5.6 respectively.

The torque signals produced by the SFS controller are given in Figure 5.13, from which the smoothness of the applied control signal is an outstanding feature of the approach with SFS type controller structure.

In Figure 5.14, the trajectories followed in the phase plane are illustrated. Apparently, during the reaching mode, the algorithm drives the state vector to the shown sliding manifold after just a few hittings. However, the augmented sliding manifold has nine more dimensions (R=9) because of the sensitivity terms. Therefore the small magnitude deviations during the sliding mode, which is visualized in two dimensions, should not mislead the reader. It must be emphasized that the algorithm is enforced to extract the path to the target by suitably minimizing the cost in (5.7), which is a weighted sum of two different terms.

The time behavior of the Lyapunov function of (5.7) for both controllers is depicted in Figure 5.15. A comparison between Figures 5.4, 5.9 and 5.15 recommends the use of the SFS structure in terms of the magnitude of the cost measure.

In Figures 5.16 and 5.17, the evolution of the adjustable parameters is illustrated. The observed results demonstrate that the effect of noise driven parameter adjustment activity is reduced and no parameter drift is observed.

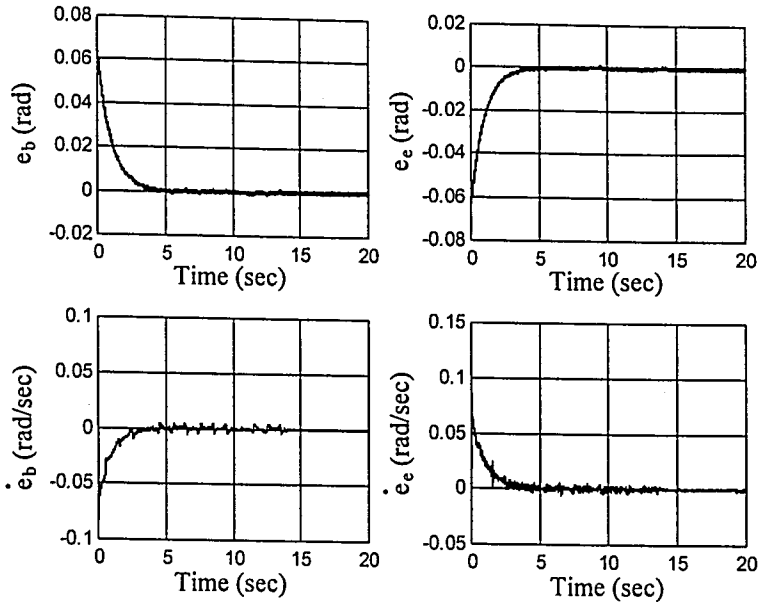

Figure 5.12. State tracking errors for SFS controller using (5.26)

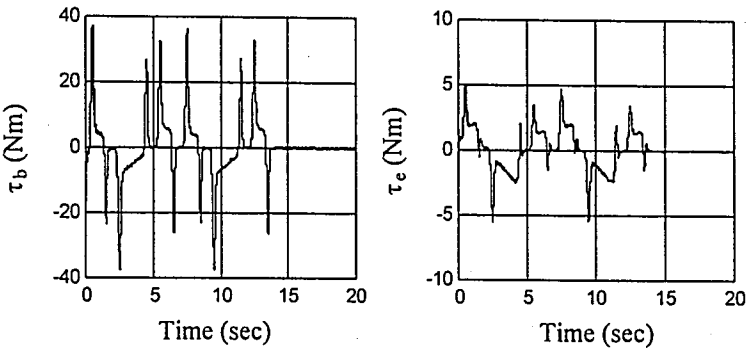

Figure 5.13. Applied torque signals for SFS controller using (5.26)

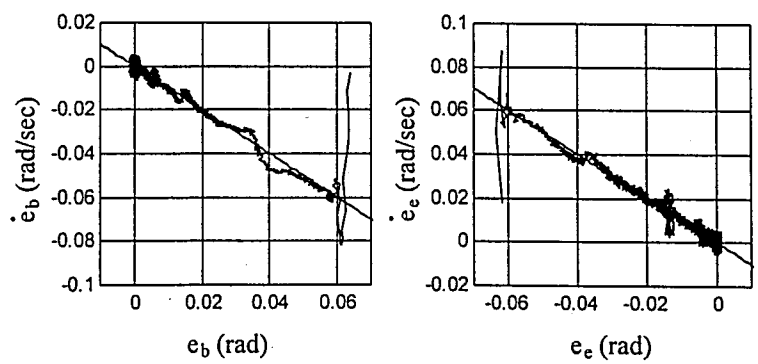

Figure 5.14. Trajectories in the phase plane for SFS controller using (5.26)

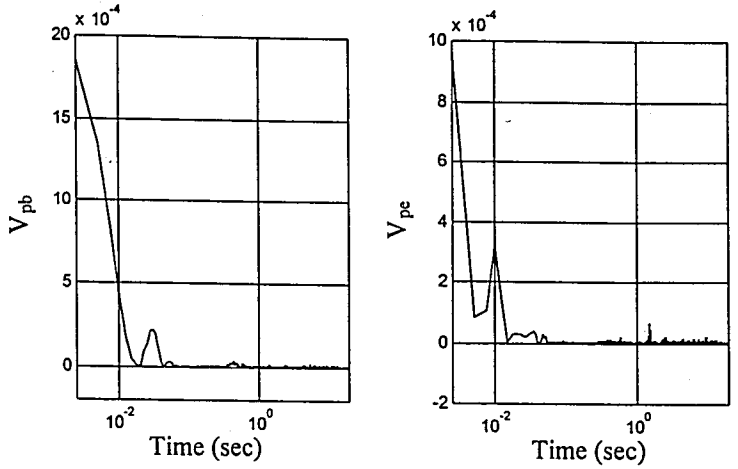

Figure 5.15. Time behavior of the Lyapunov functions for SFS controller using (5.26)

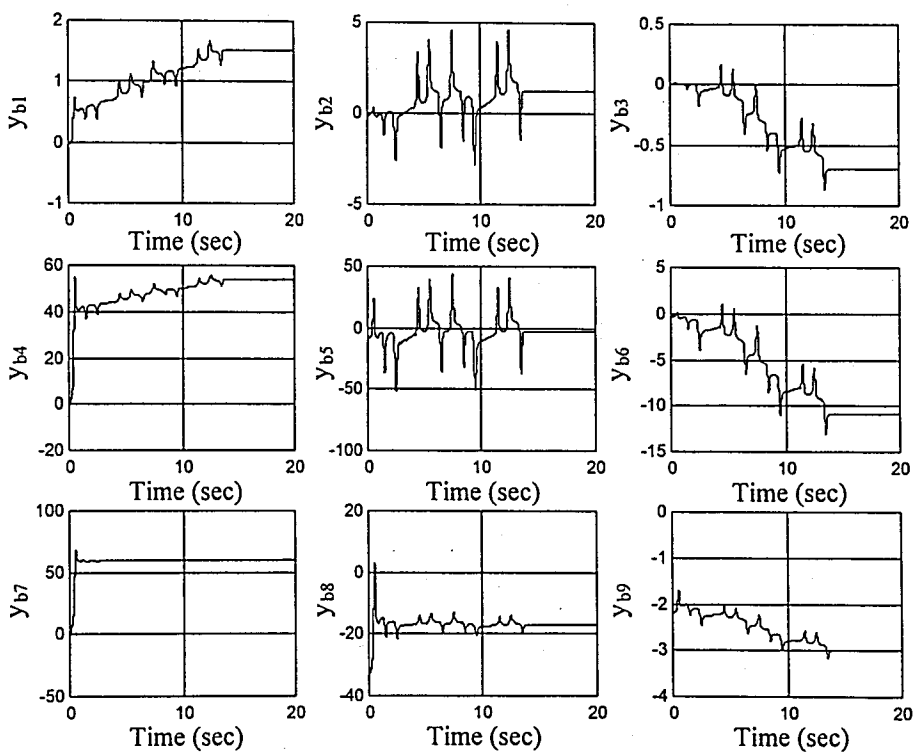

Figure 5.16. Evolution of the parameters of the base link SFS controller using (5.26)

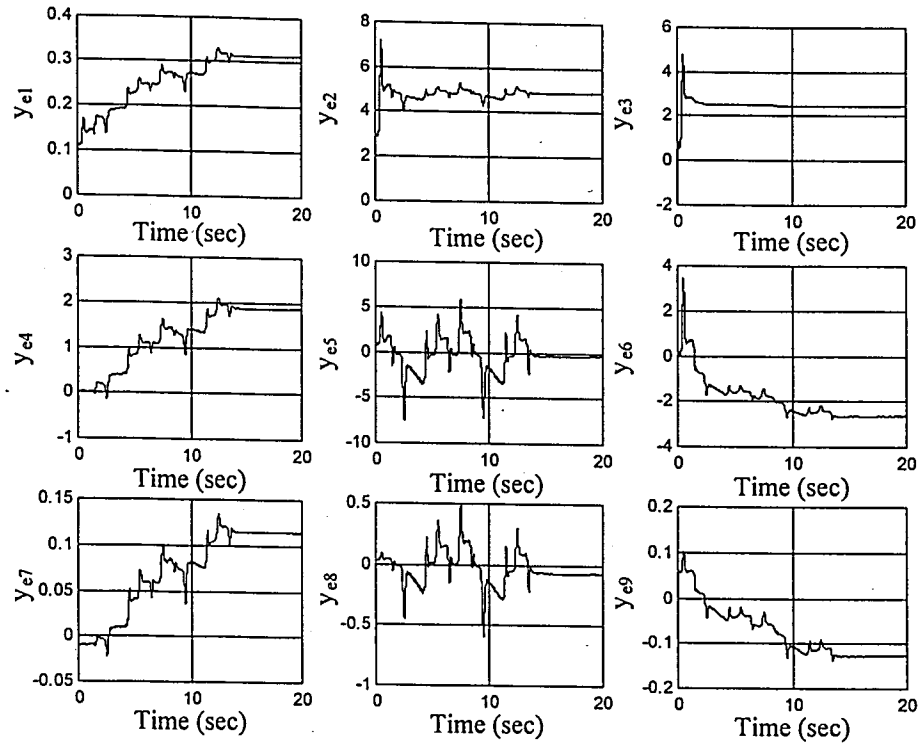

Figure 5.17. Evolution of the parameters of the elbow link SFS controller using (5.26)

5.2.4. Formulation and Simulations for ANFIS Controller

Utilizing the ANFIS structure as the controller for the robotic manipulator, the defuzzifier parameters, i.e. the entries of the Y matrix of (2.15) are adjusted and the parameters of the membership functions are kept constant. It should be noted that, the Y matrix is Rx(m+1) dimensional, with R being the number of rules contained in the rule base. Based on this, the entries of the j^{th} column of Y matrix, which is denoted by $\underline{Y}_j$ vector, have the dynamic behavior described as

$$\dot{\underline{Y}}_j = -K\left(\mu I + \dot{\rho}\Psi(s_p)\underline{u}^2_{A_j}\underline{w}_n\underline{w}_n^T\right)^{-1}\underline{S}_{a_j}T_{\underline{\phi}}(s_p) \quad j=1,\ldots,m+1 \tag{5.27}$$

The variable $\underline{S}_a$ for ANFIS controller is a matrix of dimensions Rx(m+1). The definition of the entry $\underline{S}_{a_{ij}}$ is given as

$$\underline{S}_{a_{ij}} = \frac{\Psi(s_p)\underline{w}_{n_i}\underline{u}_{A_j}}{\left|\Psi(s_p)\underline{w}_{n_i}\underline{u}_{A_j}\right| + \delta} \quad i=1,\ldots,R \tag{5.28}$$

In the simulation studies, the uncertainty bound parameters denoted by K_b and K_e have been chosen as 10^4 and 10^3 respectively. The simulation results are illustrated in Figures 5.18 through 5.21 except the graphs displaying the time behavior of the parameters, which would require 54 subplots in total.

