VARIABLE STRUCTURE SYSTEMS BASED ONLINE LEARNING ALGORITHMS FOR TYPE-1 AND TYPE-2 FUZZY NEURAL NETWORKS

by

Özkan Çiğdem

B.S, Electrical and Electronics Engineering, K.T.U., 2008

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

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2011

ACKNOWLEDGEMENTS

I would like to express my heartfelt appreciation to my supervisor Prof. Okyay Kaynak who gave me the opportunity to work with him. I have learned a lot from him over the last two years, and he has had a great influence on me. This thesis could not have come into being without his guidance. I would like to extend my gratitude to Assoc. Prof. Yağmur Denizhan and Prof. Fikret Gürgen for serving on my committee and their academic guidance.

I thank my colleague Dr. Erdal Kayacan who has provided me with irreplaceable guidance and friendship during the composition of my thesis. Moreover, I am also grateful to my other colleagues Dr. Mojtaba Ahmadieh Khanesar, Yeşim Öniz, Dr. Fulya Kunter, Çisel Aras, and Hakan Karaoğuz for their help.

Additionally, I would like to thank Boğaziçi University and Ecole Polytechnique Federale de Lausanne (EPFL) for giving me the opportunity to reach this point in my academic career, and conduct this research.

I would also like to thank Kerri Barbara Schultz, who has done a lot for me in both my life and my thesis, and to my precious nephew Toprak Çağan, who has caused nothing but joy in my life.

Last but not least, I would like to express my utter indebtedness to my family, who has given me endless love and guidance, throughout the entire course of my academic career. There are no words with which I could properly express my appreciation of their steadfast support.

ABSTRACT

VARIABLE STRUCTURE SYSTEMS BASED ONLINE LEARNING ALGORITHMS FOR TYPE-1 AND TYPE-2 FUZZY NEURAL NETWORKS

Type-2 fuzzy logic systems are proposed in the literature as an alternative to type-1 fuzzy logic systems because of their abilities to more effectively model rule uncertainties. This thesis extends the idea of using the sliding mode control theory in the training of type-1 fuzzy neural networks to type-2 fuzzy neural networks. In the approach, instead of trying to minimize an error function, the weights of the network are tuned by the proposed algorithm in a way that the error is enforced to satisfy a stable equation. The parameter update rules are derived, and the convergence of the weights is proved by Lyapunov stability method. The performance of the proposed learning algorithms is tested for both type-1 and type-2 fuzzy neural networks on a real-time laboratory servo system. Simulation and experimental results indicate that the proposed type-2 fuzzy neural network with the proposed learning algorithm is more robust to uncertainties and computationally effective than its type-1 fuzzy counterpart.

ÖZET

TİP-1 VE TİP-2 NÜRO BULANIK SİSTEM İÇİN DEĞİŞKEN YAPILI SİSTEMLER TABANLI ÇEVRİMİÇİ ÖĞRENME ALGORİTMALARI

Tip-2 bulanık mantık sistemleri, tip-1 bulanık mantık sistemlerine göre belirsizlikleri daha etkin bir şekilde modelleyebildiğinden bu sistemlere alternatif bir sistem olarak önerilmiştir. Bu tezde, tip-1 nüro-bulanık ağların eğitilmesinde kullanılan kayma kipli kontrol kuramı fikri, tip-2 nüro-bulanık ağların eğitilmesi için geliştirilmiştir. Önerilen yaklaşımda, hata fonksiyonunu sıfırlamak yerine, ağın öğrenme parametreleri öyle ayarlanacaktir ki, hata kararlı bir denklemi sağlamaya zorlanacaktır. Parametre güncelleme kuralları elde edilmiş ve öğrenme algoritmasının kararlılığı Lyapunov kararlılık metodu ile kanıtlanmıştır. Önerilen öğrenme algoritmasının performansı servo sistem olan gerçek zamanlı bir laboratuvar düzeneğinde test edilmiştir. Benzetim ve gerçek zamanlı deney sonuçlarından, tip-2 nüro-bulanık ağların, tip-1 nüro-bulanık ağlara oranla, belirsizliklere karşı daha gürbüz olduğu ve hesap açısından daha etkin olduğu görülmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	The linguistic value defined by fuzzy set on universes of discourse X
$\widetilde{A_{in}}$	Type-2 antecedent fuzzy sets
B	The linguistic value defined by fuzzy set on universes of discourse Y
B_{in}	Type-1 consequent fuzzy sets
c	The mean value of a Gaussian membership function
$\overline{c}$	The mean value of a Gaussian upper membership function
$\underline{c}$	The mean value of a Gaussian lower membership function
$C\Phi$	Back emf constant
e	The feedback error
$\dot{e}$	The change rate of feedback error
E	Induced electromotive force
f_{ij}	The weight coefficient
I_A	Armature current
J	Moment of inertia of the system
K_I	Integral parameter
K_M	Torque constant
K_P	Proportional parameter
L_A	Armature winding inductance
M	The torque produced by the motor
M_B	Acceleration torque
M_L	Load torque
R_A	Armature winding resistance
S_c	The zero adaptive learning error
S_p	The sliding line
T_A	Electrical time constant
T_M	Mechanical time constant
U_A	Armature terminal voltage

V_p	Lyapunov function
w_d	The output value of the proposed adaptive neuro system
w_d	The target value of the proposed adaptive neuro system
w_i	The firing strength
$\overline{W}_{ij}$	The output signal of the neuron ij
$\underline{W_{ij}}$	The lower output signals of the neuron ij
$\overline{W_{ij}}$	The upper output signals of the neuron ij
$\underline{\widetilde{W_{ij}}}$	The normalized values of the lower output signals of the neuron ij
$\widetilde{\overline{W_{ij}}}$	The normalized values of the upper output signals of the neuron ij
Y_i	The output of ith rule which is a type-1 fuzzy set
$alpha$	The design constant
χ	The constant determining the slope of the sliding surface
χ_A	Characterstic function of the X universe
$\mu_A(x)$	Type-1 membership function for a type-1 fuzzy set A
$\tilde{\mu}_{\widetilde{A_{in}}}(x)$	Type-2 membership function for a type-2 fuzzy set $\widetilde{A_{in}}$
σ	The standard deviation of a Gaussian membership function
$\overline{\sigma}$	The standard deviation of a Gaussian upper membership function
$\underline{\sigma}$	The standard deviation of a Gaussian lower membership function
τ	The input signal of the system to be controlled
τ_c	The output signal of PI controller
τ_n	The output signal of the fuzzy neural network
Ω	Speed of the rotor

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
FEL	Feedback-Error-Learning
FLC	Fuzzy Logic Controller
FLS	Fuzzy Logic System
FNN	Fuzzy Neural Network
FOU	Footprint of Uncertainty
GA	Genetic Algorithm
PI	Proportional-Integral
PD	Proportional-Derivative
SMC	Sliding Mode Control
TSK	Takagi-Sugeno-Kang
T1FLS	Type-1 Fuzzy Logic System
T2FLS	Type-2 Fuzzy Logic System
T1FLC	Type-1 Fuzzy Logic Controller
T2FLC	Type-2 Fuzzy Logic Controller
T1FNN	Type-1 Fuzzy Neural Network
T2FNN	Type-2 Fuzzy Neural Network
VSS	Variable Structure System

1. INTRODUCTION

1.1. Literature Review

In real-time industrial control systems uncertainty is an unavoidable issue. Generally, when measurements are taken, they include uncertainties. These inaccuracies can arise from a variety of sources in most controlled systems for example, due to, lack of modeling information and incorrect interpretation of obtained data. Just like uncertainties in a system, ambiguities in spoken language can cause misunderstandings leading to miscommunication in daily life [1]. For example, when people communicate, they rarely speak in strictly yes-no form, rather they use quantitative adjectives (such as very, completely or a little) in order to present a clearer explanation with regard to the amount to which a thing is a certain way.

Logic found its basis in the thinking of ancient philosophers. The philosopher Aristotle defined logic on a bivalent basis, believing that all things could be defined as either true or false. The problem with this type of thinking is that it leaves the world to be seen of black and white as opposed to color. Since interpretation of information provided is at the discretion of the person receiving the information, problems can come up. For instance, when one says “It is freezing”, the temperature might be subzero or just cold in her opinion. People can naturally deal with these kinds of imprecisions, however, it is not easy for a machine to cope with them. While expressing complex mathematical concepts, it is especially useful to give a more accurate view of how much something is one way rather than another, and to have a clearer view of potentially valuable information that would otherwise be vague. In order to overcome the ambiguities which typically occur in engineering applications, Zadeh introduced the concept of fuzzy sets in 1965 which are sets with uncertain amplitudes [2]. The concept of fuzzy sets is based on the degree of memberships rather than true or false.

After Zadeh published his first paper on fuzzy sets [2], he opened the door for other researchers to study type-1 fuzzy logic (T1FL) and he quickly amassed a large number of followers. Another critical point for fuzzy logic came about in the year 1974, when Ebrahim Mamdani and S. Assilian used fuzzy logic to control a steam engine for the first time [3]. After this initial break-through experiment, fuzzy logic began to gain popularity as a method of control. In 1976, fuzzy logic had its first implementation in an industrial manner. It was used to aid in the operation of a cement kiln by two prominent Danish companies [4].

During the last few decades, T1FL has been applied in many industrial areas such as elevator drive systems, robotics, DC-DC converters [5–7]. In [5], since the proportional (P) and integral (I) values speed controller cannot generally be set to a large value due to its mechanical resonance, instead of conventional speed controller, a fuzzy logic controller (FLC) was used in elevator drive systems. Experimental results show that for high-performance elevator drive systems, the proposed FLC is better than the conventional PI controller in speed control. In [6], to accomplish a real-time and robust control performance in reactive manners, FLC is used to encode the behaviors for the quadruped walking robots which learn and execute soccer-playing behaviors. These experimental studies show the effectiveness of the controller in representing behavior of the robots. In [7], in order to smoothen the output power fluctuation of a variable-speed wind farm, a FLC is used as a reference adjuster to control the DC-DC converter. Simulation studies demonstrate that using FLC enhances the control ability of the overall system, and reduces the cost of the energy capacitor system, yet keeps the size of the energy storage system small.

When a system has a large amount of uncertainties, type-1 fuzzy logic systems (T1FLSs) may not be capable of achieving the desired level of performance with a reasonable complexity of structure [8]. In such cases, the use of type-2 fuzzy sets were introduced as a preferable approach in the literature by Zadeh in 1975 as an extension of type-1 fuzzy sets. A type-2 fuzzy set has fuzzy membership grades which means the membership for each element of this set is fuzzy set in $[0, 1]$ [9]. In the literature, there are many applications of type-2 fuzzy logic systems (T2FLS) [10–15]. In [10], a type-2 self-organizing fuzzy neural system and its hardware implementation is proposed. It is reported that using interval type-2 fuzzy sets in that structure enables the overall system to be more robust than using type-1 fuzzy systems (T1FLSs). The VLSI implementation of T2FLSs is discussed in literature and it is shown that the inference speed can be sufficiently high for real-time applications [11]. In [12], T2FLSs are applied to real-time mobile robots for indoor and outdoor environments. The real-time implementations of this study demonstrate that a traditional type-1 fuzzy logic controller (T1FLC) cannot handle the uncertainties in the system effectively, and a type-2 fuzzy logic controller (T2FLC) using type-2 fuzzy sets gives a better performance. Furthermore, with the latter approach, the number of rules to be used may be reduced (it should be noted that this may not mean a corresponding decrease in the parameters to be updated). In [13], a novel inference mechanism is proposed for an interval type-2 Takagi-Sugeno-Kang (TSK) FLC system when antecedents are type-2 fuzzy sets and consequents are crisp numbers (A2-C0). Case studies reveal that the proposed inference engine clearly outperforms its type-1 TSK counterpart. An interval T2FLC architecture is proposed to resolve nonlinear control problems of vehicle active suspension systems in [14]. Simulation results show that the proposed control algorithm not only handles the system uncertainty effectively as well as improves control performance but also saves actuator energy. In [15], a novel design methodology of interval type-2 TSK FLCs for modular and reconfigurable robot manipulators with uncertain dynamic parameters is presented. The results presented show that the developed controller can outperform some well-known linear and nonlinear controllers in terms of tracking performances for different configurations.

Since fuzzy neural network (FNN) brings together the advantages of fuzzy logic and artificial neural network (ANN), it has become a topic of interest for scientists and used in this thesis [16]. It incorporates the advantages of neural networks since they have low-level learning as well as computational power and the advantage of fuzzy logic since it has high-level interpretation of human knowledge. For tuning the parameters of a FNN, there are a number of methods used in the literature such as gradient-descent-based algorithm and genetic algorithms (GAs). While these methods are widely used for tuning the parameters of ANNs and FNNs, they may show some drawbacks in online learning. For instance, although the gradient-descent method is easy to implement, it includes partial derivatives of the overall system error with respect to the weights of the network and the parameters of the membership functions such as center and/or sigma values. As a result, it may be very difficult to make these computations and the convergence speed may be slow especially when the search space is complex. Additionally, the tuning process can easily be trapped at a local minima [17]. To overcome these drawbacks, evolutionary approaches have been proposed [18]. Although GAs do not include partial derivatives, the computational burden may be rather high and its stability is questionable. Furthermore, the optimal values for the stochastic operators are difficult to derive and therefore, its use is very difficult in real-time applications . In order to handle these difficulties, variable structure systems (VSSs) theory based algorithms are proposed [19–21] for parameter update rules of ANNs and FNNs.

Sliding mode control (SMC) is a preferred option among other techniques in the literature because it guarantees the robustness of the system in the case of external disturbances and uncertainties. The main idea behind this control scheme is to restrict the motion of the system in a plane referred to as the *sliding surface*, where a predefined function of the error is zero [22]. SMC theory not only makes the overall system more robust but also creates faster convergence than the traditional learning techniques in online tuning of ANNs and FNNs [23]. There are various studies in the literature that aim to use the robustness property of SMC in the learning process of ANNs and FNNs [24].

In this thesis, feedback-error-learning (FEL) is used as the control structure for both type-1 fuzzy neural network (T1FNN) and type-2 fuzzy neural network (T2FNN). FEL method was first proposed by Kawato in 1988 for robot control in which a neural network works in parallel with a proportional-derivative (PD) controller [25]. In [25], FEL is defined as a learning scheme in which the feedback signal generally heads toward zero due to the fact that instead of the teaching signal or the desired output, the feedback torque is used as the error signal also because both the control and learning processes are done simultaneously.

1.2. Motivation and Scope of the Thesis

In this thesis, the output of a PI controller is used as a learning error signal to train both T1FNN and T2FNN. Even though this approach has been used with T1FNN before [26], through this study, the novel update rules are derived for T1FNN; what is more, this idea is extended to T2FNNs structures as well. More specifically, instead of trying to minimize an error function as is usually done with the gradient-based algorithms, the learning parameters are tuned by the proposed algorithm in a way that enforces the error to satisfy a stable equation. The contributions of this thesis to the existing literature are (i) to extend the parameter update rules of T1FNNs to T2FNNs by using SMC theory-based learning algorithm, and (ii) prove its stability in Lyapunov stability method.

This thesis is organized into six chapters. Chapter 1 starts with background information on T1FLS, T2FLS, SMC and FEL. The historical review and application examples of fuzzy logic and FEL are explained. The most commonly used training algorithms have been compared and the reasons of using SMC based learning algorithm have been given.

In Chapter 2, the concept and mathematical definitions of T1FLSs and T2FLSs are presented.

The basic information about T1FNN and T2FNN and the proposed FEL are shown in Chapter 3. The underlying idea and mathematical equations of adaptive fuzzy neuro scheme and the structure of FNN have been introduced.

An introduction to commonly used training methods and SMC is presented in Chapter 4. The parameter update rules for T1FNN and T2FNN are derived.

To illustrate the applicability and the efficacy of the proposed method, the control problem of a DC motor with linear and nonlinear load conditions is studied and the simulation results for T1FNN obtained are presented. The proposed algorithm is also tested on a real-time laboratory setup for both T1FNN and T2FNN.

Chapter 6 provides conclusion of this research.

2. FUZZY LOGIC THEORY

Incomplete or insufficient information in real-time industrial control applications is either due to the deficiency of modeling information or due to the fact that the right observation and control variables have not been employed. These factors create a difficulty in obtaining an accurate model. Even if the controlled system has an accurate model, there are many other uncertainties originating from the precision of the sensors, noise produced by the sensors, environmental conditions of the sensors and nonlinear characteristics of the actuators. In such cases, model-free approaches are generally preferred. The most common model-free approaches in literature are ANNs and FLSs.

Zadeh made a statement about fuzzy logic theory that *"fuzzy logic is a precise conceptual system of reasoning, deduction and computation in which the objects of discourse and analysis are, or are allowed to be, associated with imperfect information. Imperfect information is information which in one or more respects is imprecise, uncertain, incomplete, unreliable, vague or partially true"* [27].

Studies in literature show that FLSs can usually handle the uncertainties better when compared to ANNs [8]. There are two different types of approaches to FLSs design: T1FLSs and T2FLSs.

2.1. Type-1 Fuzzy Logic Systems

The core technique of fuzzy logic is focused on four main points [9].

(i) *fuzzy sets*: sets with smooth boundaries
(ii) *linguistic variables*: variables whose values are both qualitatively and quantitatively described by a fuzzy set
(iii) *possibility distributions*: constraints on the value of a linguistic variable imposed by assigning it a fuzzy set and
(iv) *fuzzy if-then rules*: a knowledge scheme for describing a functional mapping or a

logic formula that generalizes an implication in two valued logic.

