INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS: THEORY AND DESIGN

by

Erdal Kayacan

B.S., Electrical Engineering, Istanbul Technical University, 2003

M.S., Systems and Control Engineering, Bogaziçi University, 2006

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

Graduate Program in Electrical and Electronic Engineering

Boğaziçi University

2011

INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS: THEORY AND DESIGN

APPROVED BY:

Prof. Okyay Kaynak .....
(Thesis Supervisor)

Prof. Kemal Cılız .....

Assoc. Prof. Yağmur Denizhan .....

Prof. Fikret S. Gürgen .....

Prof. Hakan Temeltaş .....

DATE OF APPROVAL: 19.07.2011

ACKNOWLEDGEMENTS

First and foremost I would like to express my sincerest appreciation to my supervisor, Prof. Okyay Kaynak, who has guided me throughout my Ph.D. thesis with his patience and knowledge whilst allowing me to work in my own way. It was an honor for me to work with him during my time at Boğaziçi University.

I would like to thank Prof. Kemal Cılız and Prof. Hakan Temeltaş for serving as a member of my thesis committee and their academic guidance. I am grateful to Assoc. Prof. Yağmur Denizhan and Prof. Fikret S. Gürgen for serving on my jury.

I would also like to extend my sincerest gratitude to Dr. Mojtaba Ahmadieh Khanesar and Özkan Çiğdem for their contributions, suggestions, technical discussions and advisements which have helped to improve the quality of this thesis. In addition for their collaboration in various academic papers, a short break with them both forged life long friendships and served as a welcome time of rest.

I would like to thank Prof. Rahib Abiyev, Prof. Andon Topalov and Prof. Jim Tørresen for their academic guidance.

During the pursuit of my Ph.D. at Boğaziçi University, I have been lucky enough to meet many wonderful people who made the university a warm place for me. Many thanks to Dr. Fulya Kunter, Dr. Ebru Arısoy, Kerri Schultz, Başak Öztaş, Gamze Güzel, Zerrin Doğança, Pınar Akyazı, Deniz Seviş, Erinç Dikici, Hakan Karaoğuz, Haluk Bayram, Özgür Erkent, Dr. Sevgi Kesim Güven and Çeşminaz Eser for their friendship. I am indebted to my colleagues in Mechatronics Lab. for their patience and tolerance: Yeşim Öniz and Çisel Aras. My special thanks go to Dr. Janset Kuvulmaz.

Finally, there is no word to describe my gratitude toward my family for their endless support and love during my life.

ABSTRACT

INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS: THEORY AND DESIGN

This Ph.D. dissertation has four main objectives. Firstly, the noise reduction property of type-2 fuzzy logic systems that use a novel type-2 fuzzy membership function is studied. A number of papers exist in literature that claim the performance of type-2 fuzzy logic systems is better than that of type-1 fuzzy logic systems under noisy conditions, and this claim is supported by simulation studies only for some specific systems. In this dissertation, a simpler type-2 fuzzy logic system is considered with the novel membership function proposed in which the effect of input noise in the rule base is shown numerically in a general way. Secondly, fuzzy c-means clustering algorithm is proposed for type-2 fuzzy logic systems to determine the initial places of the membership functions to ensure that the gradient descent algorithm used afterwards converges in a shorter time. Thirdly, Levenberg-Marquardt algorithm is proposed for type-2 fuzzy neural networks. While conventional gradient descent algorithms use only the first order derivative, the proposed algorithm used in this dissertation benefits from the first and the second-order derivatives which makes the training procedure faster. Finally, a novel sliding mode control theory-based learning algorithm is proposed to train the parameters of the 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 for both Gaussian and triangular type-2 fuzzy membership functions, and the convergence of the weights is proven by Lyapunov stability method. The simulation results indicate that the type-2 fuzzy structure with the proposed learning algorithm results in a better performance than its type-1 fuzzy counterpart.