The state tracking errors are illustrated in Figure 5.18, from which it can be inferred that objective of precise tracking is met. One should also notice that the velocity errors carry some small magnitude deviations because of the aim of the devised parameter tuning strategy.

In Figure 5.19, the produced control signals are depicted. The smoothness of the signals illustrated is a prominent feature of the approach.

The behavior in the phase space is illustrated in Figure 5.20, in which it is apparent that a sliding mode takes place after a few hittings to the sliding manifold. As discussed before, since the switching function is augmented with the sensitivity terms, the original sliding manifold denoted by $\underline{s}_A$ for the ANFIS controller has 27 more dimensions for each controller. In terms of the tracking precision, one can conclude that the strategy is able to achieve tracking precision, which is also confirmed by the time behavior of the Lyapunov functions depicted in Figure 5.21.

If the results obtained with the SFS controller is compared with those of the ANFIS controller, one cannot see any major difference. However, as each one of the ANFIS controllers possesses 27 adjustable parameters, a total of 54 parameters are adjusted during one sampling period, whereas this number for the SFS type and the GRBFNN type controllers is 18 in total. Therefore, the use of ANFIS structure is not recommended for the purpose of achieving the design objectives of the problem in hand with less computational cost. Nevertheless, the introduced learning strategy applies to the ANFIS architecture for its applications other than the one studied here.

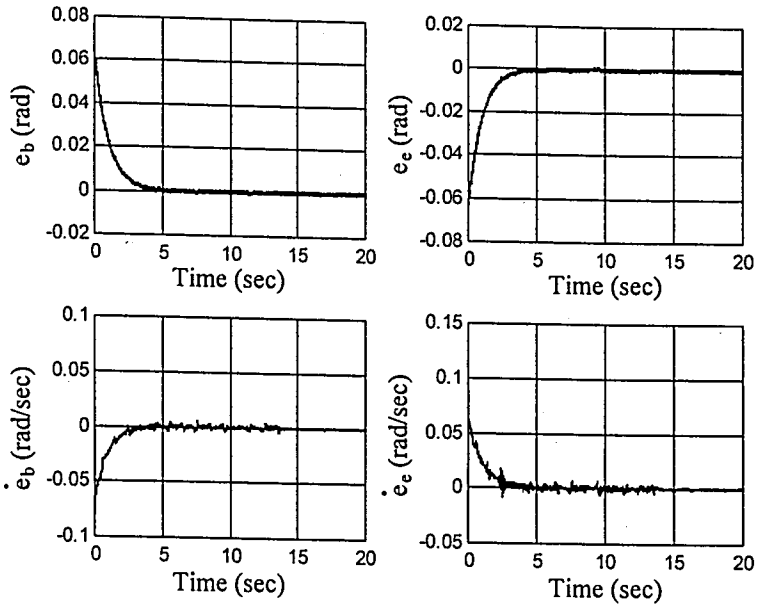

Figure 5.18. State tracking errors for ANFIS controller using (5.27)

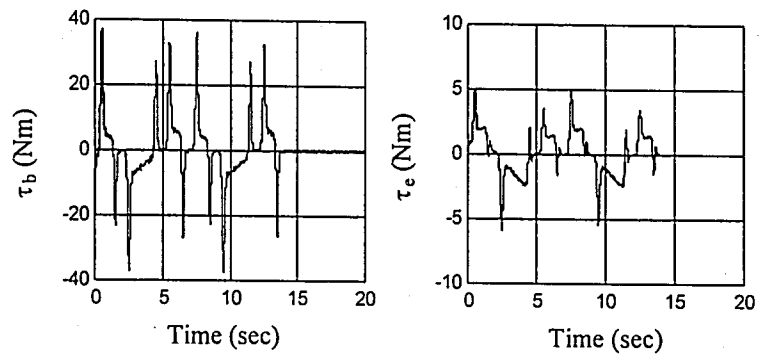

Figure 5.19. Applied torque signals for ANFIS controller using (5.27)

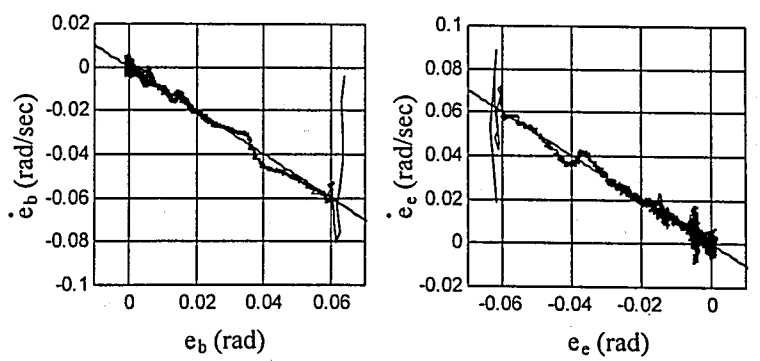

Figure 5.20. Trajectories in the phase plane for ANFIS controller using (5.27)

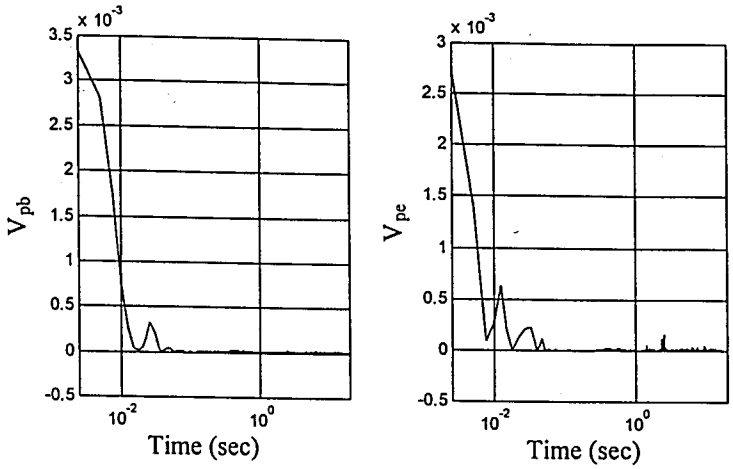

Figure 5.21. Time behavior of the Lyapunov functions for ANFIS controller using (5.27)

5.2.5. Formulation and Simulations for FNN Controller

The formulation of the algorithm for FNN structure deserves a special care, because the output of the structure is not linear in some of the parameters. It should be clear that the terms in (5.14) and (5.16) are the ordinary gradient vector and the Hessian matrix respectively. However, the design needs the terms in (5.15) and (5.17), the evaluation of which is quite tedious, and once evaluated, processing of which will be a sort of bound extraction. In the simulations, the effect of the last two terms of (5.13) is absorbed into the uncertainty bound parameters K_b and K_e, which have been selected as 10^4 and 10^3 respectively. Besides, a FNN structure having two inputs, two hidden neurons with hyperbolic tangent nonlinear activation functions and one-output is used as the controller for each axis. The structure and the labeling of the adjustable parameters have explicitly been illustrated in Figure 5.22. Clearly, for such a simple FNN structure, a total of nine adjustable parameters are of interest.

According to the simulation studies performed with FNN controller, as illustrated in Figure 5.23, the tracking precision is not found to be satisfactory. Although the positional errors tend to remain in the vicinity of the origin, this is not the case in velocity errors, which display a considerable amount of instant spikes.

As illustrated in Figure 5.24, the produced control signal does not have as much amount of sharp changes as the ADALINE controller produces (See Figure 5.2), the values

produced by the controllers are apparently not the values to drive the system to a sliding mode.

In Figure 5.25, the behavior in the phase plane is illustrated for both links. Because of the fluctuations in the velocity errors, a desired sliding mode is not observed in the phase space. The time behavior of the Lyapunov functions depicted in Figure 5.26 also justifies the poor performance with FNN structure.

It must be noted that the parameters of the FNN controller do not have drifts because of the noise driven adjustment activity in the vicinity of the origin, but the observed time evolution of the parameters is not the sought one.

A final remark on the use of FNN structure as the controller and the observed poor performance must account for the terms in (5.15) and (5.17). Since the output of the FNN controller is not linear in some of its parameters, the tuning law described in (5.11) cannot compensate the deficiencies caused by this nonlinearities and needs modifications taking care of the parameters influencing the output nonlinearly. An important result of these remarks explains why the adjustment law in (5.11) is postulated solely for the structures, whose outputs are linear functions of the adjustable parameters.