2.1.1. Type-1 Fuzzy Sets and Membership Functions

There are many vague expressions on daily conversation from gossip such as "My best friend's girlfriend is so skinny" to a biologist's statement that "the number of fishes living in Atlantic Ocean is decreasing dramatically." Fuzzy sets can overcome such uncertain concepts which cannot be expressed by conventional set theory. To define such ambiguous expressions on conventional set, a threshold value should be set such as "the girls under 55 kg are skinny." Through this value, it is possible to make groups who are skinny and who are fat. Such conventional sets are called "crisp sets".

Definition 2.1. Let A represent a *crisp set* on the universe X. Its characteristic function χ_A can be defined by a mapping as follows:

$$\chi_A : X \to \{0,1\} \quad \text{as} \qquad \chi_A = \begin{cases} 1 & x \in X \\ 0 & x \notin X \end{cases} \tag{2.1}$$

Equation 2.1 indicates that if the element x belongs to A, χ_A is 1, and if it does not belong to A, χ_A is 0. Unlike the conventional set, a fuzzy set [2] expresses the degree to which an element belongs to a set. Therefore, the characteristic function of fuzzy set is allowed to have values between 0 and 1, which denotes the degree of membership of an element in a given set.

The methods of expressing fuzzy sets can be roughly divided into two as in the following definitions.

Definition 2.2. Let the universe X be $X = \{x_1, x_2, ..., x_n\}$. For *discrete expression (when the universe is finite)*, a fuzzy set A on X can be represented as follows:

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + ... + \mu_A(x_n)/x_n = \sum_{i=1}^{n} \mu_A(x_i)/x_i \tag{2.2}$$

Definition 2.3. For *continuous expression (when the universe X is an infinite set)*, a fuzzy set A on X can be represented as follows:

$$A = \int_x \mu_A(x_i)/x_i \tag{2.3}$$

When X denotes the universe of discourse, x represents an element of the universe, X and A denote a fuzzy set, it can be characterized by its membership function, $\mu_A(x)$ as follows:

$$\mu_A(x) : X \to [0, 1] \tag{2.4}$$

Membership function states that values assigned to the elements of the universal set, X, fall within a specified range. At the same time, it also represent the membership grade of these elements in fuzzy set A. In literature, there are three prevailingly used membership functions which are Gaussian, triangular and trapezoidal ones.

Definition 2.4. A *type-1 Gaussian membership function* is specified by two parameters $\{c, \sigma\}$:

$$Gaussian(x; c, \sigma) = \mu(x) = e^{-(x-c)^2/\sigma^2} \tag{2.5}$$

A Gaussian membership function is determined completely by c and σ; c represents the membership functions center and σ determines membership functions width as can be seen in Figure 2.1.

Definition 2.5. A *type-1 triangular membership function* is specified by three parameters $\{a, b, c\}$ where a is the left endpoint, b is the central point, and c is the right endpoint as follows:

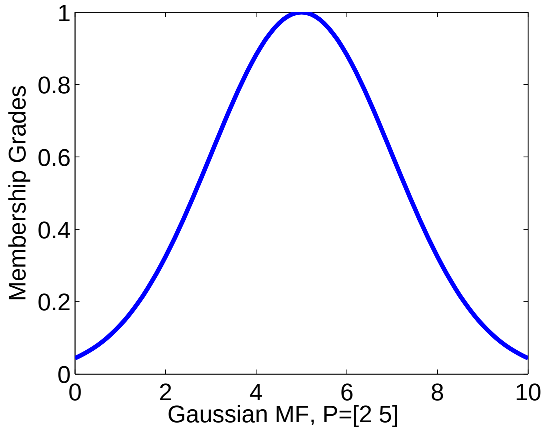


Figure 2.1. Gaussian membership function.

$$Triangle(x;a,b,c) = \mu(x) = \begin{cases} 0, & x \leq a. \\ (x-a)/(b-a), & a \leq x \leq b. \\ (c-x)/(c-b), & b \leq x \leq c. \\ 0, & c \leq x. \end{cases}$$

The parameters a, b, c (where $a < b < c$) determine the x coordinates of the three corners of the underlying triangular membership function as it can be seen in Figure 2.2.

Definition 2.6. A *type-1 trapezoidal membership function* is specified by four parameters $\{a, b, c, d\}$ as follows:

$$Trapezoid(x;a,b,c,d) = \mu(x) = \begin{cases} 0, & x \leq a. \\ (x-a)/(b-a), & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ (d-x)/(d-c), & c \leq x \leq d. \\ 0, & d \leq x. \end{cases}$$

The parameters a, b, c, d (where $a < b \leq c < d$) determine the x coordinates of the four corners of the underlying trapezoidal membership function as it can be seen in Figure 2.3.

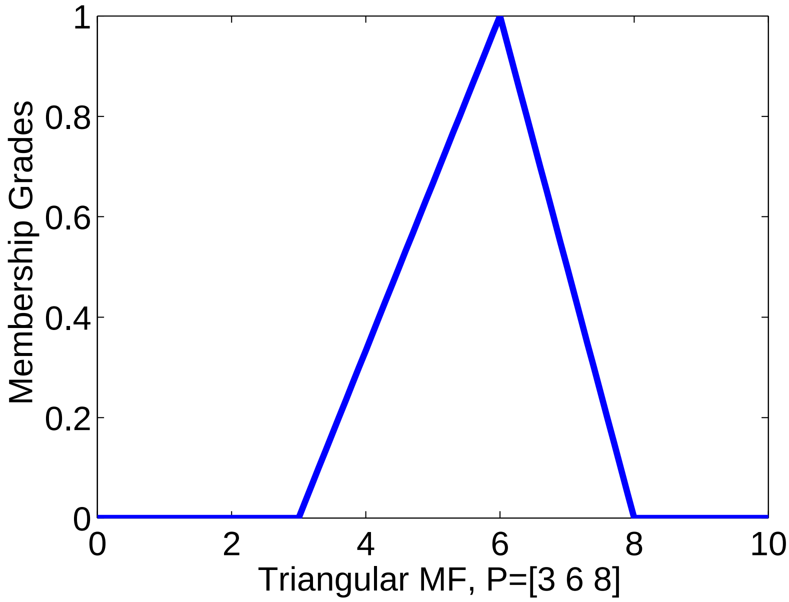


Figure 2.2. Triangular membership function.

2.1.2. If-Then Rules

Fuzzy sets and their operations are the subjects and verbs of fuzzy logic. *If-then* rule statements are used to formulate the conditional statements that comprise fuzzy logic.

In also daily language, there are many examples of *if-then* rule such as

- If the relative humidity is 100%, then it rains.
- If speed is slow, then pressure should be high.
- If a person's IQ is high, then the person is smart.

A fuzzy *if-then* rule is expressed in the form of:

$$\textbf{if } x \textit{ is } A \textbf{ then } y \textit{ is } B \tag{2.6}$$

where A and B are linguistic values defined by fuzzy sets on universes of discourse X and Y, respectively. In general, "x is A" is called "antecedent" or "premise" and "y is

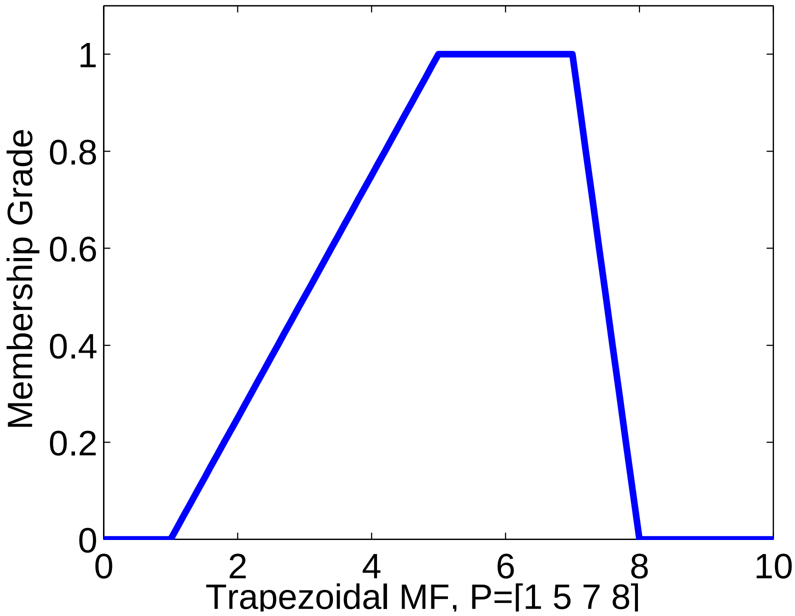


Figure 2.3. Trapezoidal membership function.

B" is called the "consequence".

2.1.3. Type-1 Fuzzy Logic System Scheme

A T1FLS consists of four components which are fuzzifier, rule base, inference engine and defuzzifier as it is seen in Figure 2.4. The fuzzification interface converts the input values such that they can be understood in the rule-base. In other words, crisp input values are transformed into the membership degrees for the fuzzy set antecedents. The knowledge base includes both rule base and data based. The rule base is made up of fuzzy rules. The data base provides information for fuzzification and defuzzification processes. It also stores membership functions used in fuzzy rules.

There are two commonly used types of fuzzy inference models which are Mamdani [28] and Takagi-Sugeno-Kang (TSK) ones [29, 30]. In [31], by using Mamdani fuzzy inference model, a steam engine and boiler combination was controlled through the linguistic control rules taken by experienced human operators.

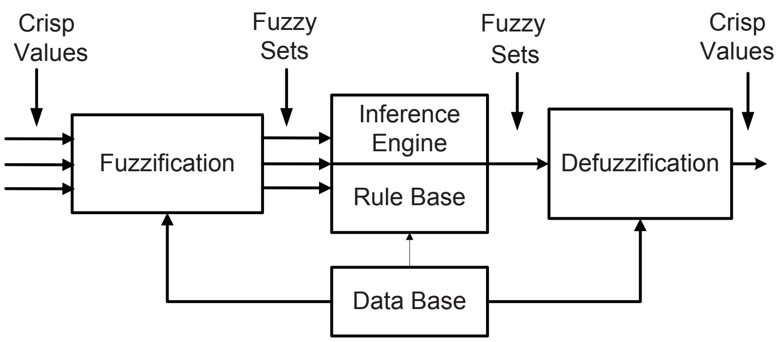


Figure 2.4. Type-1 fuzzy logic system.

TSK fuzzy model was proposed to develop a systematic approach to generating fuzzy rules from a given input-output data set. A general form of TSK fuzzy rule for two inputs is represented as follows:

$$\textbf{if } x_1 \ is \ A \textbf{ and } x_2 \ is \ B \textbf{ then } y = f(x_1, x_2) \tag{2.7}$$

where both A and B are fuzzy sets in antecedent and $y = f(x_1, x_2)$ is a function in the consequence. Generally, $f(x_1, x_2)$ is a polynomial function and if it is not changed by the input values, the model is called "zero-order TSK model". When the function is a first-order polynomial, then it is called "first-order TSK model" and so on. By using the weighted average of all outputs, the final output is presented as follows:

$$y = \frac{\sum_{i=1}^{N} w_i y_i}{\sum_{i=1}^{N} w_i} \tag{2.8}$$

where w_i is firing strength and y_i is a first-order polynomial output.

2.2. Type-2 Fuzzy Logic Systems

The type-2 fuzzy sets have been introduced by Zadeh in 1975 and in the late 1990's it became more popular when Mendel made T2FLSs easy to use by providing some practical algorithms [8]. Type-2 fuzzy sets are proposed as an extension of the ordinary type-1 fuzzy sets which aim to model the uncertainties better in the rule base of the system. When a system has large amount of uncertainties, T1FLSs may not be able to achieve the desired performance level with a reasonable structural complexity [8]. In such cases, the use of T2FLSs is suggested as a preferable approach in the literature. In [32], T2FLSs are applied to real-time mobile robots for indoor and outdoor environments. Real-time implementations in [32] show that a traditional T1FLC cannot handle the system uncertainties effectively, and a T2FLC using type-2 fuzzy sets results in a better performance. Moreover, with the latter approach, the number of rules to be determined may be reduced (it should be noted that this may not mean a corresponding decrease in the parameters to be updated).

2.2.1. Type-2 Fuzzy Sets and Membership Functions

The membership grades of type-1 fuzzy sets are any crisp numbers in $[0, 1]$, a type-2 membership grade can take values in the closed interval of $[0, 1]$ which is called primary membership. On the other hand, there is a secondary membership value corresponding to each primary membership value defines the possibility of the primary memberships [32]. While secondary membership functions can take values in the closed interval of $[0, 1]$ in generalized T2FLSs, they are interval sets (either zero or one) in interval T2FLSs. Since the general T2FLSs are computationally intensive, most researchers prefer to use interval T2FLSs in their publications due to the fact that the computations are more manageable.

Similar to type-1 case, the primary and the secondary membership functions of type-2 fuzzy sets may have different shapes such as Gaussian, triangular, and trapezoidal. However, the most common one used in literature is Gaussian membership function.

Definition 2.7. A *type-2 Gaussian membership function* consists of an infinite number of type-1 ones. The general representation of type-2 Gaussian membership function shown in Figure 2.5 is given as follows:

$$\tilde{\mu}(x) = \exp\left[-\frac{(x-c)^2}{\sigma^2}\right] \tag{2.9}$$

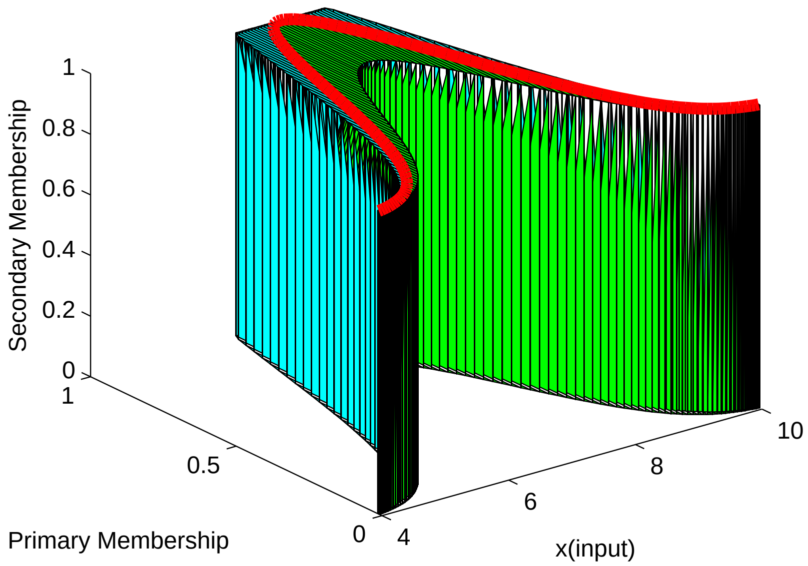


Figure 2.5. Generalized type-2 Gaussian membership function.

Definition 2.8. Uncertainty, conveyed by the union of all type-2's primary memberships, consists of a bounded region that is called *footprint of uncertainty (FOU)*.

By projecting a three-dimensional membership function in two-dimensional, the FOU of the function can be taken. When the third-dimension value is set to a fix value, that means the third-dimension is ignored, and FOU is used to describe it, then it is called interval type-2 fuzzy set. As it can be seen in Figure 2.6, the FOU of an interval type-2 fuzzy set is bounded with two type-1 membership functions. The upper bound of FOU is called “upper membership function”, while the lower bound of it is called “lower membership function”.

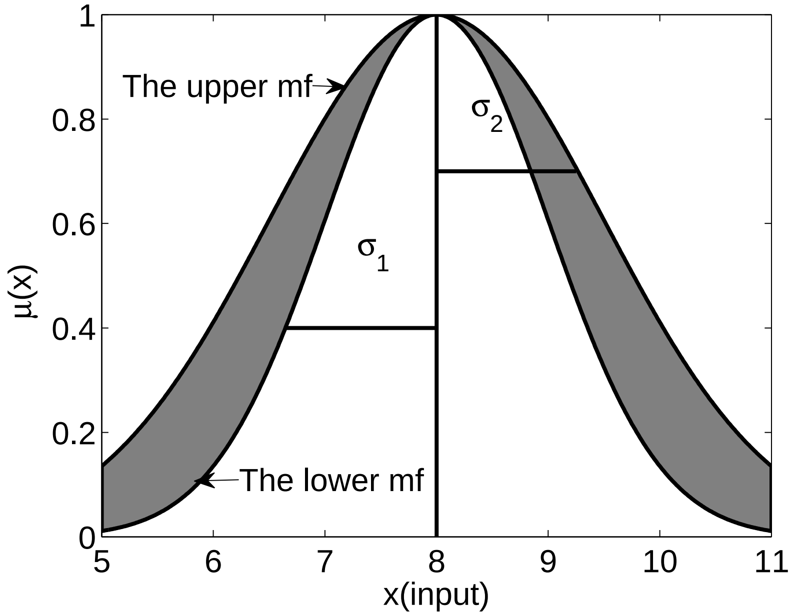


Figure 2.6. Gaussian membership function.

2.2.2. If-Then Rules

If-then rule for type-2 fuzzy logic is not too much different than type-1 one. The difference between them is about the nature of the membership functions which means T2FLS's antecedent or consequent sets are type-2 as well. Since this does not have an important effect on forming the rules, the structure of the rules stays totally the same for them. When a type-2 TSK have n inputs $x_1 \in X_1$, ..., $x_n \in X_n$ and one output $y \in Y$, the ith rule of first-order type-2 TSK model with M rules, each giving n antecedent can be given as follows:

$$R_i : \textbf{If } x_1 \textit{ is } \widetilde{A_{i1}} \textit{ and } ... \; x_n \textit{ is } \widetilde{A_{in}} \textbf{ then } Y_i = B_{i0} + B_{i1}x_1 + ... + B_{in}x_n \qquad (2.10)$$

where $(i = 1, ..., M)$, B_{ij} $(j = 1, ..., n)$ are consequent type-1 fuzzy sets, Y_i is the output of ith rule is also type-1 fuzzy set, and $\widetilde{A_{ik}}$ $(k = 1, ..., n))$ are type-2 antecedent fuzzy sets.

2.2.3. Type-2 Fuzzy Logic System Scheme

As type-1 FLS scheme, type-2 FLSs also include fuzzifier, rule base, fuzzy inference engine, and output processor as it is seen in Figure 2.7.

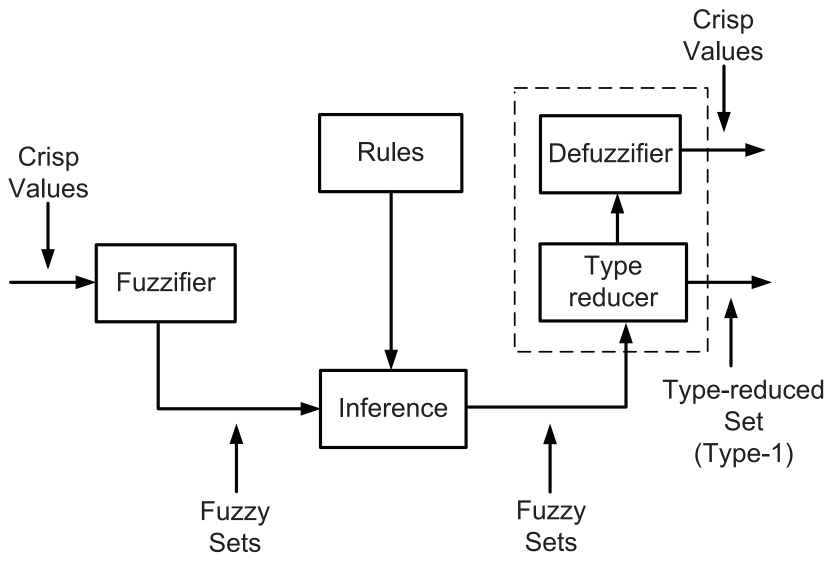


Figure 2.7. Type-2 fuzzy logic system.

In fuzzifier process, crisp input values are converted to a fuzzy set. The rules are generally *if-then* rules given in (2.10). Combining the rules and giving a mapping from input fuzzy sets to the output fuzzy sets are done in inference engine process. Unlike from type-1 defuzzifier which gives crisp output, type-2 defuzzifier gives a type-1 fuzzy set. Due to the fact that from type-2 output sets obtained from inference engine, type-1 fuzzy set is taken; hence, it is called “type-reduction”.

Similar to type-1 fuzzy systems, in type-2 ones there are two mostly used fuzzy inference systems which are Mamdani and TSK models. Both of these models characterized by *if-then* rules and have the same antecedent part. However, their consequent structures are not the same. Whereas Mamdani rule is a fuzzy set, TSK one’s is a function.

The form of the *ith* rule zero-order TSK type-2 fuzzy system for *nth* inputs is given as follows:

$$\textbf{If } x_1 \ is \ \widetilde{A_{i1}} \ and \ x_2 \ is \ \widetilde{A_{i2}} \ and \ ... \ and \ x_n \ is \ \widetilde{A_{in}} \ \textbf{then} \ y \ is \ b_i \tag{2.11}$$

where all x_n's are inputs, $\widetilde{A_{in}}$'s are antecedent sets, y is the output, and b_i's are crisp values [32].

3. FUZZY NEURAL NETWORKS

Since the FNNs combine the capability of fuzzy reasoning to handle uncertain information and the capability of ANNs to learn from input-output data sets and as such they have become a popular approach in engineering fields [33]. “While fuzzy logic provides an inference mechanism under cognitive uncertainty, computational neural networks offer exciting advantages, such as learning, adaptation, fault-tolerance, parallelism and generalization” [34].

3.1. The Control Scheme and FNN Structure

In this thesis FEL is used as the control structure for both type-1 fuzzy neural network (T1FNN) and type-2 fuzzy neural network (T2FNN).

3.1.1. The Feedback-Error Learning (FEL) Control Scheme

The control scheme proposed in this thesis is presented on Figure 3.1 where the FNN block with two inputs and one output can be a T1FNN or T2FNN. As it is mentioned in the introduction part of the thesis, this scheme is known as FEL control structure in literature [25]. It implements a TSK fuzzy model as presented on Figure 3.2 and Figure 3.3 for the T1FNN and T2FNN cases, respectively. In recent years, the TSK fuzzy model has gained more and more attention, especially in fuzzy control. This is due to the fact that by means of the TSK fuzzy model one is able to blend a number linearized models of the system [35].

The PI controller shown on Figure 3.1 acts as an ordinary feedback controller to ensure the stability of the system and as an inverse reference model of the response of the system under control. The PI control law is described as follows:

$$\tau_c = K_P e + K_I \int_x e dx \tag{3.1}$$

where $e = \omega_d - \omega$ is the feedback error, ω_d is the target value, K_P and K_I are the controller gains.

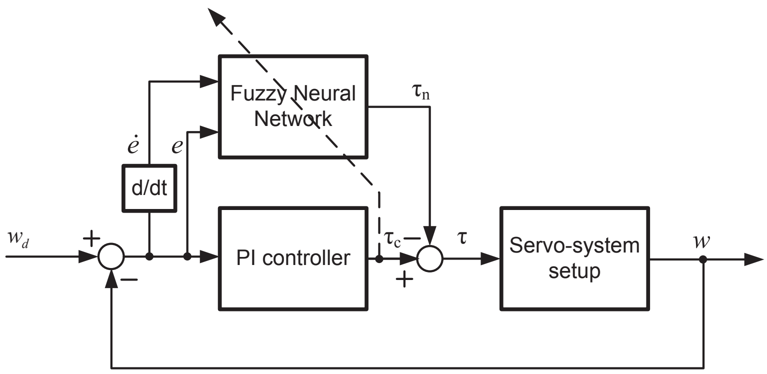


Figure 3.1. Block diagram of the proposed adaptive fuzzy neuro scheme.

3.1.2. Type-1 Fuzzy Neural Networks

The incoming signals, $x_1(t)$=$e(t)$ and $x_2(t)$=$\dot{e}(t)$, are fuzzified by using Gaussian membership functions, and are associated with 1_i and 2_j fuzzy subsets respectively which are defined by their corresponding membership functions $\mu_{1_i}(x_1)$ and $\mu_{2_j}(x_2)$ for $i = 1, ..., I$ and $j = 1, ..., J$ as it can be seen in Figure 3.2.

The fuzzy *if-then* rule R_{ij} of a first-order TSK model with two input variables where the consequent part is a linear function of the input variables can be defined as follows:

$$R_{ij} : \textbf{If } x_1 \textit{ is } 1_i \textit{ and } x_2 \textit{ is } 2_j, \textbf{ then } f_{ij} = a_i x_1 + b_j x_2 + d_{ij} \tag{3.2}$$

where a_i, b_j and d_{ij} are given constants. In the current investigation the coefficients a_i and b_j in the TSK fuzzy rule R_{ij} are assumed to be equal to zero.

The firing strength of the rule R_{ij} is obtained as a T-norm of the membership

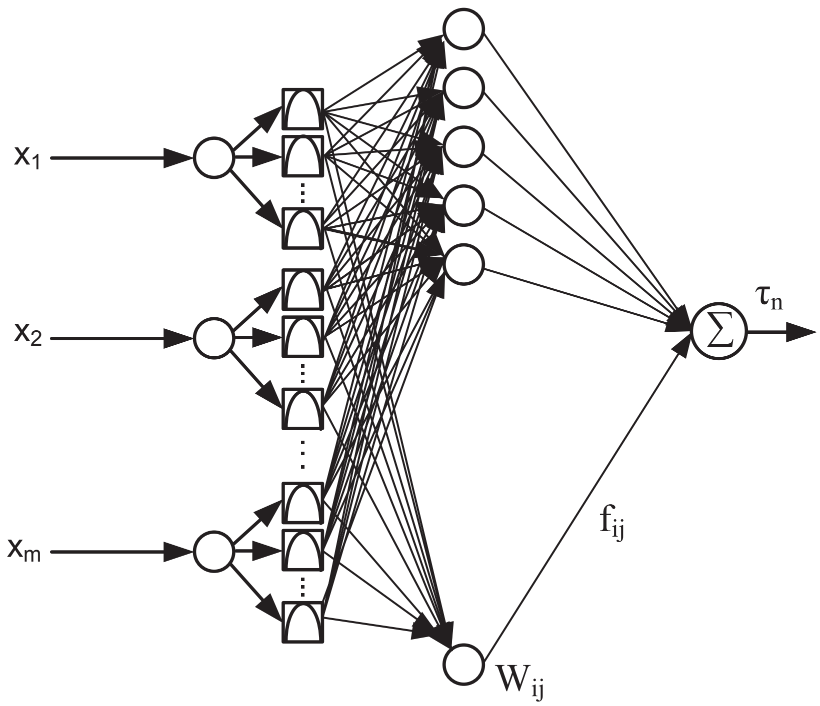


Figure 3.2. Structure of T1FNN for m inputs.

functions in the premise part (by using a multiplication operator):

$$W_{ij} = \mu_{1_i}(x_1)\mu_{2_j}(x_2) \tag{3.3}$$

The Gaussian membership functions $\mu_{1_i}(x_1)$ and $\mu_{2_j}(x_2)$ of the inputs x_1 and x_2 in the above expression have the following appearance:

$$\begin{aligned} \mu_{1_i}(x_1) &= \exp\left[-\frac{(x_1 - c_{1_i})^2}{\sigma_{1_i}^2}\right] \\ \mu_{2_j}(x_2) &= \exp\left[-\frac{(x_2 - c_{2_j})^2}{\sigma_{2_j}^2}\right] \end{aligned} \tag{3.4}$$

where the real constants $c, \sigma > 0$ are among the tunable parameters of the above fuzzy neural structure.

Equation (3.3) can be rewritten also as follows:

$$W_{ij} = \exp\left[-\frac{(x_1 - c_{1_i})^2}{\sigma_{1_i}^2} - \frac{(x_2 - c_{2_j})^2}{\sigma_{2_j}^2}\right] \tag{3.5}$$

The output signal of the FNN $\tau_n(t)$ is calculated as a weighted average of the output of each rule [35]:

$$\tau_n(t) = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}W_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}} \tag{3.6}$$

After the normalization of Equation (3.6), the output signal of the FNN will acquire the following form:

$$\tau_n(t) = \sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\overline{W}_{ij} \tag{3.7}$$

where $\overline{W}_{ij}$ is the normalized value of the output signal of the neuron ij from the second hidden layer of the network:

$$\bar{W}_{ij} = \frac{W_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}} \tag{3.8}$$

The input signal τ to the system to be controlled is as follows:

$$\tau = \tau_c - \tau_n \tag{3.9}$$

where τ_c and τ_n are the control signals generated by the PI controller and the fuzzy neuro feedback controller, respectively.

The following vectors have been specified:

- $X(t) = [x_1(t) \;\; x_2(t)]^T$ is the vector of the time varying input signals;

- $\overline{W}(t) = \left[\overline{W}_{11}(t) \;\; \overline{W}_{12}(t) \ldots \overline{W}_{21}(t) \;\ldots \overline{W}_{ij}(t) \;\ldots \overline{W}_{IJ}(t)\right]^T$ is the vector of the normalized output signals of the neurons from the second hidden layer;

- $\sigma_1 = [\sigma_{1_1} \ldots \sigma_{1_i} \ldots \sigma_{1_I}]^T$, $\sigma_2 = \left[\sigma_{2_1} \ldots \sigma_{2_j} \ldots \sigma_{2_J}\right]^T$, $c_1 = [c_{1_1} \ldots c_{1_i} \ldots c_{1_I}]^T$ and $c_2 = \left[c_{2_1} \ldots c_{2_j} \ldots c_{2_J}\right]^T$ are the vectors of the tuning parameters σ and c of the Gaussian membership functions relevant to the fuzzification of the signals supplied to the first and second input of the T1FNN, respectively;

- $f(t) = [f_{11}(t) \;\; f_{12}(t) \;\ldots f_{21}(t) \;\; f_{22}(t) \;\ldots f_{ij}(t) \;\ldots f_{IJ}(t)]$ is the vector of the time variable weight coefficients of the connections between the neurons from the second hidden layer and the output neuron of the fuzzy rule-based neural network.

Due to the control scheme adopted (Figure 3.1), where the conventional controller serves to guarantee stability in compact space, the input signals $x_1(t)$ and $x_2(t)$, and their time derivatives can be considered bounded. The following assumptions have been used in this thesis:

$$|x_1(t)| \leq B_x, \;\; |x_2(t)| \leq B_x \quad \forall t \tag{3.10}$$

$$|\dot{x}_1(t)| \leq B_{\dot{x}}, \;\; |\dot{x}_2(t)| \leq B_{\dot{x}} \quad \forall t \tag{3.11}$$

where B_x and $B_{\dot{x}}$ are assumed to be some known positive constants.

Based on the same arguments, the vectors defining the tuning parameters σ and c of the Gaussian membership functions are considered bounded as follows:

$$\|\sigma_1\| \leq B_\sigma, \;\; \|\sigma_2\| \leq B_\sigma, \;\; \|c_1\| \leq B_c, \;\; \|c_2\| \leq B_c \tag{3.12}$$

where B_σ and B_c are some known positive constants.

It will be assumed that, due to physical constraints, the time variable weight coefficients of the connections between the neurons in the second hidden layer and the output neuron are also bounded, *i.e.*,

$$|f_{ij}| \leq B_f \quad \forall t \tag{3.13}$$

for some positive constant B_f.

From Equations (3.3)-(3.8) and Equations (3.10)-(3.12) it follows that $0 < \overline{W}_{ij} < 1$. In addition, it can be easily seen from Equation (3.8) that $\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W}_{ij} = 1$.

From Equations (3.10)-(3.13) it follows that τ and $\dot{\tau}$ will be bounded signals too, *i.e.*

$$|\tau(t)| \leq B_\tau, \quad |\dot{\tau}(t)| \leq B_{\dot{\tau}} \quad \forall t \tag{3.14}$$

where B_τ and $B_{\dot{\tau}}$ are some known positive constants.

3.1.3. Type-2 Fuzzy Neural Networks

The fuzzy *if-then* rule R_{ij} of a zeroth-order type-2 TSK model with two input variables where the consequent part is a crisp number can be defined as follows:

$$R_{ij} : \textbf{If } x_1 \ is \ \widetilde{1}_i \ and \ x_2 \ is \ \widetilde{2}_j, \ \textbf{then } f_{ij} = d_{ij} \tag{3.15}$$

The T2FNN considered in this study uses type-2 membership functions in the premise part and crisp numbers in the consequent part as shown in Figure 3.3 (where a general first order TSK model is shown). This structure is called A2-C0 fuzzy system [13].

In the first layer of Figure 3.3, the input signals are fed into the system. In

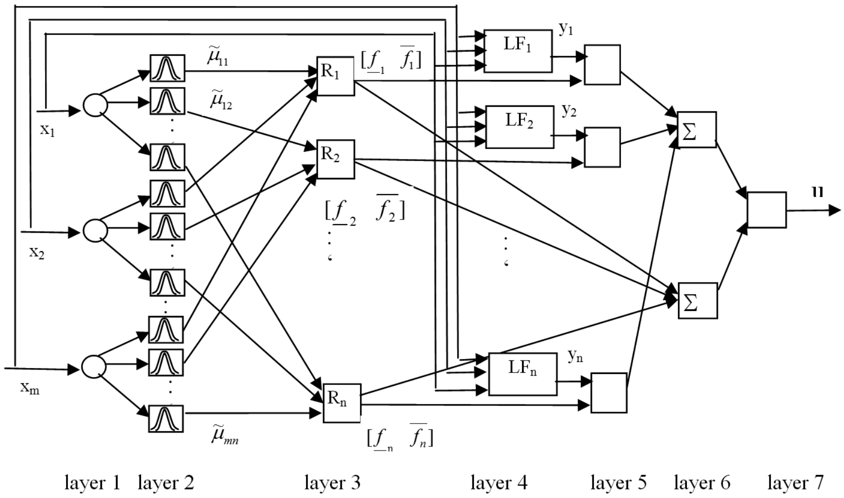


Figure 3.3. Structure of T2FNN for m inputs.

the second layer, each node corresponds to one linguistic term. In this layer, for each input signal entering the system, the upper and the lower membership degrees $\underline{\mu}$ and $\overline{\mu}$ are determined. The third layer calculates the firing strengths of the rules which are realized using the *prod* t-norm operator.

$$\underline{W_{ij}} = \underline{\mu}_{1_i}(x_1)\underline{\mu}_{2_j}(x_2) \tag{3.16}$$

$$\overline{W_{ij}} = \overline{\mu}_{1_i}(x_1)\overline{\mu}_{2_j}(x_2) \tag{3.17}$$

The fourth layer determines the outputs of the linear functions f_{ij} $(i = 1, \ldots, I$ and $j = 1, \ldots, J)$, in the consequent parts. In this thesis, as has been stated earlier, a zero order system is assumed and therefore

$$f_{ij} = d_{ij} \tag{3.18}$$

The fifth, the sixth and the seventh layers perform the type reduction and the defuzzification operations. After determining the firing strengths of rules, the defuzzified output of the type-2 TSK fuzzy system is determined. The final output of T2FNN is determined as:

$$Y_{TSK} = \int_{W_{11}\epsilon[\underline{W}_{11},\overline{W}_{11}]} \cdots \int_{W_{IJ}\epsilon[\underline{W}_{IJ},\overline{W}_{IJ}]} 1 \Bigg/ \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}(x) f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}(x)} \tag{3.19}$$

where f_{ij} is given by the *if-then* rule. In this study, the inference engine for type-2 TSK system proposed in [36] is used. The inference engine replaces the type-reduction which is given as:

$$\tau_n = \frac{q\sum_{i=1}^{M}\sum_{j=1}^{N} \underline{W}_{ij} f_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} \underline{W}_{ij}} + \frac{(1-q)\sum_{i=1}^{M}\sum_{j=1}^{N} \overline{W}_{ij} f_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} \overline{W}_{ij}} \tag{3.20}$$

The design parameter, q, weights the sharing of the lower and the upper firing levels of each fired rule [36]. $\overline{W}_{ij}$ and $\underline{W}_{ij}$are determined using Equations (3.16)-(3.17), and f_{ij} is determined using Equation (3.18).