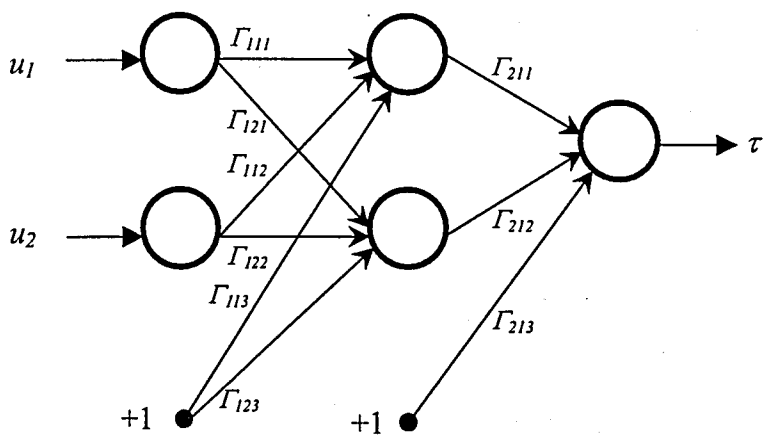

Figure 5.22. The configuration of the FNN used as controller

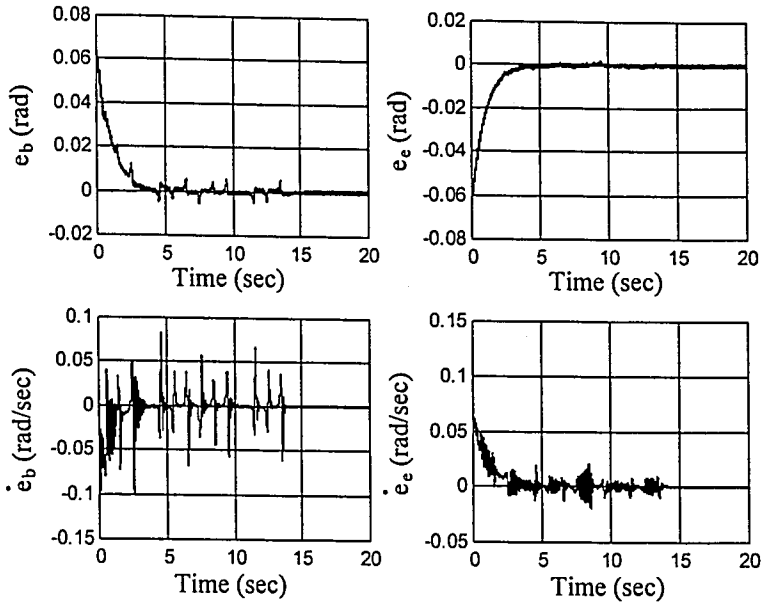

Figure 5.23. State tracking errors for FNN controller using (5.11)

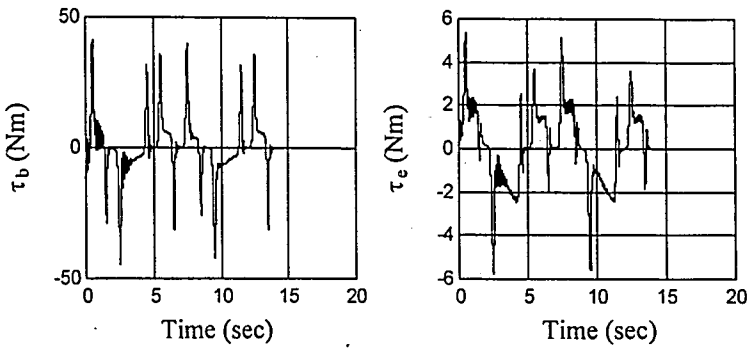

Figure 5.24. Applied torque signals for FNN controller using (5.11)

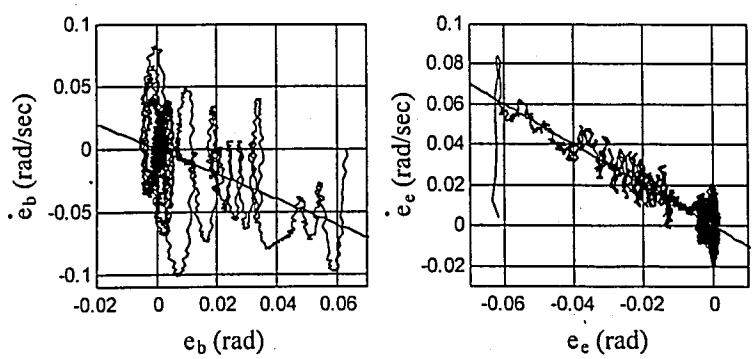

Figure 5.25. Trajectories in the phase plane for FNN controller using (5.11)

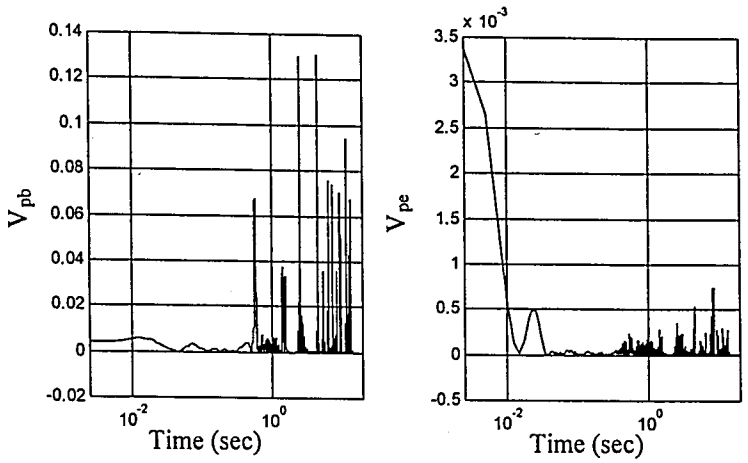

Figure 5.26. Time behavior of the Lyapunov functions for FNN controller using (5.11)

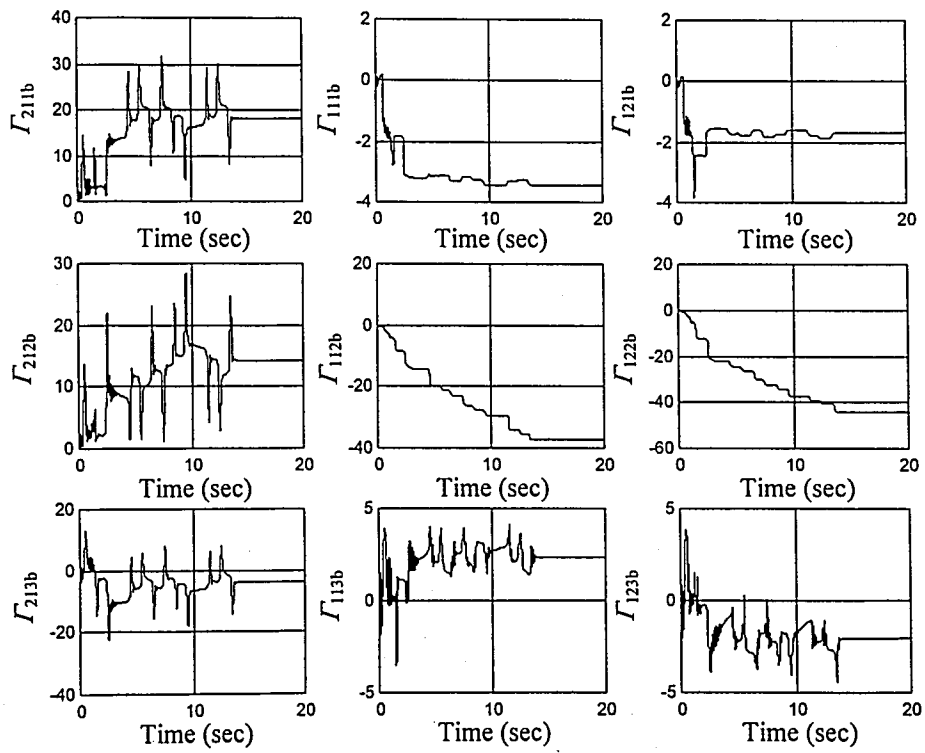

Figure 5.27. Evolution of the parameters of the base link FNN controller using (5.11)

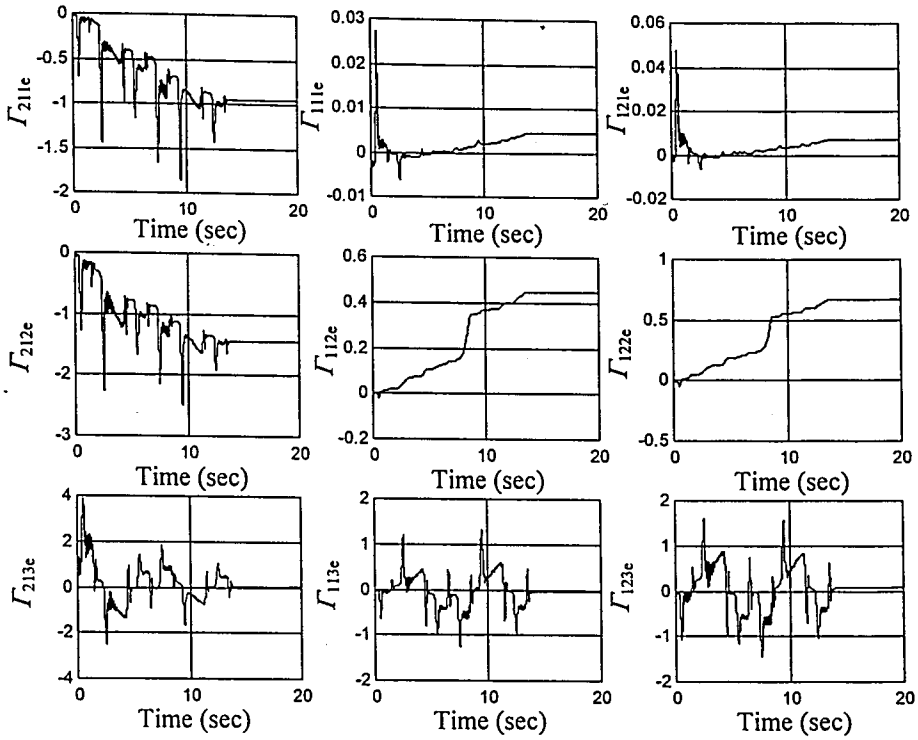

Figure 5.28. Evolution of the parameters of the elbow link FNN controller using (5.11)

5.3. A Discussion on the Results

The results obtained using the dynamic parameter adaptation law in (5.11) demonstrate that the method is successful as far as the controller structures in which the output is linear are concerned. For the FNN structure, the nonlinear dependence of the output and some of the adjustable parameters makes it difficult to manipulate the terms in (5.15) and (5.17). For the particular application example considered, numerous simulations carried out with this strategy have shown that if the uncertainty bound is increased for an improvement on the base link velocity error behavior, the elbow link is adversely influenced because of the interdependency between the variables involved.

Another important point that should be commented upon is the computational requirements of the algorithm. It is apparent that as the number of adjustable parameters increases, the necessity to matrix inversion causes more computations between the successive control periods. This aspect of the algorithm may be a restrictive factor for real-time applications. Because of the matrix inversion at each control period, the time required

for the evaluation of the relevant time derivatives for the adjustable parameters is more than the method based on a single-term Lyapunov function discussed in the fourth section. Nevertheless, the performance of the algorithm is good in the sense of robustness and tracking performance.

A comparison between the tuning strategies discussed in the fourth section and this section stipulates that the effect of noise driven parameter adjustment activity observed using the tuning law in (5.11) is less than that of the tuning law in (4.9). The simulations carried out clarify that the parameter drift problem is alleviated by the use of the method discussed in this section.