In Figure 3.3, Layer 5 computes the product of the firing levels $\underline{W_{ij}}$ and $\overline{W_{ij}}$ and the linear functions f_{ij}. Layer 6 includes two summation blocks. One of these blocks computes the sum of the output signals from Layer 5 (the nominator part of Equation (3.20)) and the other block computes the sum of the output signal of Layer 4 (denominator part of Equation (3.20)). Layer 7 calculates output of the network using Equation (3.20).

After the calculation of the output signal in the T2FNN, the training of the network is started. The training includes an adjustment of the parameters c_{ij} and σ_{ij} in the membership functions in the second layer and the parameters of the linear functions in the fourth layer. In the next section, the parameter update rules of T2FNN are derived.

In Figures 3.4a-3.4b, Gaussian type-2 fuzzy membership functions $\underline{\mu_{1i}(x_1)}$, $\overline{\mu_{1i}(x_1)}$, $\underline{\mu_{2j}(x_2)}$, and $\overline{\mu_{2j}(x_2)}$ of the inputs x_1 and x_2 are shown. They have uncertain standard deviation as well as uncertain mean and their mathematical expressions are as follows:

$$\underline{\mu_{1i}(x_1)} = \exp\left[-\left(\frac{x_1 - \underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)^2 \right] \tag{3.21}$$

$$\overline{\mu_{1i}(x_1)} = \exp\left[-\left(\frac{x_1 - \overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)^2 \right] \tag{3.22}$$

$$\underline{\mu_{2j}(x_2)} = \exp\left[-\left(\frac{x_2 - \underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)^2 \right] \tag{3.23}$$

$$\overline{\mu_{2j}(x_2)} = \exp\left[-\left(\frac{x_2 - \overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)^2 \right] \tag{3.24}$$

where the real constants $\underline{c}, \overline{c}, \underline{\sigma}, \overline{\sigma} > 0$ are among the tunable parameters of the fuzzy neuro structure.

Hence, Equations (3.16)-(3.17) can be rewritten as follows:

$$\underline{W_{ij}} = \exp\left[-\left(\frac{x_1 - \underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)^2 - \left(\frac{x_2 - \underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)^2 \right] \tag{3.25}$$

$$\overline{W_{ij}} = \exp\left[-\left(\frac{x_1 - \overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)^2 - \left(\frac{x_2 - \overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)^2 \right] \tag{3.26}$$

After the normalization of Equation (3.20), the output signal of the FNN will acquire the following form:

$$\tau_n = q \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \widetilde{\underline{W_{ij}}} + (1-q) \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \widetilde{\overline{W_{ij}}} \tag{3.27}$$

where $\underline{\widetilde{W_{ij}}}$ and $\overline{\widetilde{W_{ij}}}$ are the normalized values of the lower and the upper output signals of the neuron ij from the second hidden layer of the network:

$$\underline{\widetilde{W_{ij}}} = \frac{\underline{W_{ij}}}{\sum_{i=1}^{I} \sum_{j=1}^{J} \underline{W_{ij}}} \tag{3.28}$$

$$\overline{\widetilde{W_{ij}}} = \frac{\overline{W_{ij}}}{\sum_{i=1}^{I} \sum_{j=1}^{J} \overline{W_{ij}}} \tag{3.29}$$

The input signal to the plant, τ, is as follows:

$$\tau = \tau_c - \tau_n \tag{3.30}$$

where τ_c and τ_n are the control signals generated by the PI controller and the fuzzy neuro feedback controller, respectively.

The following vectors have been specified:

- $X(t) = [x_1(t) \;\; x_2(t)]^T$ is the vector of the time varying input signals;
- $\underline{\widetilde{W}}(t) = \left[\underline{\widetilde{W_{11}}}(t) \; \underline{\widetilde{W_{12}}}(t) \ldots \underline{\widetilde{W_{21}}}(t) \; \ldots \underline{\widetilde{W_{ij}}}(t) \; \ldots \underline{\widetilde{W}_{IJ}}(t)\right]^T$ is the vector of the normalized lower output signals of the neurons from the second hidden layer;
- $\overline{\widetilde{W}}(t) = \left[\overline{\widetilde{W_{11}}}(t) \; \overline{\widetilde{W_{12}}}(t) \ldots \overline{\widetilde{W_{21}}}(t) \; \ldots \overline{\widetilde{W_{ij}}}(t) \; \ldots \overline{\widetilde{W}_{IJ}}(t)\right]^T$ is the vector of the normalized upper output signals of the neurons from the second hidden layer;
- $\underline{\sigma}_1 = \left[\underline{\sigma}_{1_1} \ldots \underline{\sigma}_{1_i} \ldots \underline{\sigma}_{1_I}\right]^T$, $\underline{\sigma}_2 = \left[\underline{\sigma}_{2_1} \ldots \underline{\sigma}_{2_j} \ldots \underline{\sigma}_{2_J}\right]^T$, $\underline{c}_1 = \left[\underline{c}_{1_1} \ldots \underline{c}_{1_i} \ldots \underline{c}_{1_I}\right]^T$ and $\underline{c}_2 = \left[\underline{c}_{2_1} \ldots \underline{c}_{2_j} \ldots \underline{c}_{2_J}\right]^T$ are the vectors of the tuning parameters $\underline{\sigma}$ and $\underline{c}$ of the Gaussian membership functions relevant to the first and the second lower membership functions of the T2FNN, respectively;

- $\overline{\sigma}_1 = [\overline{\sigma}_{1_1} \ldots \overline{\sigma}_{1_i} \ldots \overline{\sigma}_{1_I}]^T$, $\overline{\sigma}_2 = \left[\overline{\sigma}_{2_1} \ldots \overline{\sigma}_{2_j} \ldots \overline{\sigma}_{2_J}\right]^T$, $\overline{c}_1 = [\overline{c}_{1_1} \ldots \overline{c}_{1_i} \ldots \overline{c}_{1_I}]^T$ and $\overline{c}_2 = \left[\overline{c}_{2_1} \ldots \overline{c}_{2_j} \ldots \overline{c}_{2_J}\right]^T$ are the vectors of the tuning parameters $\overline{\sigma}$ and $\overline{c}$ of the Gaussian membership functions relevant to the the first and the second upper membership functions of the T2FNN, respectively.

Due to the control scheme adopted Figure 3.1, where the conventional controller serves to guarantee stability in compact space, the input signals $x_1(t)$ and $x_2(t)$, and their time derivatives can be considered bounded. The following assumptions have been used in this thesis:

$$|x_1(t)| \leq \widetilde{B_x}, \quad |x_2(t)| \leq \widetilde{B_x} \quad \forall t \tag{3.31}$$

$$|\dot{x}_1(t)| \leq \widetilde{B_{\dot{x}}}, \quad |\dot{x}_2(t)| \leq \widetilde{B_{\dot{x}}} \quad \forall t \tag{3.32}$$

where $\widetilde{B_x}$ and $\widetilde{B_{\dot{x}}}$ are assumed to be some known positive constants.

Based on the same arguments,the vectors defining the tuning parameters $\underline{\sigma}$, $\overline{\sigma}$, $\underline{c}$, and $\overline{c}$ of the Gaussian membership functions are considered bounded as follows:

$$\|\underline{\sigma}_1\| \leq B_{\underline{\sigma}}, \quad \|\underline{\sigma}_2\| \leq B_{\underline{\sigma}}, \quad \|\underline{c}_1\| \leq B_{\underline{c}}, \quad \|\underline{c}_2\| \leq B_{\underline{c}} \tag{3.33}$$

$$\|\overline{\sigma}_1\| \leq B_{\overline{\sigma}}, \quad \|\overline{\sigma}_2\| \leq B_{\overline{\sigma}}, \quad \|\overline{c}_1\| \leq B_{\overline{c}}, \quad \|\overline{c}_2\| \leq B_{\overline{c}} \tag{3.34}$$

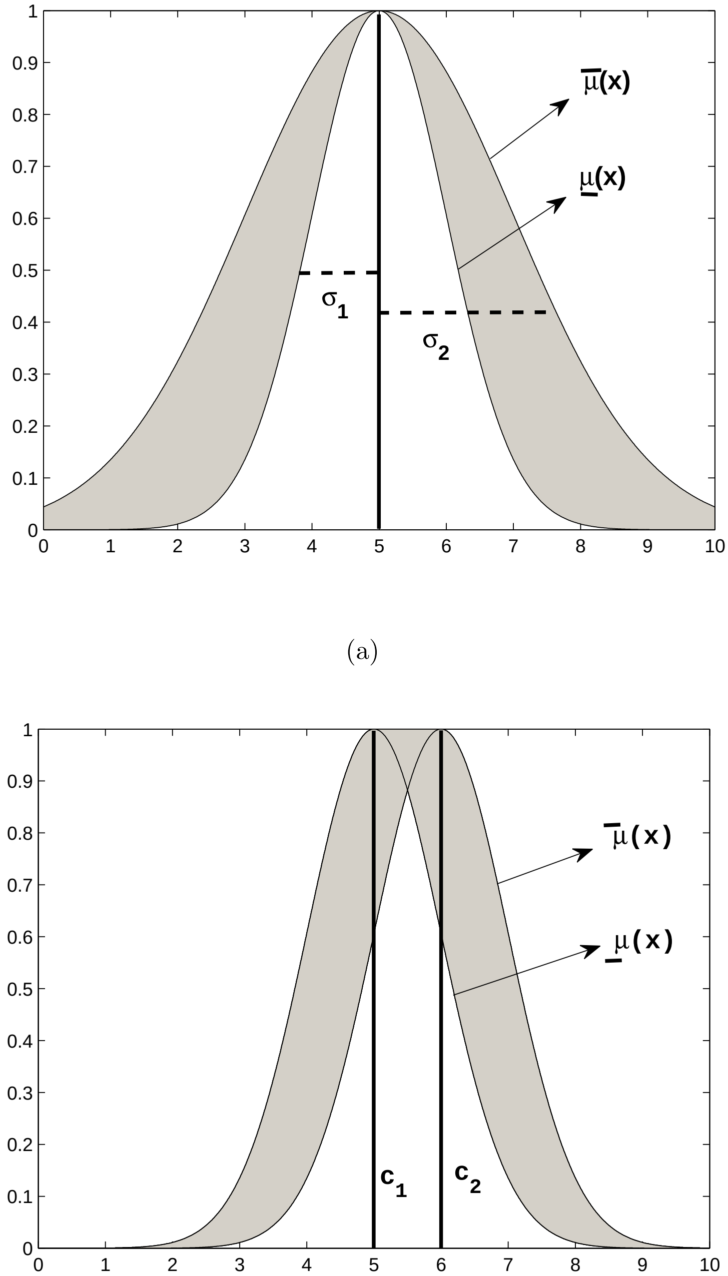
where $B_{\underline{\sigma}}$, $B_{\overline{\sigma}}$, $B_{\underline{c}}$, and $B_{\overline{c}}$ are some known positive constants.

It is to be noted that $0 < \underline{\widetilde{W_{ij}}} < 1$ and $0 < \widetilde{\overline{W_{ij}}} < 1$. In addition, it can be easily seen that $\sum_{i=1}^{I} \sum_{j=1}^{J} \underline{\widetilde{W_{ij}}} = 1$ and $\sum_{i=1}^{I} \sum_{j=1}^{J} \widetilde{\overline{W_{ij}}} = 1$. It is also considered that, τ

and $\dot{\tau}$ are bounded signals , *i.e.*

$$|\tau(t)| \leq B_{\tau}, \quad |\dot{\tau}(t)| \leq B_{\dot{\tau}} \quad \forall t \tag{3.35}$$

where B_{τ} and $B_{\dot{\tau}}$ are some known positive constants.



(a)

(b)

Figure 3.4. Type-2 fuzzy set with uncertain standard deviation (a) and uncertain mean (b).

4. TRAINING ALGORITHMS FOR FNNs

For tuning the parameters of FNNs, mostly gradient based and evolutionary algorithms are used in literature. The former works well when the system in hand has very slow variations in its dynamics. However, since the gradient-based algorithms (e.g. dynamic back propagation) include partial derivatives, the convergence speed may be slow especially when the search space is complex. What is more, with the repetitive algorithms, a number of numerical robustness issues may emerge when they are applied over long periods of time [37]. In addition to these drawbacks, the tuning process can easily be trapped into a local minimum [17]. To alleviate the problems mentioned, the use of the latter algorithm has been suggested [18]. However, the stability of such approaches is questionable and the optimal values for the stochastic operators are difficult to derive. Furthermore, the computational burden can be very high. To overcome these issues, VSSs theory-based algorithms are proposed for the parameter update rules of ANNs and T1FNNs as robust learning algorithms [19], [20].

4.1. The Basics of Sliding Mode Control

SMC is an alternative control method to handle the uncertainties in both linear and nonlinear systems in that it guarantees the robustness of a system in the case of changing working conditions and modeling ambiguities. Since SMC decreases the order of a system by one resulting in a possible simplification of the design procedure [38].

Using the sliding mode control theory principles [39] the zero value of the learning error coordinate $\tau_c(t)$ can be defined as time-varying sliding surface, *i.e.*,

$$S_c(\tau_n, \tau) = \tau_c(t) = \tau_n(t) + \tau(t) = 0 \tag{4.1}$$

which is the condition that the fuzzy neural network is trained to become a nonlinear regulator to obtain the desired response during the tracking-error convergence movement by compensation for the nonlinearity of the controlled plant.

The sliding surface for the nonlinear system under control $S_p(e, \dot{e})$ is defined as:

$$S_p(e, \dot{e}) = \dot{e} + \chi e \tag{4.2}$$

with χ being a constant determining the slope of the sliding surface.

Definition 4.1. A sliding motion will appear on the sliding manifold $S_c(\tau_n, \tau) = \tau_c(t) = 0$ after a time t_h, if the condition $S_c(t)\dot{S}_c(t) = \tau_c(t)\,\dot{\tau}_c(t) < 0$ is satisfied for all t in some nontrivial semi-open subinterval of time of the form $[t, t_h) \subset (-\infty, t_h)$.

It is desired to devise a dynamical feedback adaptation mechanism, or online learning algorithm for the fuzzy neural network parameters such that the sliding mode condition of the above definition is enforced.

4.1.1. Sliding-Mode Control-Based Learning Algorithm for T1FNN

Theorem 4.1. *If the adaptation law for the parameters of the considered T1FNN is chosen respectively as:*

$$\dot{c}_{1i} = \dot{x_1} \tag{4.3}$$

$$\dot{c}_{2j} = \dot{x_2} \tag{4.4}$$

$$\dot{\sigma_{1i}} = -\frac{(\sigma_{1i})^3}{(x_1 - c_{1i})^2}\alpha sgn(\tau_c) \tag{4.5}$$

$$\dot{\sigma_{2j}} = -\frac{(\sigma_{2j})^3}{(x_2 - c_{2j})^2}\alpha sgn(\tau_c) \tag{4.6}$$

$$\dot{f}_{ij} = -\frac{\overline{W}_{ij}}{\overline{W}^T\overline{W}}\alpha \operatorname{sign}(\tau_c) \tag{4.7}$$

where α is a sufficiently large positive design constant satisfying the following inequality:

$$\alpha > B_{\dot{\tau}} \tag{4.8}$$

then, given an arbitrary initial condition $\tau_c(0)$, the learning error $\tau_c(t)$ will converge firmly to zero during a finite time t_h.

Proof. The proof is given in Appendix A due to space constraints. □

In the adaptation laws of Equations (4.3)-(4.7), in order to avoid divisions by very small numbers, a lower limit of 0.001 is imposed on the denominators.

The relation between the sliding line S_p and the zero adaptive learning error level S_c, if χ is taken as $\chi = \frac{K_P}{K_I}$, is determined by the following equation:

$$S_c = \tau_c = K_I \int e + K_P e = K_I \left(\int e + \frac{K_P}{K_I} e \right) = K_I S_p \tag{4.9}$$

The tracking performance of the feedback control system in servo system can be analyzed by introducing the following Lyapunov function candidate:

$$V_p = \frac{1}{2} S_p^2 \tag{4.10}$$

Theorem 4.2. *If the adaptation strategy for the adjustable parameters of the T1FNN is chosen as in Equations (4.3)-(4.7), then the negative definiteness of the time derivative of the Lyapunov function in Equation (4.10) is ensured.*

Proof. The proof is given in Appendix B owing to space limits. □

It is noted that assuming the sliding mode control task is achievable using τ_c as a learning error for the T1FNN together with the adaptation laws Equations (4.3)-(4.7) enforces the desired reaching mode followed by a sliding regime for the system under control.

4.1.2. Sliding-Mode Control-Based Learning Algorithm for T2FNN

Theorem 4.3. *If the adaptation law for the parameters of the considered fuzzy neuro network is chosen respectively as:*

$$\dot{\underline{c}}_{1i} = \dot{\bar{c}}_{1i} = \dot{x}_1 \tag{4.11}$$

$$\dot{\underline{c}}_{2j} = \dot{\bar{c}}_{2j} = \dot{x}_2 \tag{4.12}$$

$$\dot{\underline{\sigma_{1i}}} = -\frac{(\underline{\sigma_{1i}})^3}{(x_1 - \underline{c_{1i}})^2} \alpha sgn(\tau_c) \tag{4.13}$$

$$\dot{\underline{\sigma_{2j}}} = -\frac{(\underline{\sigma_{2j}})^3}{(x_2 - \underline{c_{2j}})^2} \alpha sgn(\tau_c) \tag{4.14}$$

$$\dot{\bar{\sigma}}_{1i} = -\frac{(\bar{\sigma}_{1i})^3}{(x_1 - \bar{c}_{1i})^2} \alpha sgn(\tau_c) \tag{4.15}$$

$$\dot{\bar{\sigma}}_{2j} = -\frac{(\bar{\sigma}_{2j})^3}{(x_2 - \bar{c}_{2j})^2} \alpha sgn(\tau_c) \tag{4.16}$$

$$\dot{f}_{ij} = -\frac{(q\underline{\widetilde{W_{ij}}} + (1-q)\overline{\widetilde{W_{ij}}})}{(q\underline{\widetilde{W}} + (1-q)\overline{\widetilde{W}})^T (q\underline{\widetilde{W}} + (1-q)\overline{\widetilde{W}})} \alpha sgn(\tau_c) \tag{4.17}$$

where α is a sufficiently large positive design constant satisfying the inequality:

$$\alpha > B_{\dot{\tau}} \tag{4.18}$$

then, given an arbitrary initial condition $\tau_c(0)$, the learning error $\tau_c(t)$ will converge firmly to zero during a finite time t_h.

Proof. The proof is given in Appendix C due to space constraints. □

In the adaptation laws of Equations (4.11)-(4.17), in order to avoid divisions by very small numbers, a lower limit of 0.001 is imposed to the denominators.

It is well-known that sliding mode controllers suffer from high-frequency oscillations in the control input, which are called *chattering*. Chattering is an undesirable phenomena because it may excite the high-frequency dynamics of the system. There are two common methods used to eliminate it [40]: (*i*) The use of a saturation function to replace the signum function and (*ii*) the use of a boundary layer so that an equivalent control replaces the corrective one when the system is inside this layer.

Since when applying the second method, a finite steady-state error would always exist, most of the approaches use the saturation or the sigmoid function to replace the signum function. In order to reduce the chattering effect the function in Equation (4.19) has been used in this investigation instead of the sign function in the dynamic strategy described in Equations (4.11)-(4.17).

$$\mathrm{sign}(\tau_c) = \frac{\tau_c}{|\tau_c| + \delta} \tag{4.19}$$

where $\delta = 0.05$.

5. SIMULATION AND REAL-TIME RESULTS

In this thesis, SMC-theory based learning algorithm for T1FNN and T2FNN is tested on a permanently excited servo system specifically DC motor for both linear and nonlinear load conditions.

5.1. Mathematical Description of the Permanently Excited Servo System

The experimental setup [41] consists of two DC motors, which are connected by a mechanical clutch. The first motor is used for the control of the rotation speed or the shaft angle. The second one acts as a generator, by means of which nonlinear load conditions can be created (See Figure 5.1).

Figure 5.1. Servo system setup.

The nomenclature of the symbols being used is given in Table 1. The transfer function of the overall system can easily be derived as follows:

$$\omega(s) = \frac{1}{C\Phi}\frac{1}{1+T_M s+T_M T_A s^2}U_A(s) - \frac{R_A}{K_M C\Phi}\frac{1+T_A s}{1+T_M s+T_M T_A s^2}M_L(s) \qquad (5.1)$$

where

$$T_M = \frac{JR_A}{K_M C\Phi} \quad \text{and} \quad T_A = \frac{L_A}{R_A}$$

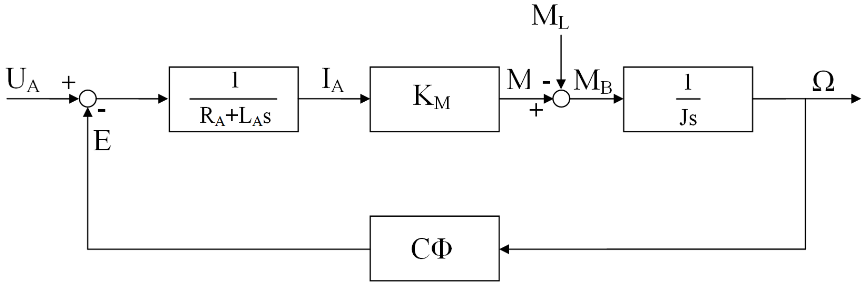


Figure 5.2. Block diagram of the motor with load.

Table 5.1. Nomenclature.

Name	**Description**
M_L	Load torque
K_M	Torque constant
I_A	Armature current
Ω	Speed of the rotor
$C\Phi$	Back emf constant
M_B	Acceleration torque
T_A	Electrical time constant
T_M	Mechanical time constant
U_A	Armature terminal voltage
E	Induced electromotive force
R_A	Armature winding resistance
L_A	Armature winding inductance
J	Moment of inertia of the system
M	The torque produced by the motor

The numerical values used in this study are: Armature terminal voltage= $24V$, rated torque= $0.096Nm$, moment of inertia of the system= 80.45x10$^{-6}kgm^2$, armature inductance= $3mH$, armature resistance= $3.13ohm$, back emf constant= $0.06Vs$, torque constant$= 0.06Nm/A$.

5.2. Case 1: Type-1 FNN Simulation and Real-time Results

5.2.1. Simulation Studies

The dynamic model of AMIRA DR300 DC motor experimental setup is used to test the performance of the FNS with the proposed learning algorithm. The sampling time is set to 1 ms for all the simulations.

In order to determine the efficiency and the accuracy of the proposed controller, two different types of load conditions are considered. Figures 5.3-5.5 show the speed responses of the motor for the proposed controller. The corresponding load condition starts with a value of 0 Nm, and increases suddenly to 0.032 Nm at the 2.5^{th}s, then again increases to 0.048 Nm at the 5^{th}s, finally reaches to 0.096 Nm (rated torque value for this motor) at the 7.5^{th}s. As can be seen from Figures. 5.3-5.5, FNS adapts its parameters when the load of the motor changes suddenly on different time periods. At the beginning, the dominating control signal has been the one coming from the PI controller. After a short time period however, using the control signal τ_c as a learning error, the neuro-fuzzy feedback controller has been able to take over the control, thus becoming the leading controller (see Figure 5.5).

Figures 5.6a-5.6b show the initial and the final places of the membership functions for the error and the time derivative of the error. As can be seen from this figure, both the center and the variance values of the Gaussian membership functions are being updated by the proposed learning algorithm.

Figures 5.7-5.9 show the speed response of motor under load condition which is proportional to the square of the speed, i.e. $M_L(t) = 0.00017(\omega)^2$. This type of load corresponds to the load-torque characteristics of centrifugal fans, pumps and blowers. This type of load requires much lower torque at low speeds than at high speeds. In both cases, the FNS with the proposed learning algorithms gives satisfactory performance.

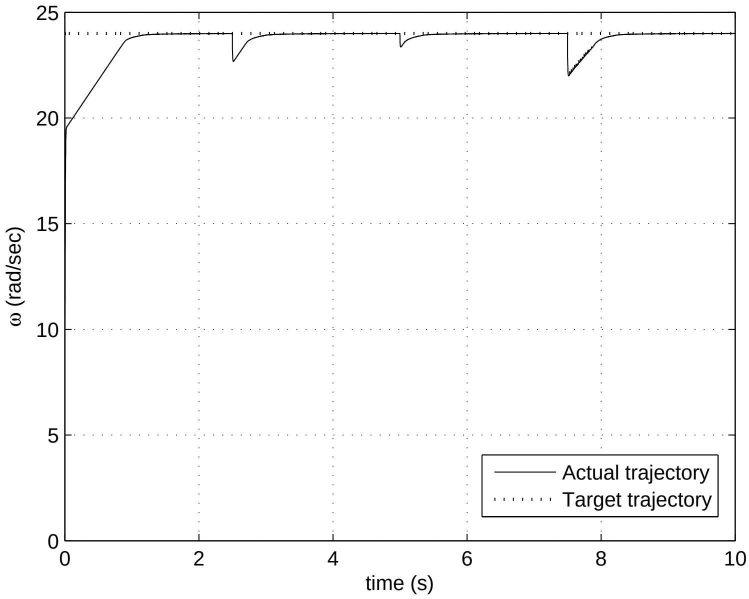


Figure 5.3. The speed responses of the motor for FNS.

5.2.2. Real-time Studies

AMIRA DR300 DC motor experimental setup is used to test the performance of the FNS with the proposed learning algorithm. The sampling time is set to 10 ms for all the experiments. Besides, the speed of the motor and the load torque are scaled to [-1,1] in real-time experiments.

Figures 5.10-5.12 show the output speed of the real-time setup, the error of the overall system and the control signals coming from the conventional controller (PI) and the FNS with the proposed learning algorithm, respectively. The corresponding load conditions start with a value of 0.024Nm, and increases suddenly to 0.048Nm at the 3.5^{th}s, then decreases to 0.024Nm at the 7.5^{th}s.

Similar to the simulation results, the performance of PI control law has been improved by the hybrid control methodology consisting of a conventional PI controller and FNS. As can be seen from Figure 5.12, the FNS is trying to take over the control

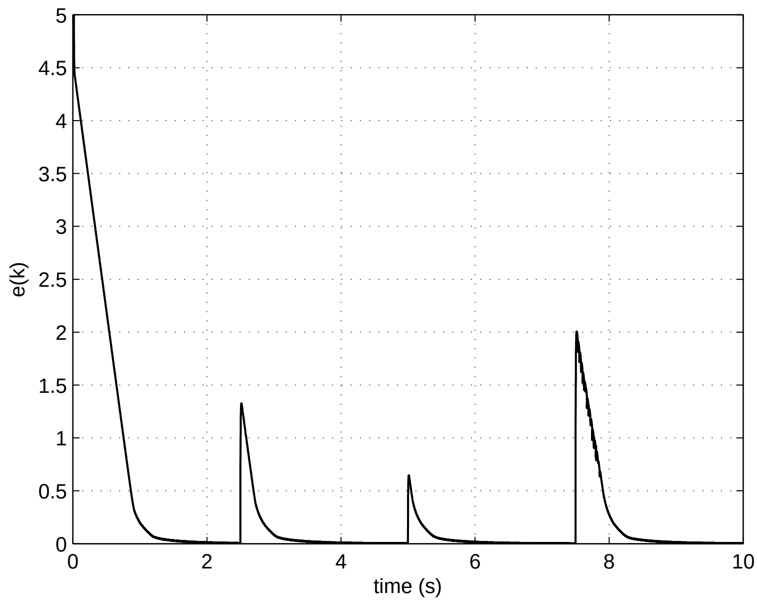


Figure 5.4. The error between the reference signal and the output of the motor for FNS.

operation after a short time period. The output of the PI controller tends to go to zero simultaneously.

5.3. Case 2: Type-2 FNN Real-time Results

AMIRA DR300 DC motor experimental setup is used to test the performance of the T2FNS with the proposed learning algorithm. The sampling time is set to 10 ms for all the experiments. In order to determine the efficiency and the accuracy of the proposed controller, two different types of load are used. Figures 5.13-5.15 show the speed responses of the motor for the proposed controller. The corresponding load condition starts with a value of 0.048Nm, and decreases suddenly to 0.024Nm on 3.5^{th}s, finally reaches to 0.048Nm on 7^{th}s. The load given to the system is noisy, and the noise power is 0.00005.

As can be seen from Figures 5.13-5.15, T2FNS adapts its parameters when the load of the motor changes suddenly on different time periods. At the beginning, the dominating control signal has been the one coming from the PI controller. After a

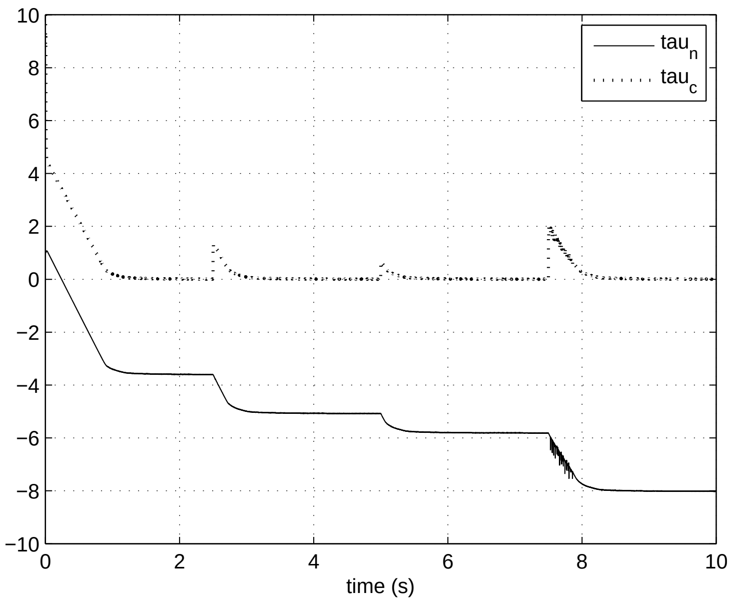


Figure 5.5. The control signals of the PI and FNS.

short time period however, using the control signal τ_c as a learning error, T2FNN-based controller has been able to take over the control, thus becoming the leading controller (see Figure 5.15)

Figures 5.16- 5.18 show the speed response of motor under load condition which is proportional to the square of the speed, i.e. $M_L(t) = 0.192 * (\omega)^2 Nm$. This type of load corresponds to the load-torque characteristics of centrifugal fans, pumps and blowers. This type of load requires much lower torque at low speeds than at high speeds. In both cases, the FNS with the proposed learning algorithms gives satisfactory performance.

Table 5.2. Square of the error values at each time step.

	Load type 1	**Load type 2**
T1FNS	1.8170	1.5475
T2FNS	1.7505	1.5436

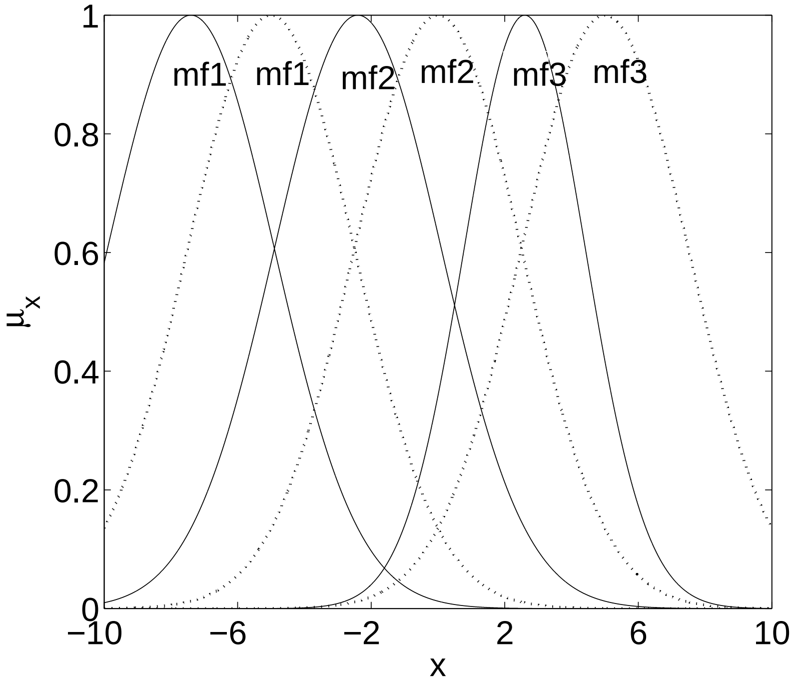
To be able to make a quantative comparison between T1FNNs and T2FNNs, Table 5.2 is given. As can be seen from it, T2FNN gives more accurate results than its type-1 counterpart. Moreover, the the initial and the final places of the type-2 membership functions can be seen in Table 5.3 and Table 5.4.

Table 5.3. Initial values of the centers and standard deviations of the input membership functions for load type 1.

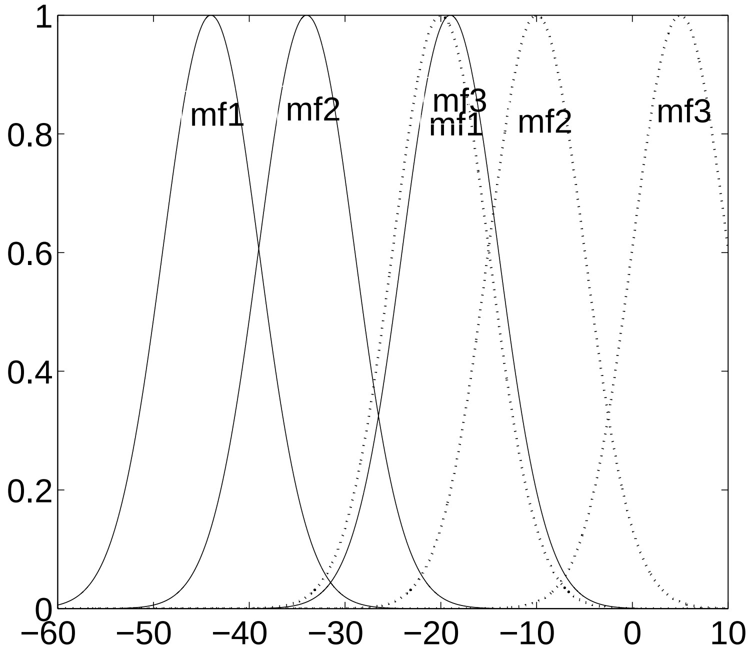
	Load type 1		
c_{1i}	-0.5000	0.0000	0.5000
c_{2i}	-5.0000	0.0000	5.0000
$\overline{\sigma_{1i}}$	0.5000	0.5000	0.5000
$\underline{\sigma_{1i}}$	0.2500	0.2500	0.2500
$\overline{\sigma_{2i}}$	5.0000	5.0000	5.0000
$\underline{\sigma_{2i}}$	2.5000	2.5000	2.5000

Table 5.4. Final values of the centers and standard deviations of the input membership functions for load type 1.