6. LEARNING RATE SELECTION IN GRADIENT BASED TRAINING STRATEGIES FOR SLIDING MODE CONTROL

One of the major problems in training of computationally intelligent architectures is the extraction of the parameter tuning information, the construction of which is generally achieved by utilizing the gradient based methods. Error backpropagation technique is the most popular method employed in parameter adjustment. The application spectrum of the algorithm covers a wide variety of areas extending from speech and pattern recognition to identification and control of nonlinear dynamic systems [38-39,47,50,86-104]. Once the designer decides on the form of the parameter update rule, selection of an appropriate learning rate sequence arises as a problem. When the learning rate assumes small values convergence takes a long time, while large values can increase the speed of convergence at the cost of introducing an oscillatory behavior, which may ultimately result in divergence. In the literature, several modifications to the original form of gradient based parameter adjustment rules are proposed. Namely, the adaptation of the learning rate, introduction of a momentum term or assigning an individual learning rate to each adjustable parameter are the mostly used ones [1]. Nevertheless, utilizing these precautions transfers the ambiguity on the learning rate to the selection of the parameters of learning rate adaptation mechanism or to the selection of momentum term coefficient.

In this section, three different forms of the gradient technique are considered and the analytic form of the learning rate is extracted. The form of the learning rate for each case is formulated for ADALINE structure. The modifications for GRBFNN, SFS and ANFIS structures are also presented.

6.1. Standard Gradient Rule and the Proposed Learning Rate Selection

Consider the Lyapunov function in (4.12). If this measure is assumed as the cost function, the ordinary gradient based parameter update rule prescribes the parameter derivatives as given in (6.1) in which, Λ is an $(m+1)\text{x}(m+1)$ matrix and $\underline{N}_{\phi}$ is as defined in (6.2). The procedure explained in (6.1) and (6.2) is also known as MIT rule in the literature [34].

$$\dot{\underline{\phi}} = \Lambda \underline{N}_{\phi} \tag{6.1}$$

$$\underline{N}_{\phi} = s_c \frac{\partial \tau}{\partial \underline{\phi}} \tag{6.2}$$

The time derivative of the Lyapunov function can be evaluated as follows.

$$\begin{aligned} \dot{V}_c &= \dot{s}_c s_c \\ &= \left(\dot{\underline{\phi}}^T \underline{u}_A + \underline{\phi}^T \dot{\underline{u}}_A - \dot{\tau}_d \right) s_c \\ &= \underline{N}_{\phi}^T \Lambda^T \underline{u}_A s_c + \left(\underline{\phi}^T \dot{\underline{u}}_A - \dot{\tau}_d \right) s_c \\ &\le \underline{N}_{\phi}^T \Lambda^T \underline{u}_A s_c + \left(B_{\phi} B_{\dot{u}_A} + B_{\dot{\tau}_d} \right) |s_c| \end{aligned} \tag{6.3}$$

Selecting a strictly positive scalar κ and enforcing the equality

$$\dot{s}_c s_c = -\kappa |s_c| \tag{6.4}$$

guarantees the negativeness of the time derivative of the Lyapunov function and leads to the equality described as

$$\underline{N}_{\phi}^T \Lambda^T \underline{u}_A s_c = -\left(\kappa + B_{\phi} B_{\dot{u}_A} + B_{\dot{\tau}_d} \right) |s_c| \tag{6.5}$$

Or, more explicitly;

$$\underline{N}_{\phi}^T \Lambda^T \underline{u}_A = -K \operatorname{sgn}(s_c) \tag{6.6}$$

where, K is as defined as

$$K = \kappa + B_{\phi} B_{\dot{u}_A} + B_{\dot{\tau}_d} \tag{6.7}$$

The solution to the equation in (6.6) is

$$\Lambda = -\frac{\underline{u}_A \underline{N}_{\phi}^T}{\underline{N}_{\phi}^T \underline{N}_{\phi} \underline{u}_A^T \underline{u}_A} K \mathrm{sgn}(s_c) \tag{6.8}$$

which is the form of the learning rate matrix for the structures, whose outputs are linear in adjustable parameters. The use of the above quantity as the learning rate enforces the differential equation in (4.9), a detailed analysis of the implications of which is given in the fourth section. The explicit form of the selection described in (6.8) for ADALINE, GRBFNN and SFS structures require the learning rates formulated in (6.9), (6.10) and (6.11) respectively.

$$\Lambda = \begin{cases} -\dfrac{K \underline{u}_A \underline{u}_A^T}{\left(\underline{u}_A^T \underline{u}_A\right)^2 |s_c|} & s_c \neq 0 \\ 0_{(m+1)\mathrm{x}(m+1)} & s_c = 0 \end{cases} \tag{6.9}$$

$$\Lambda = \begin{cases} -\dfrac{K \underline{w} \underline{w}^T}{\left(\underline{w}^T \underline{w}\right)^2 |s_c|} & s_c \neq 0 \\ 0_{(h+1)\mathrm{x}(h+1)} & s_c = 0 \end{cases} \tag{6.10}$$

$$\Lambda = \begin{cases} -\dfrac{K \underline{w}_n \underline{w}_n^T}{\left(\underline{w}_n^T \underline{w}_n\right)^2 |s_c|} & s_c \neq 0 \\ 0_{(R+1)\mathrm{x}(R+1)} & s_c = 0 \end{cases} \tag{6.11}$$

For the ANFIS architecture, as discussed in section 5.2.4, the j^{th} column of the defuzzifier matrix Y is defined as $\underline{Y}_j$ and the learning rate matrix multiplying the gradient information for this column is denoted by Λ_j (j=1,...,m+1) and defined as

$$\Lambda_j = \begin{cases} -\dfrac{K\underline{w}_n\underline{w}_n^T}{\left(\underline{w}_n^T\underline{w}_n\right)^2 \underline{u}_{A_j}^2 |s_c|} & s_c \neq 0 \text{ and } \underline{u}_{A_j} \neq 0 \\ 0_{RxR} & s_c = 0 \text{ or } \underline{u}_{A_j} = 0 \end{cases} \tag{6.12}$$

in which, Λ_j is RxR dimensional for each column of Y.

In the simulations the ADALINE structure has been used as the controller and the learning rate variable (Λ) is a (m+1)x(m+1) matrix. The plant and the simulation settings are as those discussed in section 4.3.2. The aim here is to demonstrate that the phase plane behavior depicted in Figure 4.11 could be obtained by adopting the adjustment rule in (6.1) and the learning rate formulated in (6.9).

As pointed out above, if one uses a GRBFNN structure with h hidden neurons or a SFS structure with R rules, the corresponding dimensions of the learning rate matrix would be hxh and RxR respectively. Similarly, for ANFIS structure, if the rule base contains R rules, for each column of the defuzzifier matrix Y, a RxR dimensional matrix would be required to describe the learning rate matrix of $\underline{Y}_j$. Therefore, the results presented in this section are confined only to the application to ADALINE controller structure.

In Figure 6.1, the entries of the learning rate matrix for the base link controller (Λ_b), and in Figure 6.2, those for the elbow link controller (Λ_e) are illustrated. It is seen that the quantities can assume values larger than unity and less than zero.

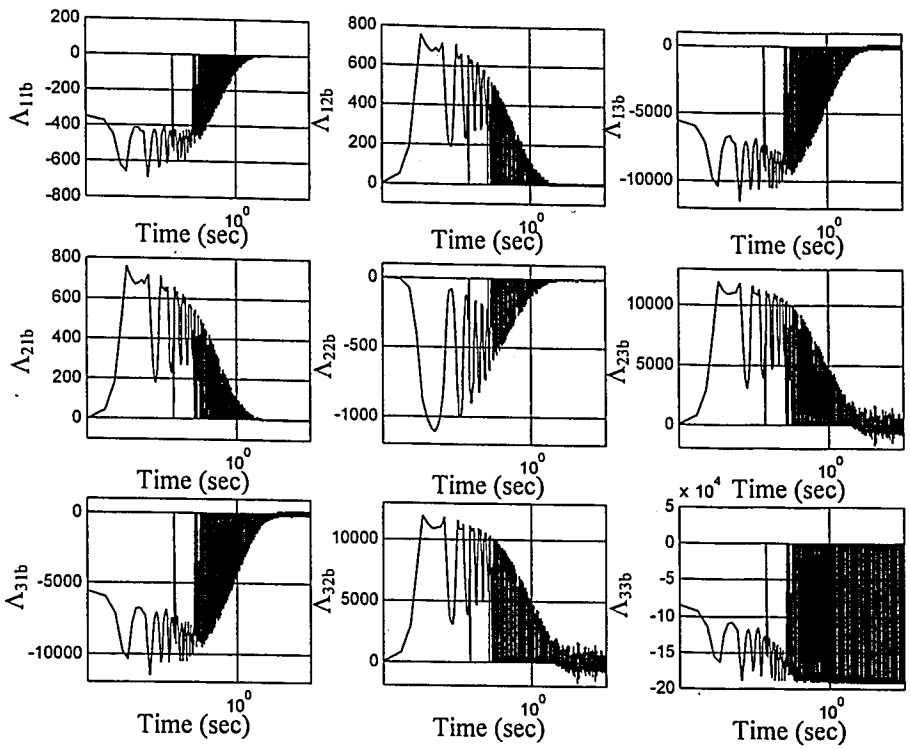

Figure 6.1. Time behavior of the base link controller parameters (Λ_b) using (6.9)

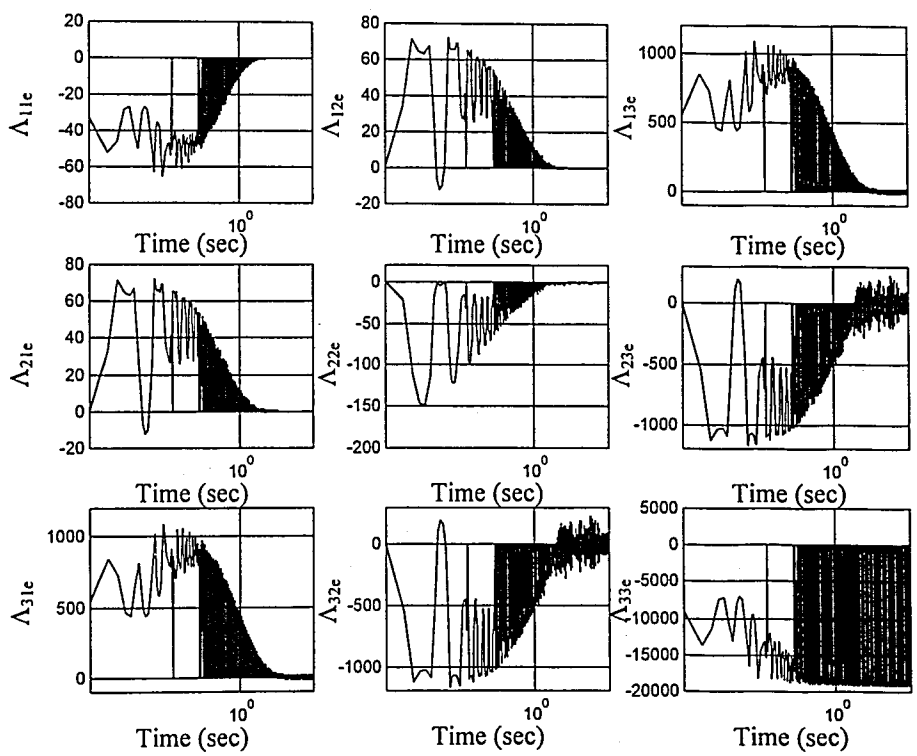

Figure 6.2. Time behavior of the elbow link controller parameters (Λ_e) using (6.9)