	Load type 1		
c_{1i}	-0.4498	0.0051	0.5550
c_{2i}	-4.0537	0.9462	5.9463
$\overline{\sigma_{1i}}$	0.3052	0.0296	0.1856
$\underline{\sigma_{1i}}$	0.2530	0.0113	0.1759
$\overline{\sigma_{2i}}$	1.6410	0.1591	0.9979
$\underline{\sigma_{2i}}$	1.3108	0.1032	0.6066



(a)



(b)

Figure 5.6. The initial (dotted lines) and final (line) places of the membership functions for the input 1 (a) and input 2 (b).

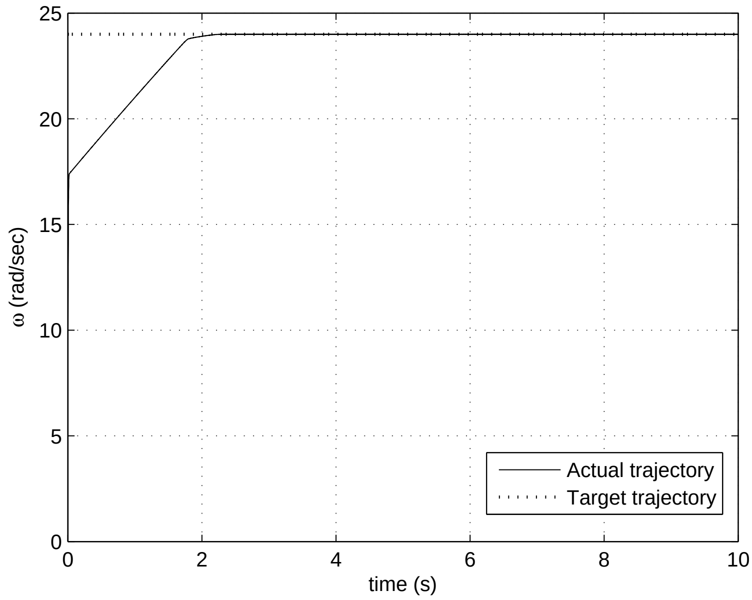


Figure 5.7. The speed responses of the motor for FNS.

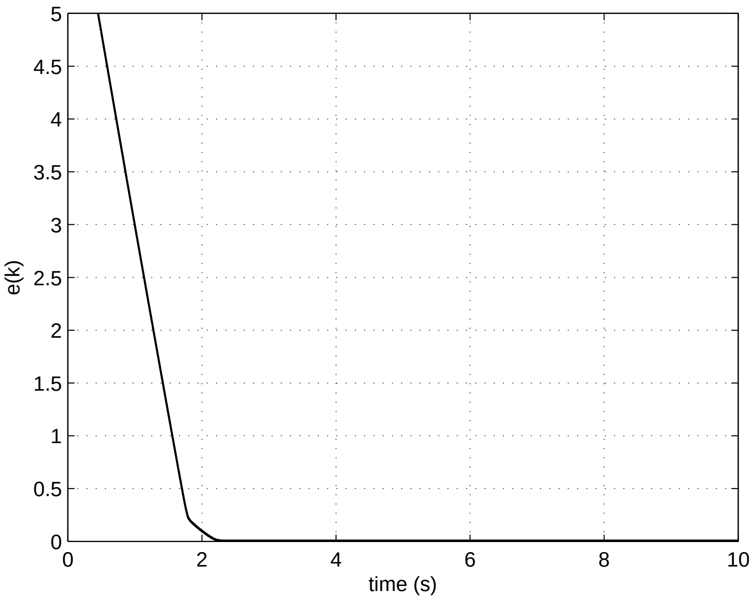


Figure 5.8. The error between the reference signal and the output of the motor for FNS.

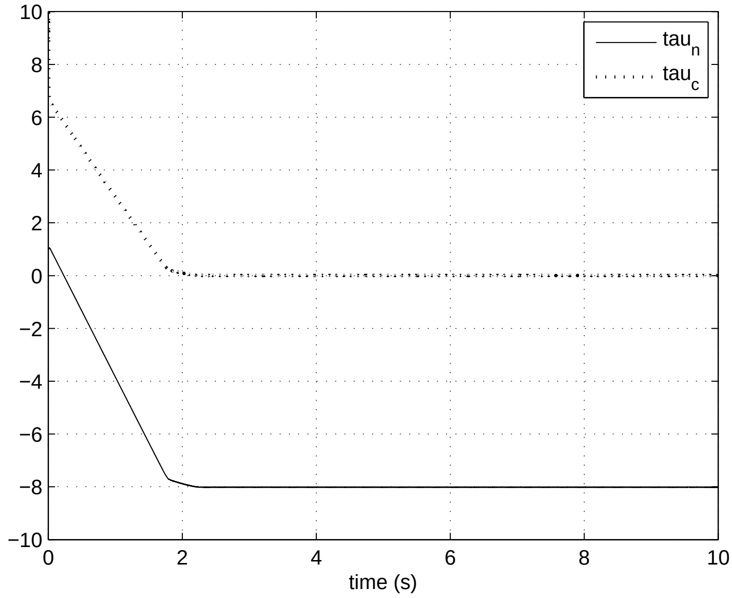


Figure 5.9. The control signals of the PI and FNS.

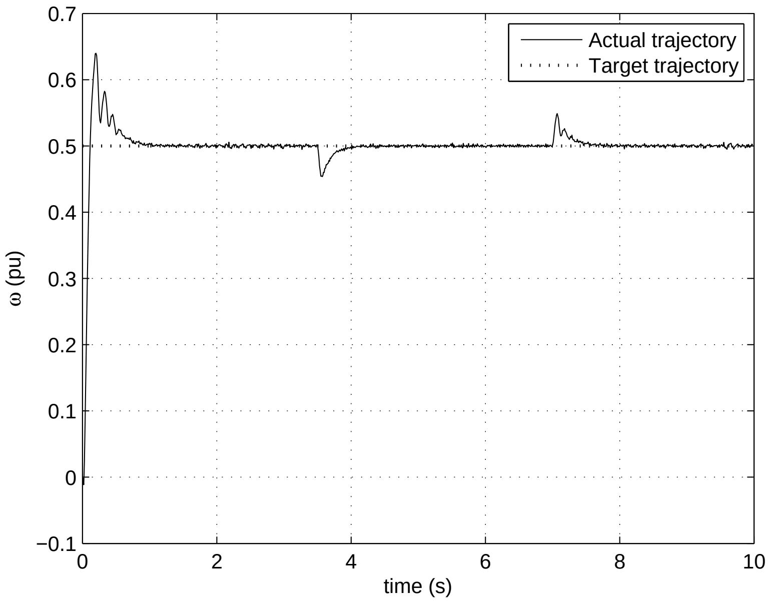


Figure 5.10. The speed responses of the motor for FNS.

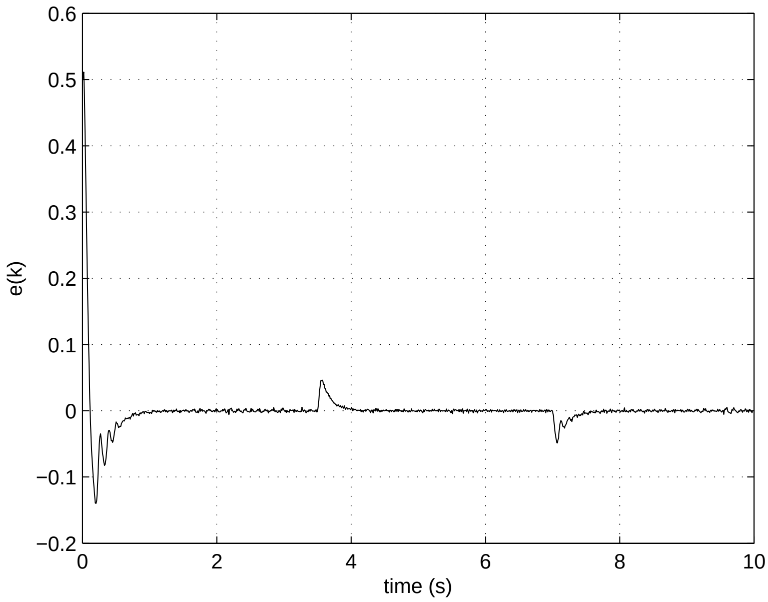


Figure 5.11. The error between the reference signal and the output of the motor for FNS.

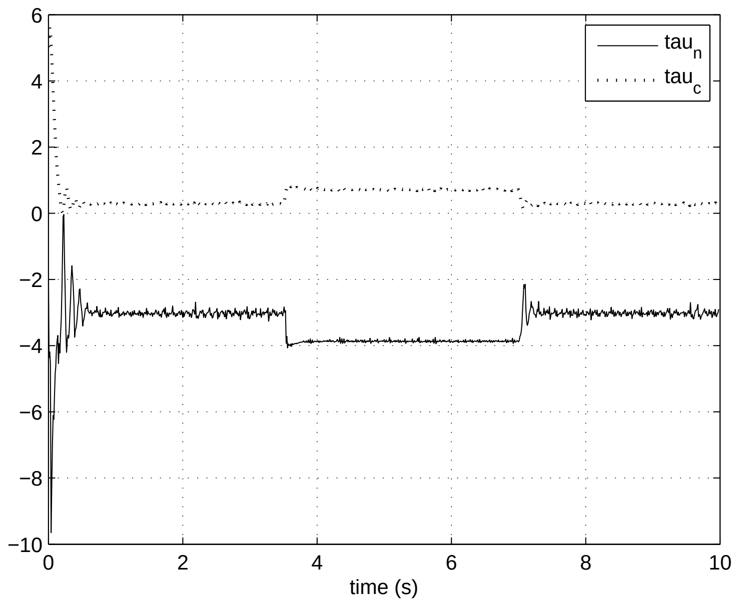


Figure 5.12. The control signals of the PI and FNS.

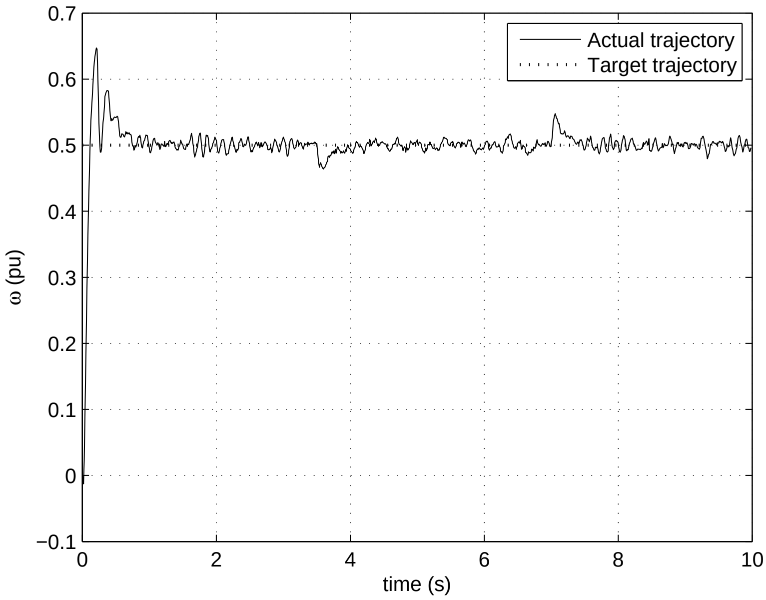


Figure 5.13. The speed responses of the motor for FNS.

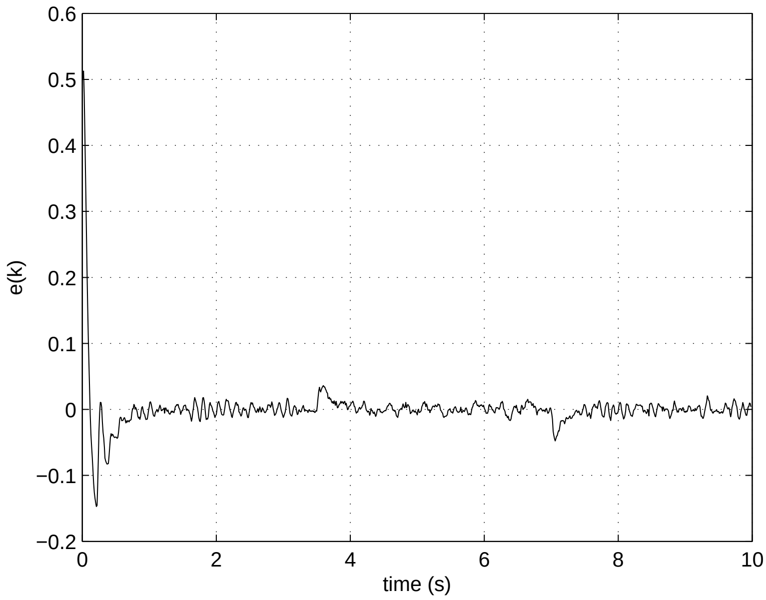


Figure 5.14. The error between the reference signal and the output of the motor for FNS.

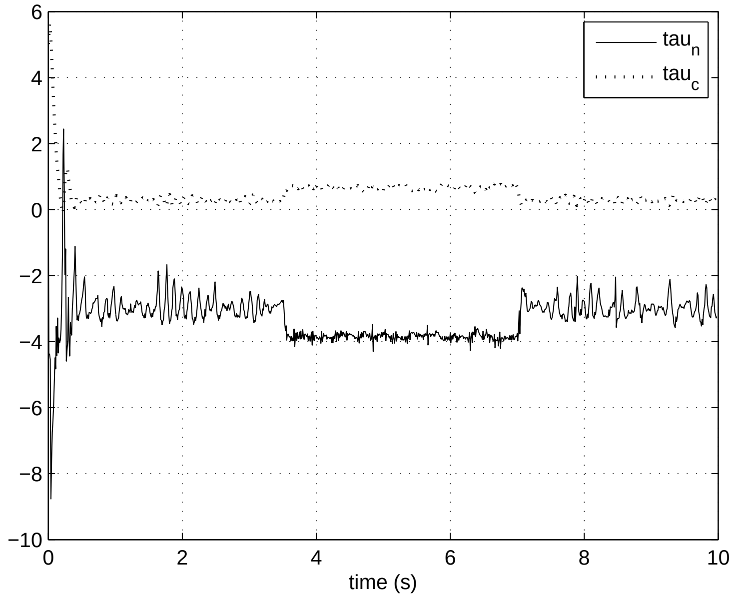


Figure 5.15. The control signals of the PI and FNS.

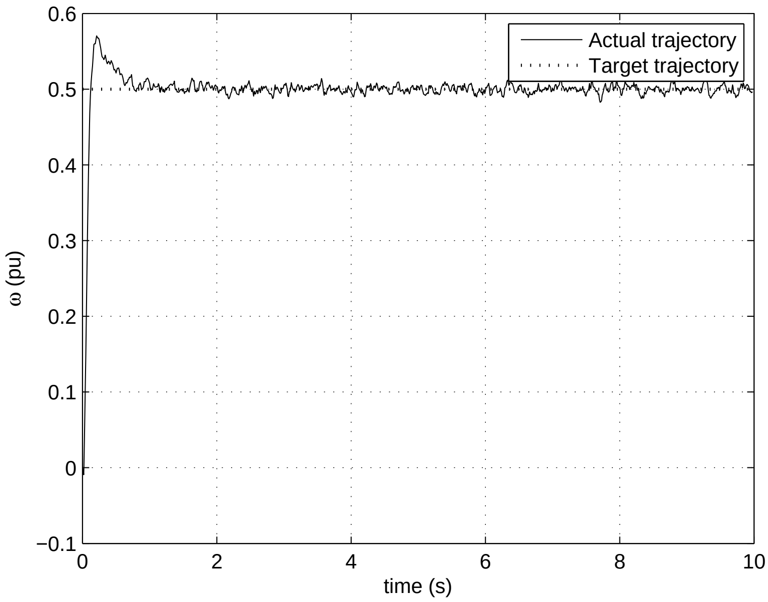


Figure 5.16. The speed responses of the motor for FNS.

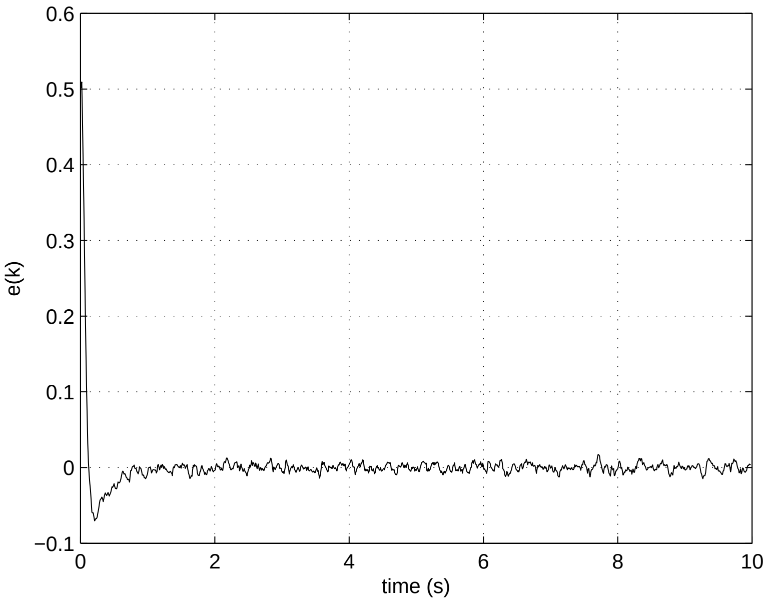


Figure 5.17. The error between the reference signal and the output of the motor for FNS.

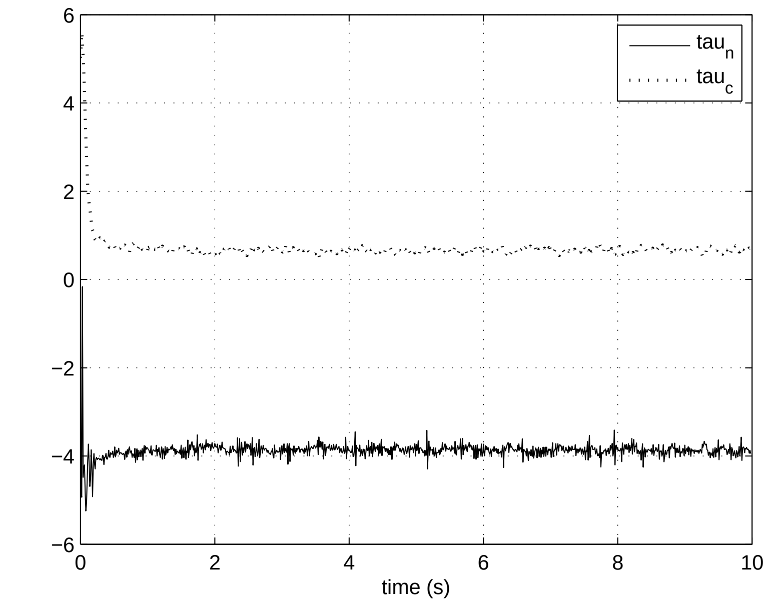


Figure 5.18. The control signals of the PI and FNS.

6. CONCLUSION

In industrial applications, uncertainty is an inevitable problem and FLSs provide a way to handle ambigious knowledge. Since FLSs are able to overcome imprecision information and ANNs are capable of learning from input-output data sets , FNNs combine them to have a better behavior. To train FNNs, there are commonly used training methods such as gradient based and evolutionary algorithms. Owing to the fact that the former includes partial derivatives, the convergence speed might be slower than the desired time and the latter's computational burden is too much, it is not useful in all cases. As an alternative approach, in this thesis, an SMC-based learning algorithm is proposed to train T2FNNs for the first time in literature besides training T1FNNs. The methods proposed in this study has the ability to learn the plant model online instead of using an accurate predefined dynamical equations of the system. Using a combination of FLSs, ANNs, and VSSs theory harmoniously allows us to better handle the uncertainties and lack of modeling information.

The main objective of this thesis are to train T2FNNs by using SMC-based learning algorithm. The parameter update rules are derived for a structure with two inputs, each being modeled by type-2 membership functions with uncertain standard deviations but fixed means. In order to show the effectiveness of the algorithm, the method is tested through simulations and on a real-time laboratory set-up, namely, servo system. It is seen through the studies carried out that the T2FNN with sliding mode learning algorithm can control the real-time servo system effectively, with better noise rejection capabilities as compared to a T1FNN structure. The prominent feature of the learning algorithm proposed (in addition to its robustness) is its computational simplicity as compared with gradient based and genetic algorithms. Hence, it makes the proposed algorithm relatively easy to implement.

APPENDIX A: Proof of Theorem 1

$$\mu_{1i}(x_1) = \exp\left[-(\frac{x_1 - c_{1i}}{\sigma_{1i}})^2 \right] \tag{A.1}$$

$$\mu_{2j}(x_2) = \exp\left[-(\frac{x_2 - c_{2j}}{\sigma_{2j}})^2 \right] \tag{A.2}$$

The time derivatives of Equations (A.1)-(A.2) are as follows:

$$\dot{\mu_{1i}(x_1)} = -2(\frac{x_1 - c_{1i}}{\sigma_{1i}})(\frac{x_1 - c_{1i}}{\sigma_{1i}})' \exp\left[-(\frac{x_1 - c_{1i}}{\sigma_{1i}})^2 \right] \tag{A.3}$$

$$\dot{\mu_{2j}(x_2)} = -2(\frac{x_2 - c_{2j}}{\sigma_{2j}})(\frac{x_2 - c_{2j}}{\sigma_{2j}})' \exp\left[-(\frac{x_2 - c_{2j}}{\sigma_{2j}})^2 \right] \tag{A.4}$$

By defining Equations (A.5)-(A.6), the Equations (A.3)-(A.4) are written in a simpler way as follows:

$$A_{1i} = (\frac{x_1 - c_{1i}}{\sigma_{1i}}) \tag{A.5}$$

$$A_{2j} = (\frac{x_2 - c_{2j}}{\sigma_{2j}}) \tag{A.6}$$

Then Equations (A.3)-(A.4) can be written as follows:

$$\dot{\mu_{1i}(x_1)} = -2A_{1i}(A_{1i})'\mu_{1i}(x_1) \tag{A.7}$$

$$\dot{\mu_{2j}}(x_2) = -2A_{2j}(A_{2j})'\mu_{2j}(x_2) \tag{A.8}$$

$$W_{ij} = \mu_{1i}(x_1)\mu_{2j}(x_2) \tag{A.9}$$

The normalized value of the output signal of the neuron ij from the second hidden layer of the network can be written as below:

$$\overline{W_{ij}} = \frac{W_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}} \tag{A.10}$$

By using Equation (A.9), the time derivative of Equation (A.10) is taken as follows:

$$\dot{\overline{W_{ij}}} = \frac{\Big(\mu_{1i}(x_1)\mu_{2j}(x_2)\Big)'\Big(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})^2} - \frac{\Big(W_{ij}\Big)\Big(\sum_{i=1}^{I}\sum_{j=1}^{J} \mu_{1i}(x_1)\mu_{2j}(x_2)\Big)'}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})^2} \tag{A.11}$$

$$\begin{aligned}\dot{\overline{W_{ij}}} = {} & \frac{\Big(\dot{\mu_{1i}}(x_1)\mu_{2j}(x_2) + \mu_{1i}(x_1)\dot{\mu_{2j}}(x_2)\Big)\Big(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})^2} \\ & - \frac{(W_{ij})\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\dot{\mu_{1i}}(x_1)\mu_{2j}(x_2) + \mu_{1i}(x_1)\dot{\mu_{2j}}(x_2)\Big)\Big]}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})^2}\end{aligned} \tag{A.12}$$

Equation (A.12) can be simplified as follows, by using Equation (A.10):

$$\begin{aligned}\dot{\overline{W_{ij}}} = {} & \frac{\Big(\dot{\mu_{1i}}(x_1)\mu_{2j}(x_2) + \mu_{1i}(x_1)\dot{\mu_{2j}}(x_2)\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})} \\ & - \frac{(\overline{W_{ij}})\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\dot{\mu_{1i}}(x_1)\mu_{2j}(x_2) + \mu_{1i}(x_1)\dot{\mu_{2j}}(x_2)\Big)\Big]}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})}\end{aligned} \tag{A.13}$$

$$\dot{\overline{W_{ij}}} = \frac{\Big(-2A_{1i}(A_{1i})'\mu_{1i}(x_1)\mu_{2j}(x_2) - 2A_{2j}(A_{2j})'\mu_{1i}(x_1)\mu_{2j}(x_2)\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})}$$
$$- \frac{(\overline{W_{ij}})\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(-2A_{1i}(A_{1i})'\mu_{1i}(x_1)\mu_{2j}(x_2)\Big)\Big]}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})}$$
$$- \frac{(\overline{W_{ij}})\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(-2A_{2j}(A_{2j})'\mu_{1i}(x_1)\mu_{2j}(x_2)\Big)\Big]}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})} \quad \text{(A.14)}$$

$$\dot{\overline{W_{ij}}} = \frac{\Big(\mu_{1i}(x_1)\mu_{2j}(x_2)\Big)\Big(-2A_{1i}(A_{1i})' - 2A_{2j}(A_{2j})'\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})}$$
$$- \frac{(\overline{W_{ij}})\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\mu_{1i}(x_1)\mu_{2j}(x_2)\Big)\Big(-2A_{1i}(A_{1i})' - 2A_{2j}(A_{2j})'\Big)\Big]}{(\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij})} \quad \text{(A.15)}$$

$$\dot{\overline{W}}_{ij} = -\overline{W}_{ij}\dot{K}_{ij} + \overline{W}_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\overline{W}_{ij}\dot{K}_{ij}\Big) \quad \text{(A.16)}$$

where

$$\dot{K}_{ij} = 2\Big(A_{1i}(A_{1i})' + A_{2j}(A_{2j})'\Big)$$

By using the following Lyapunov function, the stability condition can be checked:

$$V_c = \frac{1}{2}\tau_c^2(t) \quad \text{(A.17)}$$

The time derivative of V_c is given by:

$$\dot{V}_c = \tau_c\dot{\tau}_c = \tau_c(\dot{\tau}_n + \dot{\tau}) \quad \text{(A.18)}$$

where $\tau_n = \sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\overline{W_{ij}}$ and $\dot{\tau_n} = \sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f_{ij}}\overline{W_{ij}} + f_{ij}\dot{\overline{W_{ij}}})$.

$$\dot{V_c} = \tau_c\Big\{\sum_{i=1}^{I}\sum_{j=1}^{J}\Big[\dot{f_{ij}}\overline{W}_{ij} + f_{ij}\Big(-\overline{W}_{ij}\dot{K_{ij}} + \overline{W}_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W}_{ij}\dot{K_{ij}}\Big)\Big] + \dot{\tau}\Big\} =$$
$$= \tau_c\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f_{ij}}\overline{W}_{ij} - 2\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W}_{ij}\Big(A_{1i}(A_{1i})' + A_{2j}(A_{2j})'\Big)f_{ij} +$$
$$+ 2\sum_{i=1}^{I}\sum_{j=1}^{J}[\overline{W}_{ij}f_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W}_{ij}\Big(A_{1i}(A_{1i})' + A_{2j}(A_{2j})'\Big)] + \dot{\tau}\Big] =$$

where

$$\dot{A_{1i}} = \frac{(\dot{x_1} - \dot{c_{1i}})(\sigma_{1i}) - (x_1 - c_{1i})\dot{\sigma_{1i}}}{{\sigma_{1i}}^2}$$

$$\dot{A_{2j}} = \frac{(\dot{x_2} - \dot{c_{2j}})(\sigma_{2j}) - (x_2 - c_{2j})\dot{\sigma_{2j}}}{{\sigma_{2j}}^2}$$

Equation (A.19) can be obtained by using Equations (4.3)-(4.6);

$$A_{1i}\dot{A_{1i}} = A_{2j}\dot{A_{2j}} = \alpha sgn(\tau_c) \tag{A.19}$$

$$\dot{V_c} = \tau_c\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f_{ij}}\overline{W}_{ij} - 4\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W}_{ij}(\alpha sgn(\tau_c))f_{ij} +$$
$$+ 4\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\overline{W}_{ij}f_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W}_{ij}(\alpha sgn(\tau_c))\Big) + \dot{\tau}\Big]$$

Since $\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}} = 1$,

$$\dot{V_c} = \tau_c\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f_{ij}}\overline{W}_{ij} + \dot{\tau}\Big]$$

where

$$\dot{f}_{ij} = -\frac{\overline{W}_{ij}}{\overline{W}^T\overline{W}}\alpha \mathrm{sign}\left(\tau_c\right) \tag{A.20}$$

$$\dot{V}_c = \tau_c \bigg[-\alpha sgn(\tau_c) + \dot{\tau} \bigg] \tag{A.21}$$

$$\dot{V}_c = \bigg[-\alpha|\tau_c| + |\tau_c|B_{\dot{\tau}} \bigg] < 0 \tag{A.22}$$

$\square$

APPENDIX B: Proof of Theorem 2

Evaluating the time derivative of the Lyapunov function in Equation (4.10) yields:

$$
\begin{aligned}
\dot{V}_p = \dot{S}_p S_p &= \frac{1}{K_I^2}\dot{S}_c S_c \\
&\leq \frac{|\tau_c|}{K_I^2}\left[-\alpha + B_{\dot{\tau}}\right] \\
&< 0, \qquad \forall S_c, S_p \neq 0
\end{aligned}
\tag{B.1}
$$

□

APPENDIX C: Proof of Theorem 3

$$\underline{\mu_{1i}(x_1)} = \exp\left[-\left(\frac{x_1-\underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)^2\right] \tag{C.1}$$

$$\overline{\mu_{1i}(x_1)} = \exp\left[-\left(\frac{x_1-\overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)^2\right] \tag{C.2}$$

$$\underline{\mu_{2j}(x_2)} = \exp\left[-\left(\frac{x_2-\underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)^2\right] \tag{C.3}$$

$$\overline{\mu_{2j}(x_2)} = \exp\left[-\left(\frac{x_2-\overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)^2\right] \tag{C.4}$$

Time derivative of the all Gaussian membership functions is written as follows:

$$\underline{\dot{\mu_{1i}(x_1)}} = -2\left(\frac{x_1-\underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)\left(\frac{x_1-\underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)' \exp\left[-\left(\frac{x_1-\underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)^2\right] \tag{C.5}$$

$$\overline{\dot{\mu_{1i}(x_1)}} = -2\left(\frac{x_1-\overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)\left(\frac{x_1-\overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)' \exp\left[-\left(\frac{x_1-\overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)^2\right] \tag{C.6}$$

$$\underline{\dot{\mu_{2j}(x_2)}} = -2\left(\frac{x_2-\underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)\left(\frac{x_2-\underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)' \exp\left[-\left(\frac{x_2-\underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)^2\right] \tag{C.7}$$

$$\overline{\dot{\mu_{2j}(x_2)}} = -2\left(\frac{x_2-\overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)\left(\frac{x_2-\overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)' \exp\left[-\left(\frac{x_2-\overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)^2\right] \tag{C.8}$$

By defining (C.9)-(C.12), the equation (C.5)-(C.8) are written in a simpler way as follows:

$$\underline{A_{1i}} = (\frac{x_1 - \underline{c_{1i}}}{\underline{\sigma_{1i}}}) \tag{C.9}$$

$$\underline{A_{2j}} = (\frac{x_2 - \underline{c_{2j}}}{\underline{\sigma_{2j}}}) \tag{C.10}$$

$$\overline{U_{1i}} = (\frac{x_1 - \overline{c_{1i}}}{\overline{\sigma_{1i}}}) \tag{C.11}$$

$$\overline{U_{2j}} = (\frac{x_2 - \overline{c_{2j}}}{\overline{\sigma_{2j}}}) \tag{C.12}$$

Then Equations (C.5)-(C.12) can be written as follows:

$$\dot{\underline{\mu_{1i}(x_1)}} = -2A_{1i}(A_{1i})'\underline{\mu_{1i}(x_1)} \tag{C.13}$$

$$\dot{\overline{\mu_{1i}(x_1)}} = -2U_{1i}(U_{1i})'\overline{\mu_{1i}(x_1)} \tag{C.14}$$

$$\dot{\underline{\mu_{2j}(x_2)}} = -2A_{2j}(A_{2j})'\underline{\mu_{2j}(x_2)} \tag{C.15}$$

$$\dot{\overline{\mu_{2j}(x_2)}} = -2U_{2j}(U_{2j})'\overline{\mu_{2j}(x_2)} \tag{C.16}$$

$$\underline{W_{ij}} = \underline{\mu_{1i}(x_1)}\underline{\mu_{2j}(x_2)} \tag{C.17}$$

$$\overline{W_{ij}} = \overline{\mu_{1i}(x_1)\mu_{2j}(x_2)} \tag{C.18}$$

The normalized value of the lower and upper output signal of the neuron ij from the second hidden layer of the network can be written as below:

$$\underline{\widetilde{W_{ij}}} = \frac{\underline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}} \tag{C.19}$$

$$\widetilde{\overline{W_{ij}}} = \frac{\overline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}}} \tag{C.20}$$

The time derivative of Equations (C.19)-(C.20) are written as follows:

$$\underline{\dot{\widetilde{W_{ij}}}} = \frac{\left(\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\right)'\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}\right)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})^2} - \frac{\left(\underline{W_{ij}}\right)\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\right)'}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})^2} \tag{C.21}$$

$$\dot{\widetilde{\overline{W_{ij}}}} = \frac{\left(\overline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\right)'\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}}\right)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}})^2} - \frac{\left(\overline{W_{ij}}\right)\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\right)'}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}})^2} \tag{C.22}$$

$$\begin{aligned}\underline{\dot{\widetilde{W_{ij}}}} = {} & \frac{\left(\underline{\dot{\mu}_{1i}(x_1)\mu_{2j}(x_2)} + \underline{\mu_{1i}(x_1)\dot{\mu}_{2j}(x_2)}\right)\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}\right)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})^2} \\ & - \frac{\underline{(W_{ij})}\left[\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\underline{\dot{\mu}_{1i}(x_1)\mu_{2j}(x_2)} + \underline{\mu_{1i}(x_1)\dot{\mu}_{2j}(x_2)}\right)\right]}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})^2}\end{aligned} \tag{C.23}$$