6.2. A Modified form of the Gradient Rule and the Proposed Learning Rate Selection

The approach analyzed in section 6.1 considers that the quantity in (6.2) carries the necessary information for tuning the parameters. However, the important part of the information contained in this quantity is the direction knowledge, i.e. at each instant of time, the direction to a better parameter vector is pointed by the sign of $\underline{N}_\phi$. Therefore, a modified form of the update dynamics can be constructed as

$$\dot{\underline{\phi}} = \Lambda \operatorname{sgn}\left(\underline{N}_\phi\right) \tag{6.13}$$

Performing the mathematical manipulations as in (6.3), and imposing the condition in (6.4) for stability leads to the equality

$$\operatorname{sgn}\left(\underline{N}_\phi^T\right)\Lambda^T \underline{u}_A s_c = -\left(\kappa + B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d}\right)|s_c| \tag{6.14}$$

which determine the form of the learning rate for the law in (6.13). Or, more explicitly;

$$\operatorname{sgn}\left(\underline{N}_\phi^T\right)\Lambda^T \underline{u}_A = -K\operatorname{sgn}(s_c) \tag{6.15}$$

where, K is as defined in (6.7). The solution to the equation in (6.15) is

$$\Lambda = -\frac{\underline{u}_A \operatorname{sgn}(\underline{N}_\phi^T)}{\operatorname{sgn}(\underline{N}_\phi^T)\operatorname{sgn}(\underline{N}_\phi)\,\underline{u}_A^T \underline{u}_A} K\operatorname{sgn}(s_c) \tag{6.16}$$

which is the form of the learning rate matrix for the models in which the output is a linear function of the adjustable parameters. The explicit form of the learning rate matrix for ADALINE, GRBFNN and SFS structures are described in (6.17), (6.18) and (6.19) respectively.

$$\Lambda = -\frac{K\underline{u}_A \operatorname{sgn}\left(\underline{u}_A^T\right)}{\operatorname{sgn}(\underline{u}_A^T)\operatorname{sgn}(\underline{u}_A)\,\underline{u}_A^T\underline{u}_A} \tag{6.17}$$

$$\Lambda = -\frac{K\underline{w}1_{1xh}}{h\,\underline{w}^T\,\underline{w}} \tag{6.18}$$

$$\Lambda = -\frac{K\underline{w}_n\,1_{1xR}}{R\,\underline{w}_n^T\,\underline{w}_n} \tag{6.19}$$

For the case of ANFIS structure, the learning rate matrix denoted by Λ_j multiplying the j^{th} column of Y matrix is described as

$$\Lambda_j = \begin{cases} -\dfrac{K\underline{w}_n 1_{1xR}}{R\underline{w}_n^T\underline{w}_n\left|\underline{u}_{A_j}\right|} & \underline{u}_{A_j} \neq 0 \\ 0_{RxR} & \underline{u}_{A_j} = 0 \end{cases} \tag{6.20}$$

The purpose of obtaining the behavior in the phase illustrated in Figure 4.11 is maintained in this section too. The proposed form of the learning rate for the adjustment mechanism in (6.13) is given in (6.17) for the ADALINE controller structure. The dimensionality of the Λ matrix for SFS, GRBFNN and ANFIS structures is as those discussed in the section 6.1 and the visualization them of are skipped because of the necessity to illustrate the entries of an hxh or RxR dimensional matrices.

The corresponding learning rate sequences for the base and the elbow links are illustrated in Figures 6.3 and 6.4 respectively. What the reader should notice in these figures is again the removal of the conventional boundaries of the definition interval of the learning rate.

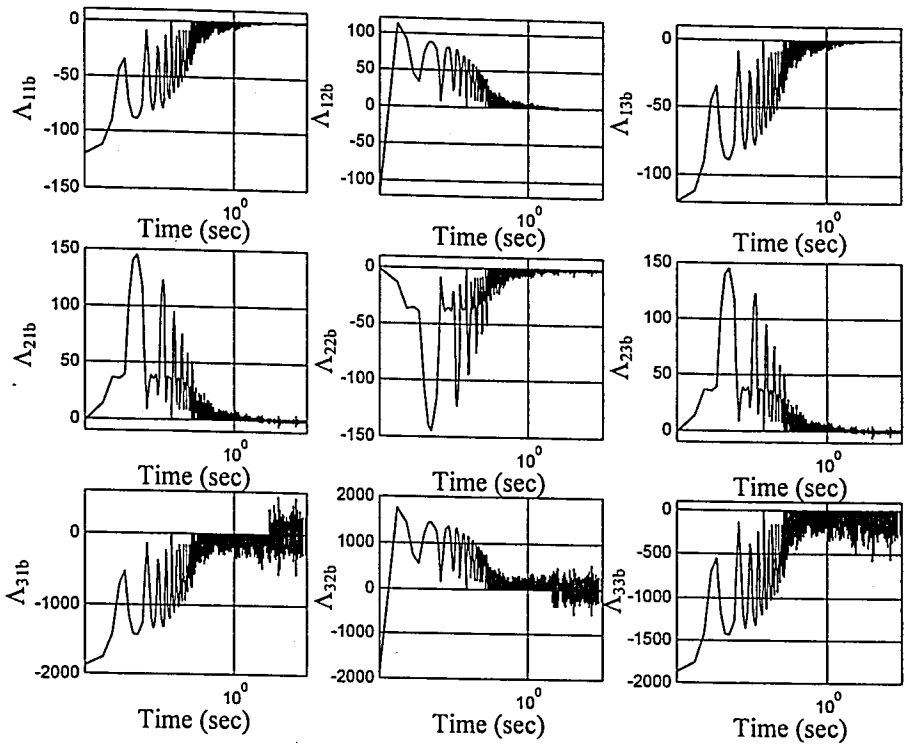

Figure 6.3. Time behavior of the base link controller parameters (Λ_b) using (6.17)

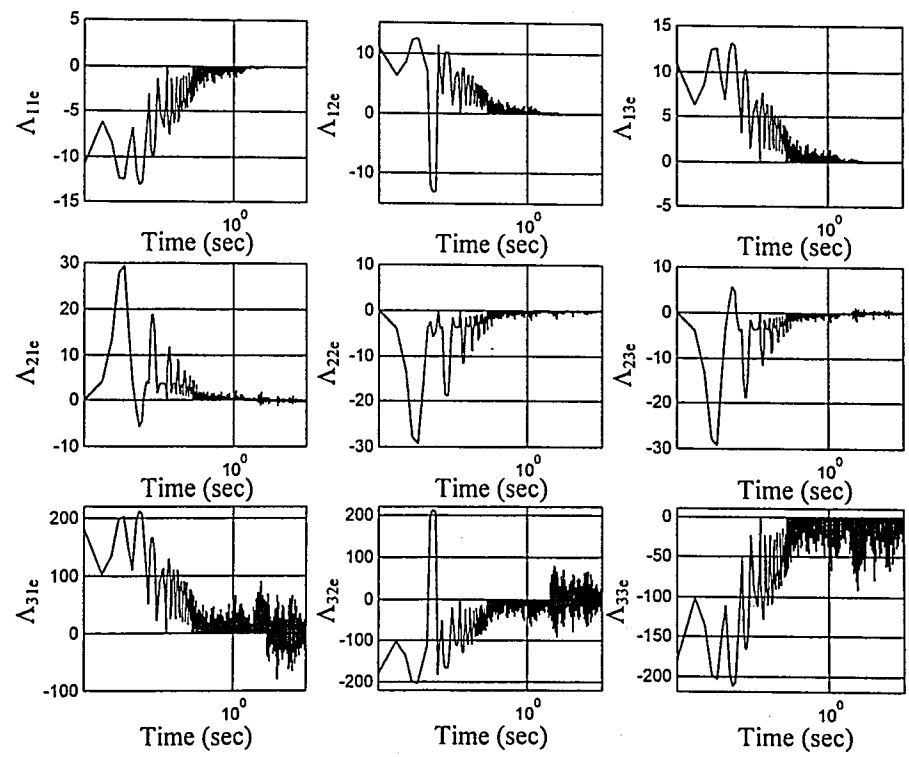

Figure 6.4. Time behavior of the elbow link controller parameters (Λ_e) using (6.17)

6.3. Learning Rate Selection in the Gradient Rule Using the Absolute Value of the Error as the Cost Function

A third alternative of parameter tuning as discussed by Åström and Wittenmark [34] assumes the absolute value of the zero learning-error level as the cost function, which is defined as

$$J = |s_c| \tag{6.21}$$

The form of the parameter adjustment for this sort of a choice is formulated as

$$\dot{\underline{\phi}} = -\Lambda \frac{\partial s_c}{\partial \underline{\phi}} \operatorname{sgn}(s_c) \tag{6.22}$$

However, the ambiguity concerning the learning rate Λ is still a problem in practical applications. If the mathematical manipulations of (6.3) are carried out, one ends up with

$$-\left(\frac{\partial s_c}{\partial \underline{\phi}}\right)^T \Lambda^T \underline{u}_A = -\left(\kappa + B_\phi B_{\dot{u}_A} + B_{\dot{\tau}_d}\right) \tag{6.23}$$

which is a result of imposing the condition in (6.4) for stability. Or, more explicitly;

$$\left(\frac{\partial s_c}{\partial \underline{\phi}}\right)^T \Lambda^T \underline{u}_A = K \tag{6.24}$$

where, K is as defined in (6.7). For the ADALINE structure, the solution to the equation in (6.24) can be given as

$$\Lambda = \frac{K}{\underline{u}_A^T \underline{u}_A} \tag{6.25}$$

which states that the learning rate is a scalar quantity. The explicit form of the learning rate for GRBFNN, SFS and ANFIS structures are described in (6.26), (6.27) and (6.28) respectively. For the case of ANFIS structure, the scalar learning rate denoted by Λ_j should be understood as the multiplier of the j^{th} column of Y matrix of the defuzzifier.

$$\Lambda = \frac{K}{\underline{w}^T \underline{w}} \tag{6.26}$$

$$\Lambda = \frac{K}{\underline{w}_n^T \underline{w}_n} \tag{6.27}$$

$$\Lambda_j = \begin{cases} \dfrac{K}{\underline{w}_n^T \underline{w}_n \underline{u}_{A_j}^2} & \underline{u}_{A_j} \neq 0 \\ 0 & \underline{u}_{A_j} = 0 \end{cases} \tag{6.28}$$

The simulation results presented in this section concern the derivation of a learning rate sequence for the adjustment law described in (6.25), which is for ADALINE structure. In contrast to the sections 6.1 and 6.2, the extracted learning rate parameter in this section is a scalar for each controller structure including the GRBFNN, SFS and ANFIS architectures. Only in the ANFIS structure, each column of the defuzzifier matrix utilize an individually quantified learning rate sequence, i.e. the number of Λ parameters is equal to the number of the columns of the Y matrix.

The aim is as discussed in the sections 6.1 and 6.2. Briefly, the learning rate in the training strategy of (6.22) has been reformulated such that the motion in the phase plane depicted in Figure 4.11 is obtained. The simulation settings discussed in this section are the same as those discussed in section 4.3.2.

In Figure 6.5, the obtained learning rate sequences are depicted. As is clear from the definition of $\underline{u}_A$ given in (2.2) and the definition of Λ in (6.25), the final value of the sequences are the relevant uncertainty bound values denoted by K.

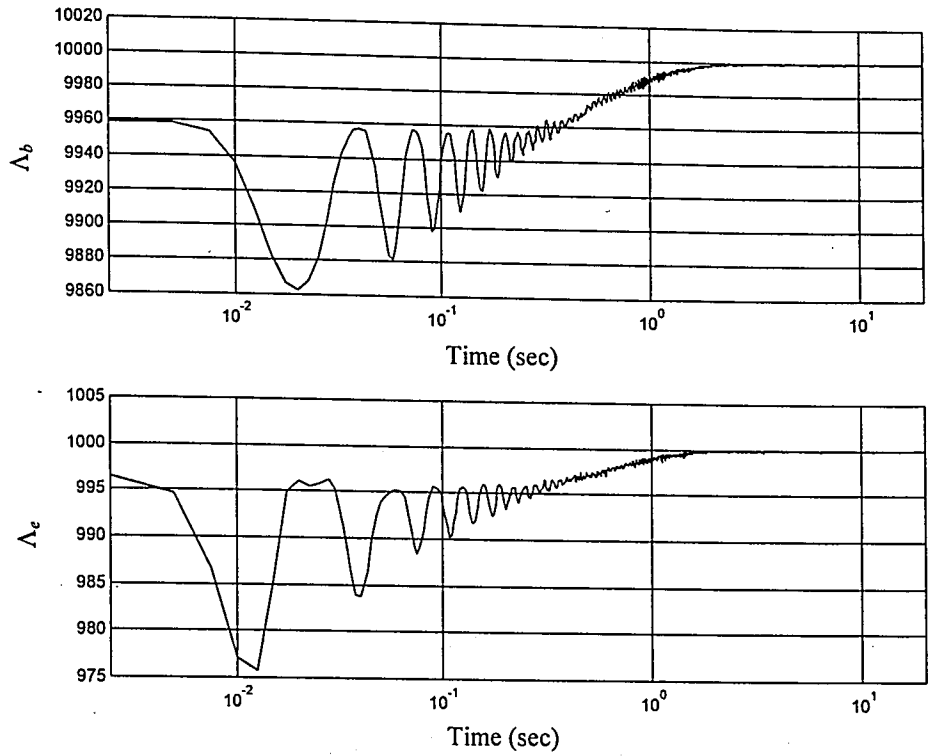

Figure 6.5. Time behavior of the base and elbow link controller parameters (Λ_b and Λ_e) using (6.25)

6.4. A Discussion on the Results

The performance of the training strategies of the computationally intelligent systems is intimately related to the selection of the design parameters. Selection of an appropriate learning rate is one of the factors influencing the learning performance. In the cases where the target behavior is defined, it is a difficult problem to construct an appropriate learning rate sequence leading to the desired behavior.

In this section, three different forms of parameter adjustment laws are studied in the sense of formulating the learning rate sequence resulting in the achievement of the zero learning-error level at the output node of a computationally intelligent controller. The constructed form of the learning rate sequence drives the controller parameters to values such that the produced control signal drives the plant under control to a sliding regime.

What should be emphasized on the obtained results are the magnitude and the sign of the learning rate coefficients. Apparently, as opposed to what is discussed in the standard practice, the studied forms of the learning rates can assume values greater that unity and less than zero. A fair comparison between the conventional approaches and the approaches presented in this section should distinguish the cases in which the target behavior is well defined as postulated in this section. However, the conventional selection is still valid for the cases where the path to the target is not clearly defined.

The application area of the approaches presented in this section is limited to on-line parameter adjustment strategies for flexible structures whose outputs are linear in the adjustable parameters. Furthermore, contrary to the heuristics of the conventional learning rate selection, the learning rate parameter is a matrix for the first two approaches and the use of which is an increase in the computational complexity. The computational burden for simple structures, e.g. ADALINE model, can be negligible but for the structures having a considerable number of adjustable parameters, the cost attached to the learning rate construction may be inefficient and one might choose the conventional procedures.

Among the three different forms of the parameter adjustment strategies, the last one seems to be the simplest one with less computational cost comparably to the others.

7. CONCLUSIONS

Solutions to problems having complicated mathematical representations, uncertain terms and mismatches with real time counterparts are difficult to achieve with conventional approaches. The field of computational intelligence offers practical alternatives to handle the stated difficulties with the help of expert knowledge. There is a continual progress in the design and the development of computationally intelligent systems and the applications reported in literature display the strengths of such approaches.

Artificial neural networks and fuzzy inference systems are the main constituents of the area of computational intelligence, the structures of which have extensively been used for solving problems having the stated difficulties. A substantial problem associated with the utilization of computationally intelligent systems is the parameter tuning strategy to be adopted for achieving a specified task. In this thesis, the problem is studied in the form of two subtasks. These are namely the use of variable structure control methodology in parameter tuning and sliding mode control of a nonlinear plant. The former enforces the parameters of the neuro-fuzzy structure to values satisfying a criterion defined on the output of the structure, while the latter results in the observation of a sliding mode in the dynamics of the plant. The achievement of the two subtasks clearly leads to the acquisition of the invariance properties of sliding control both in the parameter adjustment and in the control loops.

The plant considered in the simulations is the dynamic model of a two degrees of freedom direct drive robotic manipulator. The approaches discussed throughout the thesis assume that the governing equations of the plant dynamics are unknown, which is in contrast to the case of conventional variable structure controller design methodology.

The controller structures analyzed in the study are the Adaptive Linear Elements, Gaussian Radial Basis Function Neural Networks, Standard Fuzzy Systems, Adaptive Neuro-Fuzzy Inference Systems and Feedforward Neural Networks.

The first approach for parameter tuning is based on the adoption of a dynamic adjustment mechanism, which is applied to the controller structures, whose outputs are linear in adjustable parameters. The method presented extracts the parameter tuning information from a single term Lyapunov function. It is shown that the negative definiteness conditions of the time derivative of the selected Lyapunov function can be converted to an equivalent Lyapunov function defined on the output of the controller. The procedure establishes a relation between the sliding surface of the plant and the zero learning-error level of the controller. The adopted form of the adjustment mechanism ensures that the parameters of the controller evolve boundedly and that the sliding surface is reached in finite time. The simulations presented confirm the claims of the approach. In Table 7.1, the method is evaluated for several comparison criteria, which are Reaching Mode Performance (RMP), Sliding Mode Performance (SMP), Bounded Parameter Evolution (BPE) and Computational Complexity (CC).

Table 7.1. Overall assessment of the single-term Lyapunov function based parameter tuning strategy

	ADALINE	**GRBFNN**	**SFS**	**ANFIS**	**FNN**
RMP	Adequate	Good	Very Good	Very Good	Not Applicable
SMP	Good	Poor	Very Good	Very Good	Not Applicable
BPE	Guaranteed	Guaranteed	Guaranteed	Guaranteed	Not Applicable
CC (flops)	25	117	218	312	Not Applicable

As can be inferred from the above table, ADALINE structure offers the computationally simplest solution, however, SFS and ANFIS structures exhibit very good performance in terms of tracking precision. The performance of the GRBFNN structure during the sliding mode is poor because of the deviations around the sliding line. Therefore, for control systems having fast microcomputers or DSP hardware, the use of SFS and ANFIS structures is recommended.

The second method of parameter tuning is based on a two-term Lyapunov function. Ensuring the negative definiteness of the time derivative of the adopted form of this function leads to a parameter adjustment law, which anticipates an appropriate path to the target. The sliding surface of the first method is augmented with the direction vector, which constitutes the basis of the widely used error backpropagation based training strategy, and the extracted form of the adjustment mechanism enforces the parameters to values which maintain the motion in the vicinity of the sliding surface. The simulation results presented confirm the achievement of the objectives of the approach. In contrast to the first method, the bounded parameter evolution is not guaranteed in the second method. An assessment of the second method is given in Table 7.2.

Table 7.2. Overall assessment of the two-term Lyapunov function based parameter tuning strategy

	ADALINE	**GRBFNN**	**SFS**	**ANFIS**	**FNN**
RMP	Good	Poor	Very Good	Very Good	Highly Unsatisfactory
SMP	Good	Highly Unsatisfactory	Good	Good	Highly Unsatisfactory
BPE	Not Guaranteed	Not Guaranteed	Not Guaranteed	Not Guaranteed	Not Guaranteed
CC (flops)	198	2701	2758	7116	2872

In the case of the second method for parameter tuning, the computational requirement of each structure is more than the corresponding entry given for the first method. This fact is intimately relevant to the matrix inversion at each control period. A comparison of the SFS and ANFIS structures suggests that the SFS structure is computationally simpler than the ANFIS structure. As shown in the comparison table, the tracking performance that can be achieved with GRBFNN and FNN is far from satisfactory.

Lastly, the problem of learning rate selection has been addressed. The well-known heuristics of the conventional approaches suggest the selection of a learning rate between the interval (0,1). However, the analysis presented in the sixth section shows that if the task is well defined, an equivalence between the single-term Lyapunov function based approach and the standard gradient based approaches can be established by appropriately setting the learning rate sequence. Because of the high computational cost, the derivations presented in the sixth section are simulated only for the ADALINE type controller structure, nevertheless the formulation for the other structures is included. The results obtained stipulate that the learning rate can assume values outside the interval (0,1). In Table 7.3, a comparison among the three different forms of the parameter update rule and the properties of the extracted learning rate formulations are given. In this table, the abbreviations LR and HF stand for the Learning Rate and High Frequency respectively, and the variable m denotes the number of inputs of the ADALINE structure.