Since $\underline{\widetilde{W_{ij}}} = \frac{\underline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}}$,

$$\dot{\underline{\widetilde{W_{ij}}}} = \frac{\Bigg(\underline{\dot{\mu}_{1i}(x_1)\mu_{2j}(x_2)} + \underline{\mu_{1i}(x_1)\dot{\mu}_{2j}(x_2)}\Bigg)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} - \frac{(\underline{\widetilde{W_{ij}}})\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(\underline{\dot{\mu}_{1i}(x_1)\mu_{2j}(x_2)} + \underline{\mu_{1i}(x_1)\dot{\mu}_{2j}(x_2)}\Bigg)\Bigg]}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \tag{C.24}$$

$$\begin{aligned}\dot{\underline{\widetilde{W_{ij}}}} = &\frac{\Bigg(-2A_{1i}(A_{1i})'\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)} - 2A_{2j}(A_{2j})'\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Bigg)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \\ &- \frac{(\underline{\widetilde{W_{ij}}})\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(-2A_{1i}(A_{1i})'\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Bigg)\Bigg]}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \\ &- \frac{(\underline{\widetilde{W_{ij}}})\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(-2A_{2j}(A_{2j})'\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Bigg)\Bigg]}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})}\end{aligned} \tag{C.25}$$

$$\dot{\underline{\widetilde{W_{ij}}}} = \frac{\Bigg(\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Bigg)\Bigg(-2A_{1i}(A_{1i})' - 2A_{2j}(A_{2j})'\Bigg)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} - \frac{(\underline{\widetilde{W_{ij}}})\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Bigg)\Bigg(-2A_{1i}(A_{1i})' - 2A_{2j}(A_{2j})'\Bigg)\Bigg]}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \tag{C.26}$$

$$\dot{\underline{\widetilde{W_{ij}}}} = -\underline{\widetilde{W_{ij}}}(\underline{K_{ij}})' + \underline{\widetilde{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\widetilde{W_{ij}}}(\underline{K_{ij}})' \tag{C.27}$$

$$\dot{\underline{\overline{\widetilde{W_{ij}}}}} = -\underline{\overline{\widetilde{W_{ij}}}}(\underline{\overline{K_{ij}}})' + \underline{\overline{\widetilde{W_{ij}}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\overline{\widetilde{W_{ij}}}}(\underline{\overline{K_{ij}}})' \tag{C.28}$$

where

$$(\underline{K_{ij}})' = 2\Big(A_{1i}(A_{1i})' + A_{2j}(A_{2j})'\Big)$$

$$(\overline{K_{ij}})' = 2\Big(U_{1i}(U_{1i})' + U_{2j}(U_{2j})'\Big)$$

By using the following Lyapunov function, the stability condition is checked as follows;

$$V_c = \frac{1}{2}\tau_c^2(t) \tag{C.29}$$

$$\dot{V}_c = \tau_c\dot{\tau}_c = \tau_c(\dot{\tau}_n + \dot{\tau}) \tag{C.30}$$

$$\tau_n = \frac{q\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\underline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}} + \frac{(1-q)\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\overline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}}} \tag{C.31}$$

$$\tau_n = q\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\underline{\widetilde{W_{ij}}} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\overline{\widetilde{W_{ij}}} \tag{C.32}$$

$$\dot{\tau_n} = q\sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f_{ij}}\underline{\widetilde{W_{ij}}} + f_{ij}\underline{\dot{\widetilde{W_{ij}}}}) + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f_{ij}}\overline{\widetilde{W_{ij}}} + f_{ij}\dot{\overline{\widetilde{W_{ij}}}}) \tag{C.33}$$

$$\dot{\tau_n} = q\sum_{i=1}^{I}\sum_{j=1}^{J}\left((-\widetilde{\underline{W_{ij}}}(\underline{K_{ij}})' + \widetilde{\underline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\underline{W_{ij}}}(\underline{K_{ij}})')f_{ij} + \widetilde{\underline{W_{ij}}}\dot{f_{ij}}\right)$$
$$+ (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\left((-\widetilde{\overline{W_{ij}}}(\overline{K_{ij}})' + \widetilde{\overline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}}(\overline{K_{ij}})')f_{ij} + \widetilde{\overline{W_{ij}}}\dot{f_{ij}}\right) \quad (C.34)$$

$$\dot{V_c} = \tau_c\dot{\tau_c} = \tau_c\Bigg[q\sum_{i=1}^{I}\sum_{j=1}^{J}[\left(-\widetilde{\underline{W_{ij}}}(\underline{K_{ij}})' + \widetilde{\underline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\underline{W_{ij}}}(\underline{K_{ij}})'\right)f_{ij} + \widetilde{\underline{W_{ij}}}\dot{f_{ij}}]$$
$$+ (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}[\left(-\widetilde{\overline{W_{ij}}}(\overline{K_{ij}})' + \widetilde{\overline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}}(\overline{K_{ij}})'\right)f_{ij} + \widetilde{\overline{W_{ij}}}\dot{f_{ij}}]$$
$$+ \dot{\tau}\Bigg] \quad (C.35)$$

$$\dot{V_c} = \tau_c\Bigg[q\sum_{i=1}^{I}\sum_{j=1}^{J}[\Bigg(-\widetilde{\underline{W_{ij}}}(2A_{1i}(A_{1i})' + 2A_{2j}(A_{2j})')$$
$$+ \widetilde{\underline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\underline{W_{ij}}}(2A_{1i}(A_{1i})' + 2A_{2j}(A_{2j})')\Bigg)f_{ij} + \widetilde{\underline{W_{ij}}}\dot{f_{ij}}]$$
$$+(1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg([-\widetilde{\overline{W_{ij}}}(2U_{1i}(U_{1i})' + 2U_{2j}(U_{2j})')$$
$$+ \widetilde{\overline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}}(2U_{1i}(U_{1i})' + 2U_{2j}(U_{2j})')]f_{ij} + \widetilde{\overline{W_{ij}}}\dot{f_{ij}}\Bigg)$$
$$+ \dot{\tau}\Bigg] \quad (C.36)$$

where

$$\dot{A_{1i}} = \frac{(\dot{x_1} - \dot{\underline{c_{1i}}})(\underline{\sigma_{1i}}) - (x_1 - \underline{c_{1i}})\dot{\underline{\sigma_{1i}}}}{\underline{\sigma_{1i}}^2}$$

$$\dot{A_{2j}} = \frac{(\dot{x_2} - \dot{\underline{c_{2j}}})(\underline{\sigma_{2j}}) - (x_2 - \underline{c_{2j}})\dot{\underline{\sigma_{2j}}}}{\underline{\sigma_{2j}}^2}$$

$$\dot{U_{1i}} = \frac{(\dot{x_1} - \dot{\overline{c_{1i}}})(\overline{\sigma_{1i}}) - (x_1 - \overline{c_{1i}})\dot{\overline{\sigma_{1i}}}}{\overline{\sigma_{1i}}^2}$$

$$\dot{U_{2j}} = \frac{(\dot{x_2} - \dot{\overline{c_{2j}}})(\overline{\sigma_{2j}}) - (x_2 - \overline{c_{2j}})\dot{\overline{\sigma_{2j}}}}{\overline{\sigma_{2j}}^2}$$

Equation (C.37) can be obtained by using Equations (4.11)-(4.16);

$$A_{1i}\dot{A_{1i}} = A_{2j}\dot{A_{2j}} = U_{1i}\dot{U_{1i}} = U_{2j}\dot{U_{2j}} = \alpha sgn(\tau_c) \tag{C.37}$$

$$\begin{aligned}\dot{V_c} = \tau_c\Big[& q\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(2\big[-\underline{\widetilde{W_{ij}}}f_{ij}2\alpha sgn(\tau_c) \\ & + \underline{\widetilde{W_{ij}}}f_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\widetilde{W_{ij}}}2\alpha sgn(\tau_c)\big] + \underline{\widetilde{W_{ij}}}\dot{f_{ij}}\Big) \\ & +(1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(2\big[-\overline{\widetilde{W_{ij}}}f_{ij}2\alpha sgn(\tau_c) \\ & + \overline{\widetilde{W_{ij}}}f_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{\widetilde{W_{ij}}}2\alpha sgn(\tau_c)\big]\overline{\widetilde{W_{ij}}}\dot{f_{ij}}\Big) \\ & + \dot{\tau}\Big]\end{aligned} \tag{C.38}$$

$$\begin{aligned}\dot{V_c} = \tau_c\Big[& q\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\big[-4\alpha sgn(\tau_c)\underline{\widetilde{W_{ij}}}f_{ij} \\ & + 4\alpha sgn(\tau_c)\underline{\widetilde{W_{ij}}}f_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\widetilde{W_{ij}}}\big] + \underline{\widetilde{W_{ij}}}\dot{f_{ij}}\Big) \\ & + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\big[-4\alpha sgn(\tau_c)\overline{\widetilde{W_{ij}}}f_{ij} \\ & + 4\alpha sgn(\tau_c)\overline{\widetilde{W_{ij}}}f_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{\widetilde{W_{ij}}}\big] + \overline{\widetilde{W_{ij}}}\dot{f_{ij}}\Big) \\ & + \dot{\tau}\Big]\end{aligned} \tag{C.39}$$

$$\dot{V}_c = \tau_c\Bigg[\Bigg(- q4\alpha sgn(\tau_c)\sum_{i=1}^{I}\sum_{j=1}^{J}[\underline{\widetilde{W_{ij}}}f_{ij} - \underline{\widetilde{W_{ij}}}f_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\widetilde{W_{ij}}}]$$
$$+ q\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\widetilde{W_{ij}}}\dot{f_{ij}} - (1-q)4\alpha sgn(\tau_c)\sum_{i=1}^{I}\sum_{j=1}^{J}[\overline{\widetilde{W_{ij}}}f_{ij} - \overline{\widetilde{W_{ij}}}f_{ij}\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{\widetilde{W_{ij}}}]$$
$$+ (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{\widetilde{W_{ij}}}\dot{f_{ij}}\Bigg)$$
$$+ \dot{\tau}\Bigg] \tag{C.40}$$

Since $\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{\widetilde{W_{ij}}} = 1$ and $\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\widetilde{W_{ij}}} = 1$, Equation (C.40) becomes as follow:

$$\dot{V}_c = \tau_c\Bigg[q\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\widetilde{W_{ij}}}\dot{f_{ij}} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{\widetilde{W_{ij}}}\dot{f_{ij}} + \dot{\tau}\Bigg] \tag{C.41}$$

$$\dot{V}_c = \tau_c\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(q\underline{\widetilde{W_{ij}}}\dot{f_{ij}} + (1-q)\overline{\widetilde{W_{ij}}}\dot{f_{ij}}\Bigg) + \dot{\tau}\Bigg] \tag{C.42}$$

$$\dot{V}_c = \tau_c\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f_{ij}}\Bigg(q\underline{\widetilde{W_{ij}}} + (1-q)\overline{\widetilde{W_{ij}}}\Bigg) + \dot{\tau}\Bigg] \tag{C.43}$$

$$\dot{V}_c = \tau_c\Bigg[- \alpha sgn(\tau_c) + \dot{\tau}\Bigg] \tag{C.44}$$

$$\dot{V}_c = \Bigg[- \alpha|\tau_c| + |\tau_c|B_{\dot{\tau}}\Bigg] < 0 \tag{C.45}$$

□

REFERENCES

1. Celikyilmaz, A. and I. B. Turksen, *Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications*, Springer, 2009.

2. Zadeh, L. A., "Fuzzy sets", *Inform Control*, Vol. 8, pp. 808–821, 1965.

3. Mamdani, E. H. and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", *International Journal of Man-Machine Studies*, pp. 1 –13, 1975.

4. Yen, J. and R. Langari, *Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications*, Prentice Hall Inc., Upper Saddle River, NJ,, 1999.

5. Yu, J.-S., S.-H. Kim, B.-K. Lee, C.-Y. Won and J. Hur, "Fuzzy-Logic-Based Vector Control Scheme for Permanent-Magnet Synchronous Motors in Elevator Drive Applications", *Industrial Electronics, IEEE Transactions on*, Vol. 54, No. 4, pp. 2190 –2200, 2007.

6. Dongbing, G. and H. Huosheng, "Integration of Coordination Architecture and Behavior Fuzzy Learning in Quadruped Walking Robots", *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Vol. 37, No. 4, pp. 670 – 681, July 2007.

7. Muyeen, S., R. Takahashi, T. Murata and J. Tamura, "Integration of an Energy Capacitor System With a Variable-Speed Wind Generator", *Energy Conversion, IEEE Transactions on*, Vol. 24, No. 3, pp. 740 –749, September 2009.

8. Mendel, J. M., *Uncertain Rule-Based Fuzzy Logic Systems*, Prentice Hall, Los Angeles, CA, 1998.

9. Oscar, C. and M. Patricia, *Type-2 Fuzzy Logic: Theory and Applications*, Springer,

Verlag Berlin Heidelberg, 2008.

10. Juang, C. and Y. Tsao, "A Type-2 Self-Organizing Neural Fuzzy System and Its FPGA Implementation", *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, pp. 1537 –1548, 2008.

11. Huang, S. and Y. Chen, "VLSI Implementation of Type-2 Fuzzy Inference Processor", *in: Proceedings of IEEE ISCAS 2005*, pp. 3307 –3310, May. 2005.

12. Hagras, H. A., "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots", *IEEE Transactions on Fuzzy Systems*, pp. 524 –539, 2004.

13. Biglarbegian, M., W. Melek and J. Mendel, "On the Stability of Interval Type-2 TSK Fuzzy Logic Control Systems", *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, Vol. 40, No. 3, pp. 798 –818, 2010.

14. Cao, J., P. Li and H. Liu, "An Interval Fuzzy Controller for Vehicle Active Suspension Systems", *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 11, No. 4, pp. 885 –895, December 2010.

15. Biglarbegian, M., W. W. Melek and J. M. Mendel, "Design of Novel Interval Type-2 Fuzzy Controllers for Modular and Reconfigurable Robots: Theory and Experiments", *Industrial Electronics, IEEE Transactions on*, Vol. 58, No. 4, pp. 1371 –1384, 2011.

16. Lin, P.-Z., C.-F. Hsu, T.-T. Lee and C.-H. Wang, "Robust Fuzzy-Neural Sliding-Mode Controller Design via Network Structure Adaptation", *Control Theory Applications, IET*, Vol. 2, No. 12, pp. 1054 –1065, December 2008.

17. Topalov, A. and O. Kaynak, "Online Learning in Adaptive Neurocontrol Schemes with a Sliding Mode Algorithm", *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 31, No. 3, pp. 445 –450, December 2001.

18. Topalov, A., K.-C. Kim, J.-H. Kim and B.-K. Lee, “Fast Genetic On-line Learning Algorithm for Neural Network and Its Application to Temperature Control”, *IEEE Int. Conf. Evolutionary Computation, Nagoya, Japan*, pp. 649 –654, May 1996.

19. Parma, G., B. Menezes and A. Braga, “Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks”, *Electronics Letters*, Vol. 34, No. 1, pp. 97 –98, 1998.

20. Shuanghe, Y., Y. Xinghuo and M. Zhihong, “A Fuzzy Neural Network Approximator with Fast Terminal Sliding Mode and Its Applications”, *Fuzzy Sets and Systems*, 2004.

21. Cigdem, O., E. Kayacan and O. Kaynak, “Experimental Study of an Interval Type-2 Fuzzy Neural Network Using Sliding-Mode Online Learning Algorithm”, *Control Conference (ASCC), 2011 8th Asian*, pp. 1181 –1186, May 2011.

22. Oniz, Y., E. Kayacan and O. Kaynak, “A Dynamic Method to Forecast the Wheel Slip for Antilock Braking System and Its Experimental Evaluation”, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, Vol. 39, No. 2, pp. 551 –560, April 2009.

23. Cascella, G., F. Cupertino, A. Topalov, O. Kaynak and V. Giordano, “Adaptive Control of Electric Drives Using Sliding-Mode Learning Neural Networks”, *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*, Vol. 1, pp. 125 – 130, 2005.

24. Efe, M. O., O. Kaynak and X. Yu, “Sliding Mode Control of a Three Degrees of Freedom Anthropoid Robot by Driving the Controller Parameters to an Equivalent Regime”, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 122, No. 4, pp. 632 –640, December 2000.

25. Kawato, M., Y. Uno, M. Isobe and R. Suzuki, “Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics”, *Control Systems Maga-*

zine, IEEE, Vol. 8, No. 2, pp. 8 –15, apr 1988.

26. Topalov, A., E. Kayacan, Y. Oniz and O. Kaynak, "Adaptive Neuro-Fuzzy Control with Sliding Mode Learning Algorithm: Application to Antilock Braking System", *Asian Control Conference, 2009. ASCC 2009. 7th*, pp. 784 –789, 27-29 2009.

27. Zadeh, L. A., "Toward Extended Fuzzy Logic-A First Step", *Fuzzy Sets and Systems*, Vol. 160, pp. 3175 – 3181, 2009.

28. Mamdani, E., "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthetic Synthesis", *IEEE Transaction on Computers*, Vol. 26, pp. 1182 – 1191, 1977.

29. Sugeno, M. and G. Kang, ""Fuzzy Identification of Systems and Its Applications to Modeling and Control", *IEEE Trans. Syst., Man, Cybern.*, Vol. 15, 1985.

30. Sugeno, M. and G. Kang, "Structure Identification of Fuzzy Model", *Fuzzy Sets and Systems*, Vol. 28, pp. 15 – 33, 1988.

31. Mamdani, E. and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", *International Journal of Man-machine studies*, pp. 419 – 435, 1990.

32. Liang, Q. and J. M. Mendel, "Interval Type-2 Fuzzy Logic Systems: Theory and Design", *IEEE Trans. on Fuzzy Systems*, Vol. 8, pp. 535–550, 2000.

33. Lin, F.-J. and P.-H. Chou, "Adaptive Control of Two-Axis Motion Control System Using Interval Type-2 Fuzzy Neural Network", *Industrial Electronics, IEEE Transactions on*, Vol. 56, No. 1, pp. 178 –193, January 2009.

34. Fuller, R., *Introduction to Neuro-fuzzy Systems (Advances in Soft Computing Series)*, Springer, 2000.

35. Jin, Y., *Advanced Fuzzy Systems Design and Applications*, Physica-Verlag, 2003.

36. Begian, M. B., W. W. Melek and J. M. Mendel, “Parametric Design of Stable Type-2 TSK Fuzzy Systems”, *Proceedings of the North American Fuzzy Information Processing Systems*, pp. 1 – 6, 2008.

37. Astrom, K. and B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1995.

38. Utkin, V., J. Guldner and J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*, Taylor and Francis Inc., Philadelphia, USA, 1999.

39. Utkin, V. I., *Sliding Modes in Control Optimization*, Springer-Verlag, 1992.

40. Slotine, J. J. and W. Li, *Applied Nonlinear Control*, Prentice Hall, 1991.

41. AMIRACorp., *DR300 Laboratory Setup Speed Control with Variable Load*, Tech. rep., AMIRA, 2000.