Table 7.3. Overall assessment of the learning rate extraction approaches for ADALINE controller

	First Method Eqn. (6.9)	**Second Method Eqn. (6.17)**	**Third Method Eqn. (6.25)**
LR Dimension	$(m+1)\times(m+1)$	$(m+1)\times(m+1)$	1x1
LR HF Spectral Components	High Magnitude	Medium Magnitude	Low Magnitude
CC (flops)	112	106	23

As indicated by the results, the simplest method of constructing an appropriate learning rate sequence is the third method, by the use of which the high frequency deviations in the learning rate sequence are kept at admissibly low magnitudes and the quantity is a scalar. It should be clear that the computational complexity depends on the value of the variable m, the increase of which will clearly demonstrate the efficiency of the third approach.

The results presented and discussed throughout the thesis demonstrate that the use of VSS theory in the training of computationally intelligent systems introduces the desired properties of the VSS approach into the training process, so that a robust training of a computationally intelligent system is achieved. The analysis presented has also demonstrated that the task could be transformed into the selection of an appropriate learning rate sequence for achieving the zero learning-error level on the output of an intelligent controller with a given parameter adjustment law. The results presented confirm the applicability of the approaches in control systems.

REFERENCES

1. Haykin, S., *Neural Networks*, Macmillan College Printing Company, New Jersey, 1994.

2. Bishop, C. M., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

3. Jang, J.-S. R., C.-T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing*, PTR Prentice-Hall, 1997.

4. Wang, L., *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*, PTR Prentice-Hall, 1994.

5. Wang, L., *A Course in Fuzzy Systems and Control*, PTR Prentice-Hall, 1997.

6. Hornik, K., "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks*, Vol. 2, pp. 359-366, 1989.

7. Funahashi, K., "On the Approximate Realization of Continuous Mappings by Neural Networks", *Neural Networks*, Vol. 2, pp. 183-192, 1989.

8. Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function", *Mathematics of Control, Signals, and Systems*, Vol. 2, pp. 303-314, 1989.

9. Hung, J. Y., W. Gao and J. C. Hung, "Variable Structure Control: A Survey", *IEEE Transactions on Industrial Electronics*, Vol. 40, No. 1, pp. 2-22, February 1993.

10. Emelyanov, S. V., *Variable Structure Control Systems*, Moscow, Nauka, 1967.

11. Erbatur, K., O. Kaynak, A .Sabanovic, and I. Rudas, "Fuzzy Adaptive Sliding Mode Control of a Direct Drive Robot", *Robotics and Autonomous Systems*, Vol. 19, No. 2, pp. 215-227, December 1996.

12. Y. Byungkook and W. Ham, "Adaptive Fuzzy Sliding Mode Control of Nonlinear Systems", *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 2, pp. 315-321, 1998.

13. Ertugrul M. and O. Kaynak, "Neuro Sliding Mode Control of Robotic Manipulators", *Mechatronics*, Vol. 10, No. 1-2, pp. 243-267, 2000.

14. Ertugrul M. and O. Kaynak, "Neural Computation of the Equivalent Control in Sliding Mode For Robot Trajectory Control Applications", *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 2042-2047, May 1998

15. Gao, W. and J. C. Hung, "Variable Structure Control of Nonlinear Systems: A New Approach", *IEEE Transactions on Industrial Electronics*, Vol. 40, No. 1, pp. 45-55, February 1993.

16. Kaynak, O., F. Harashima, and H. Hashimoto, "Variable Structure Systems Theory, as Applied to Sub-time Optimal Position Control with an Invariant Trajectory", *Trans. IEE of Japan*, Sec. E, Vol. 104, No.3/4, pp. 47-52, 1984.

17. Bekiroğlu, N., *Adaptive Sliding Surface Design for Sliding Mode Control Systems*, Ph.D. Thesis, Boğaziçi University, 1996.

18. Young, K. D., V. I. Utkin, and U. Ozguner, "A Control Engineer's Guide to Sliding Mode Control", *IEEE Transactions on Control Systems Technology*, Vol. 7, No. 3, pp. 328-342, May 1999.

19. Misawa, E. A., "Discrete-Time Sliding Mode Control for Nonlinear Systems With Unmatched Uncertainties and Uncertain Control Vector", *Transactions of the ASME,*

Journal of Dynamic Systems, Measurement, and Control, Vol. 119, pp. 503-512, September 1997.

20. Misawa, E. A., "Discrete-Time Sliding Mode Control: The Linear Case", *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, Vol. 119, pp. 819-821, December 1997.

21. Kaynak, O., and A. Denker, "Discrete-Time Sliding Mode Control in the Presence of System Uncertainty", *International Journal of Control*, Vol. 57, No.5, pp. 1177-1189, 1993.

22. Efe, M. O. and O. Kaynak, "Stabilizing and Robustifying the Learning Mechanisms of Artificial Neural Networks in Control Engineering Applications", *International Journal of Intelligent Systems*, Vol.15, No.5, pp. 365-388, May 2000.

23. Efe, M. O. and O. Kaynak, "A Novel Optimization Procedure for Training of Fuzzy Inference Systems By Combining Variable Structure Systems Technique and Levenberg-Marquardt Algorithm", to appear in *Fuzzy Sets and Systems*.

24. Efe, M. O., O. Kaynak and B. M. Wilamowski, "Stable Training of Computationally Intelligent Systems By Using Variable Structure Systems Technique", *IEEE Transactions on Industrial Electronics*, Vol. 47, No. 2, pp. 487-496, April 2000.

25. Efe, M. O., A. M. Fiskiran and O. Kaynak, "Derivation of a Parameter Stabilizing Training Criterion for Adaptive Neuro-Fuzzy Inference Systems in Motion Control", to appear in *International Journal of Systems Science*, 2000.

26. Sira-Ramirez, H. and E. Colina-Morles, "A Sliding Mode Strategy for Adaptive Learning in Adalines", *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. 42, No.12, pp. 1001-1012, December 1995.

27. Yu, X., M. Zhihong and S. M. M. Rahman, "Adaptive Sliding Mode Approach for Learning in a Feedforward Neural Network", *Neural Computing & Applications*, Vol. 7, pp. 289-294, 1998.

28. Parma, G. G., B. R. Menezes and A. P. Braga, "Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks", *Electronics Letters*, Vol. 34, No. 1, pp. 97-98, January 1998.

29. Kaynak, O., "The Age of Mechatronics", (Guest Editorial), *IEEE Transactions on Industrial Electronics*, Vol.43, No.1, pp. 2-3, 1996.

30. Kirk, D. E., *Optimal Control Theory*, PTR Prentice-Hall, New Jersey, 1970.

31. Ogata, K., *Modern Control Engineering*, PTR Prentice-Hall, New Jersey, 1997.

32. Doyle, J. C., B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*, Macmillan Publishing Company, New York, 1992.

33. Söderström, T., *Discrete-time Stochastic Systems*, PTR Prentice-Hall, London, 1994.

34. Åström, K. J. and B. Wittenmark, *Computer Controlled Systems*, PTR Prentice-Hall, New Jersey, 1997.

35. Zadeh, L. A., "Fuzzy Sets", *Information and Control*, No. 8, pp. 338-353, 1965.

36. Widrow, B. and M. E. Hoff, Adaptive Switching Circuits, in *IRE WESCON Convention Record*, pp. 96-104, New York, 1960.

37. Gupta, M. M. and D. H. Rao, "Dynamical Neural Units with Applications to the Control of Unknown Nonlinear Systems", *Journal of Intelligent and Fuzzy Systems*, Vol. 1, No. 1, pp. 73-92, 1993.

38. Efe, M. O. and O. Kaynak, "A Comparative Study of Soft Computing Methodologies in Identification of Robotic Manipulators", *Robotics and Autonomous Systems,* Vol. 30, No. 3, pp. 221-230, 2000.

39. Efe, M. O., and O. Kaynak, "A Comparative Study of Neural Network Structures in Identification of Nonlinear Systems", *Mechatronics*, Vol. 9, No. 3, pp. 287-300, 1999.

40. Wang, Y-J. and C-T. Lin, "Runge-Kutta Neural Network for Identification of Dynamical Systems in High Accuracy", *IEEE Transactions on Neural Networks*, Vol. 9., No. 2, pp. 294-307, March 1998.

41. Takagi, T., and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 15, No. 1, pp. 116-132, January 1985.

42. Zalzala, A. M. S. and A. S. Morris, *Neural Networks for Robotic Control*, Ellis Horwood, United Kingdom, 1996.

43. King, R. E., *Computational Intelligence in Control Engineering*, Marcel Dekker Inc., New York, 1999.

44. Teshnehlab, M. and K. Watanabe, *Intelligent Control Based on Flexible Neural Networks*, Kluwer Academic Publishers, The Netherlands, 1999.

45. Ng, G. W., *Application of Neural Networks to Adaptive Control of Nonlinear Systems*, John Wiley & Sons Inc., United Kingdom, 1997.

46. Tsoukalas, L. H. and R. E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, John Wiley & Sons Inc., New York, 1997.

47. Narendra, K. S. and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, pp. 4-27, 1990.

48. Barto, A. G., "Connectionist Learning for Control", in *Neural Networks for Control*, Eds. Miller, T., Sutton, R. S. and Werbos, P. J., MIT Press, 1991.

49. Kawato, M., Y. Uno, M. Isobe, and R. Suzuki, "Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics", *IEEE Control Systems Magazine*, pp. 8-16, April 1988.

50. Psaltis, D., A. Sideris, and A. A. Yamamura, "A Multilayered Neural Network Controller", *IEEE Control Systems Magazine*, pp. 17-21, April 1988.

51. Guez, A., J. L. Eilbert, and M. Kam, "Neural Network Architecture for Control", *IEEE Control Systems Magazine*, pp. 22-25, April 1988.

52. Sanner, R. M. and J.-J. E. Slotine, "Gaussian Networks for Direct Adaptive Control", *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, pp. 837-863, November 1992.

53. Yen, J. and R. Langari, *Fuzzy Logic*, PTR Prentice-Hall, New Jersey, 1999.

54. Passino, K. M. and S. Yurkovich, *Fuzzy Control*, Addison-Wesley, California, 1998.

55. Berenji, H., "Fuzzy and Neural Control", in P. J. Antsaklis, and K. M. Passino (Eds.), *An Introduction to Intelligent and Autonomous Control*, pp. 215-236, Kluwer Academic Publishers, The Netherlands, 1992.

56. Ishibuchi, H. and H. Tanaka, "Fuzzy Neural Networks with Fuzzy Weights and Fuzzy Biases", in P. K. Simpson (Eds.), *Neural Networks, Theory Technology and Applications*, pp. 155-160, IEEE Technology Update Series, New Jersey, 1996.

57. Machado, R. J. and A. F. Rocha, "Inference, Inquiry and Explanation in Expert Systems by means of Fuzzy Neural Networks", in P. K. Simpson (Eds.), *Neural Networks, Theory Technology and Applications*, pp. 161-166, IEEE Technology Update Series, New Jersey, 1996.

58. Nauck, D. and R. Kruse, "A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation", in P. K. Simpson (Eds.), *Neural Networks, Theory Technology and Applications*, pp. 167-172, IEEE Technology Update Series, New Jersey, 1996.

59. Efe, M. O., A. M. Fiskiran, O. Kaynak, and I. J. Rudas, "VSS Theory Based Training of a Fuzzy Motion Control System", *Proceedings of The Eighth International Fuzzy Systems Association World Congress* (IFSA'99), Taipei, Taiwan R.O.C., August 17-20, 1999, pp. 199-203, Taiwan R.O.C., 1999.

60. Efe, M. O. and O. Kaynak, "Neuro-Fuzzy Approaches for Identification and Control of Nonlinear Systems", *International Symposium on Industrial Electronics* (Tutorial, ISIE'99), July 12-16, 1999, Bled, Slovenia, pp. TU2-TU11, Bled, 1999.

61. Rumelhart, D. E., G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation", in D. E. Rumelhart and J. L. McClelland, eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, pp. 318-362, MIT Press, Cambridge, M.A., 1986.

62. Hagan, M. T., M. B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, pp. 989-993, November 1994.

63. Hush, D. R. and B. G. Horne, "Progress in Supervised Neural Networks", *IEEE Signal Processing Magazine*, pp. 8-39, January 1993.

64. Hunt, K. J., D. Sbarbaro, R. Zbikowski and P. J. Gawthrop, "Neural Networks for Control Systems - A Survey", *Automatica*, Vol. 28, No. 6, pp. 1083-1112, 1992.

65. Efe, M. O., O. Kaynak, and Imre J. Rudas, "Increasing the Noise Rejection Capability of Neural Networks in Identification of Nonlinear Systems", *International Conference on Mechatronic Technology (ICMT'99)*, October 21-23, Pusan, Korea, pp. 363-368, 1999.

66. Efe, M. O., O. Kaynak, and I. J. Rudas, "A Novel Computationally Intelligent Architecture for Identification and Control of Nonlinear Systems", *IEEE International Conference on Robotics and Automation (ICRA'99)*, May 10-15, Detroit, Michigan, U.S.A., pp. 2073-2077, 1999.

67. Slotine, J.-J. E. and W. Li, *Applied Nonlinear Control*, Prentice-Hall, New Jersey, 1991.

68. Edwards, C. and S. K. Spurgeon, *Sliding Mode Control - Theory and Applications*, Taylor & Francis, United Kingdom, 1998.

69. Utkin, V. I., *Sliding Modes in Control Optimization*, Springer Verlag, New York, 1992.

70. Denker, A. and O. Kaynak, "Applications of VSC in Motion Control Systems", *in Variable Structure and Lyapunov Control*, Eds. A. S. I. Zinober, Springer Verlag, London, 1994.

71. Sabanovic, A., N. Sabanovic, K. Jezernik and K. Wada, "Chattering Free Sliding Modes", *Proceedings of the Third Workshop on Variable Structure Systems and Lyapunov Design*, pp. 143-148, Napoly, Italy, September 1994.

72. Erbatur, K., O. Kaynak and A. Sabanovic, "A Novel Variable Structure Control as Applied to the Control of a Direct Drive Manipulator", *Proceedings of the European Control Conference*, pp. 196-201, Rome, 1995.

73. Erbatur, K., O. Kaynak and A. Sabanovic, "Robot Trajectory Control in Cartesian Space via Sliding Modes", *Proceedings of the IEEE Industrial Electronics Conference*, Vol. 2, pp. 766-771, 1996.

74. Erbatur, K., O. Kaynak and A. Sabanovic, "An Experimental Evaluation of Chattering Free Sliding Mode Control as Applied to the Control of a Direct Drive Manipulator",

Proceedings of the Industrial Power Electronics Conference, pp. 226-231, Yokohama, 1995.

75. Erbatur, K., O. Kaynak and A. Sabanovic, "A Study on Robustness Property of Sliding Mode Controllers: A Novel Design and Experimental Investigations", *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 5, pp. 1012-1018, 1999.

76. Sabanovic, A., K. Jezernik and O. Kaynak, "Chattering Free Sliding Modes in Robotic Manipulators Control", *Mechatronics*, Vol. 1, No. 1, Ankara, 1994.

77. Erbatur, K., *A Study of Indirect Fuzzy Sliding Mode Control for Robotic Manipulators*, Ph.D. Thesis, Boğaziçi University, 2000.

78. Ertuğrul, M., *Neuro-Sliding Mode Control of Robotic Manipulators*, Ph.D. Thesis, Boğaziçi University, 1999.

79. Kaynak, O., K. Erbatur and M. Ertuğrul, "The Fusion of Computational Intelligence Methodologies in Sliding Mode Control", to appear in *IEEE Transactions on Industrial Electronics*, 2000.

80. Ha, Q. P. "Robust Sliding Mode Controller with Fuzzy Tuning", *Electronics Letters*, Vol. 32, No. 17, pp. 1626-1628, 1996.

81. Ha, Q. P. "Sliding Mode Performance Enhancement with Fuzzy Tuning", *Electronics Letters*, Vol. 33, No. 16, pp. 1421-1423, 1997.

82. Tong, S., J. Tang, and T. Wang, "Fuzzy Adaptive Control of Multivariable Nonlinear Systems", *Fuzzy Sets and Systems*, Vol. 111, No. 2, pp. 153-167, April 2000.

83. Kim, Y.-T. and Z. Bien, "Robust Self-Learning Fuzzy Controller Design for a Class of Nonlinear MIMO Plants", *Fuzzy Sets and Systems*, Vol. 111, No. 2, pp. 117-135, April 2000.

84. Direct Drive Manipulator R&D Package User Guide, Integrated Motions Incorporated, 704 Gillman Street, Berkeley, California 94710, U.S.A., 1992.

85. Efe, M. O. and O. Kaynak, "Establishment of a Sliding Mode in a Nonlinear System by Tuning the Parameters of a Fuzzy Controller", *IEEE International Conference on Systems, Man and Cybernetics (SMC-2000)*, October 8-11, Nashville, Tennessee, U.S.A. 2000, (accepted for publication).

86. Liu, Z.-Q. and F. Yan, "Fuzzy Neural Network in Case-Based Diagnostic System", *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 2, pp. 209-222, 1997.

87. Kim, E., M. Park, S. Li, and M. Park, "A New Approach to Fuzzy Modeling", *IEEE Transactions on Fuzzy Systems*, Vol. 5, No.3, pp. 328-337, 1997.

88. Lin C.-J. and C.-T. Lin, "An ART-Based Fuzzy Adaptive Learning Control Network", *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 4, pp. 477-496, 1997.

89. Juang, C.-F. and C.-T. Lin, "An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications", *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 1, pp. 12-32, 1998.

90. Halgamuge, S. K., "A Trainable Transparent Universal Approximator for Defuzzification in Mamdani-Type Neuro-Fuzzy Controller", *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 2, pp. 304-314, 1998.

91. Lai, J.-H. and C.-T. Lin, "Application of Neural Fuzzy Network to Pyrometer Correction and Temperature Control in Rapid Thermal Processing", *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 2, pp. 160-175, 1999.

92. Maeda, Y. and R. J. P. De Figueiredo, "Learning Rules for Neuro-Controller via Simultaneous Perturbation", *IEEE Transactions on Neural Networks*, Vol. 8, No. 5, pp. 1119-1130, 1997.

93. Moon, S. and J.-N. Hwang, "Robust Speech Recognition Based on Joint Model and Feature Space Optimization of Hidden Markov Models", *IEEE Transactions on Neural Networks*, Vol. 8, No. 2, pp. 194-204, 1997.

94. Efe, M. O., E. Abadoglu and O. Kaynak, "A Novel Analysis and Design of a Neural Network Assisted Nonlinear Controller for a Bioreactor", *International Journal of Robust and Nonlinear Control*, Vol. 9, No.11, pp. 799-815, September 1999.

95. Efe, M. O., O. Kaynak, and B. M. Wilamowski, "Training of Fuzzy Inference Systems by Combining Variable Structure Systems Technique and Levenberg-Marquardt Algorithm", *Proceedings of The 25th Annual Conference of the IEEE Industrial Electronics Society (IECON'99)*, November 29-December 3, San Jose, U.S.A., Vol. 2, pp. 514-519, 1999.

96. Efe, M. O. and O. Kaynak, "A Hybrid Training Procedure for Artificial Neural Networks Leading To Parametric Stability and Cost Minimization", *The 7th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'99)*, October 18-22, Barcelona, Catalonia, Spain, Vol. 1, pp. 71-77, 1999.

97. Efe, M. O., O. H. Dagci, and O. Kaynak, " Fuzzy Control of a 2-DOF Direct Drive Robot Arm by Using a Parameterized T-Norm", *The Eighth Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN'99)*, June 23-25, Istanbul, Turkey, pp. 210-218, 1999.

98. Efe, M. O., O. Kaynak and F. Keresteciogl u, "Intelligent Signal Estimation Using Cosine Neural Networks with Variable Structure Systems Based Training Procedure", *The 2nd International Conference on Recent Advances in Mechatronics (ICRAM'99)*, May 24-26, Istanbul, Turkey, pp. 57-61, 1999.

99. Efe, M. O. and O. Kaynak, "A Comparative Study of Soft Computing Methodologies in Identification of Robotic Manipulators", *Proceedings of the 3rd International Conference on Advanced Mechatronics (ICAM'98)*, August 3-6, Okayama, Japan, Vol. 1, pp. 21-26, 1998.

100.Efe, M. O. and O. Kaynak, "Identification and Control of a Nonlinear Bioreactor Plant Using Classical and Dynamical Neural Networks", *Proceedings of the International Symposium on Industrial Electronics (ISIE'97)*, July 7-11, Guimaraes, Portugal, Vol. 3, pp. 1211-1215, 1997.

101.Efe, M. O. and O. Kaynak, "Stabilizing and Robustifying the Error Backpropagation Method in Neurocontrol Applications", *IEEE International Conference on Robotics and Automation (ICRA'2000)*, April 24-28, San Francisco, U.S.A. Vol. 2, pp. 1882-1887, 2000.

102.Efe, M. O. and O. Kaynak, "Synthesis of a Robust Neurocontroller in the Face of Strong External Disturbances", *IEEE International Conference on Control Applications*, September 25-27, Anchorage, Alaska, U.S.A., 2000, (accepted for publication).

103.Efe, M. O. and O. Kaynak, "A Stabilizing and Robustifying Training Scheme for Artificial Neural Networks Used in Control of Complex Systems", *1st IFAC-Conference on Mechatronic Systems (MECHATRONICS 2000)*, September 18-20, Darmstadt, Germany, 2000, (accepted for publication).

104.Efe, M. O. and Kaynak, O., "On Stabilization of Gradient Based Training Strategies for Computationally Intelligent Systems", *IEEE Transactions on Fuzzy Systems* (accepted for publication).