ÖZET

ARALIK DEĞERLİ TİP-2 BULANIK MANTIK SİSTEMLERİ: KURAM VE TASARIM

Bu doktora tezinin dört ana hedefi bulunmaktadır. Birincisi, yeni bir tip-2 bulanık mantık üyelik fonksiyonu kullanılarak tip-2 bulanık mantık sistemlerinin gürültü ile mücadele etme yetenekleri irdelenmiştir. Literatürde, tip-2 bulanık mantık sistemlerinin gürültülü ortamlarda tip-1 bulanık mantık sistemlere göre daha başarılı sonuçlar ortaya koyduğunu iddia eden çok sayıda yayın mevcuttur ve bu iddia bazı sistemler üzerinde benzetim çalışmaları yapılarak desteklenmeye çalışılmıştır. Bu doktora tezinde ise, yeni bir tip-2 üyelik fonksiyonu önerilmiş ve basit bir tip-2 bulanık mantık sistemi ile kural tabanındaki gürültünün etkisi daha genel bir bakış açısı ile nümerik olarak gösterilmiştir. İkincisi, Gradyan tabanlı algoritmanın daha hızlı yakınsamasını garanti etmek için, tip-2 bulanık mantık sistemlerindeki üyelik fonksiyonlarının parametrelerinin ilk değerlerlerinin belirlenmesi amacıyla bulanık c-ortalamalar kümeleme algoritması kullanılmıştır. Üçüncüsü, tip-2 bulanık mantık sistemler için Levenberg-Marquardt algoritması önerilmiştir. Geleneksel Gradyan tabanlı algoritmalar sadece birinci türevi kullanırken, önerilen bu algoritma birinci ve ikinci türevleri kullanmaktadır ve bu durum öğrenme işlemini hızlandırmaktadır. Son olarak, tip-2 nüro bulanık sistemler için yeni bir kayma kipli denetim tabanlı öğrenme algoritması önerilmiştir. Bu yaklaşımda, bir hata fonksiyonunun azaltılmasının denenmesi yerine, ağ ağırlıkları hata fonksiyonu bir kararlı denklemi sağlayacak şekilde ayarlanmıştır. Parametre güncelleme kuralları, Gauss ve üçgensel tip-2 bulanık üyelik fonksiyonları için ayrı ayrı çıkartılmış ve ağ ağırlıklarının yakınsaması Lyapunov kararlılık metodu ile ispat edilmiştir. Benzetim çalışmaları göstermiştir ki önerilen yeni öğrenme algoritmasını kullanan tip-2 bulanık mantık yapısı tip-1 bulanık mantık yapısına göre daha yüksek bir başarım elde etmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_1	The parameter of the ellipsoidal type-2 membership function
a_2	The parameter of the ellipsoidal type-2 membership function
A	A type-1 fuzzy set
$\widetilde{A}$	A type-2 fuzzy set
b	Consequent parameters (crisp) of the first-order type-1 TSK model
B	Consequent parameters (type-1) of the first-order type-2 TSK model
B_m	Viscous damping coefficient
c	The center of a Gaussian (or triangular) membership function
C_d	Discharge coefficient
C_f	Dimensionless internal friction coefficient
C_{im}	Internal or cross-port leakage coefficient of the motor
d	The width of a triangular membership function
e	The feedback error
$\dot{e}$	The change rate of the feedback error
E	Induced electromotive force
f_j	The firing strength of the type-1 fuzzy sets
F_j	The firing strength of the type-2 fuzzy sets
I	Identity matrix
I_A	Armature current
J	Moment of inertia of the system
J_x	Primary membership of x
K	Number of samples
K_I	Integral parameter
K_M	Torque constant
K_P	Proportional parameter
K_q	Valve flow gain
K_r	Valve gain

L_A	Armature winding inductance
m	The number of inputs in a TSK model
M	The torque produced by the motor
M_B	Acceleration torque
M_L	Load torque
P_s	Supply pressure
q_m	Volumetric displacement of the motor
R_A	Armature winding resistance
r_j	Firing of each rule in the type-2 fuzzy logic system
S_c	The zero adaptive learning error
S_p	The sliding line
T_A	Electrical time constant
T_M	Mechanical time constant
T_r	Valve time constant
u	The output of the first order type-1 TSK model
U	The output of the first order type-2 TSK model
U_A	Armature terminal voltage
u_l	The minimum values of u
u_r	The maximum values of u
V_p	Lyapunov function
V_0	Average contained volume of each motor chamber
W	Surface gradient
$alpha$	Learning rate
β_e	Effective hulk modulus of the system
χ	The constant determining the slope of the sliding surface
η	Noise added to the system
$\mu(x)$	Type-1 membership grade
$\tilde{\mu}(x)$	Type-2 membership grade
ω	Consequent parameters of the first-order type-1 TSK model
Ω	Consequent parameters of the first-order type-2 TSK model
ρ	Oil density

$\sqcap$	Meet operator
σ	The standard deviation of a Gaussian membership function
τ	The input signal of the system to be controlled
τ_c	The output signal of the PI controller
τ_n	The output signal of the fuzzy neural network
τ_j	T-norm operator

LIST OF ACRONYMS/ABBREVIATIONS

ANN	Artificial Neural Network
BP	Back Propagation
BIBO	Bounded Input Bounded Output
DCN	Distortion Caused by Noise
FOU	Footprint of Uncertainty
FLC	Fuzzy Logic Controller
FLS	Fuzzy Logic System
FNN	Fuzzy Neural Network
GD	Gradient Descent
LM	Levenberg-Marquardt
MSE	Mean Squared Error
PID	Proportional-Derivative-Integral
PWM	Pulse Width Modulation
RMSE	Root Mean Squared Error
SNR	Signal Noise Ratio
SMC	Sliding Mode Control
SSE	Squared of the Error
TSK	Takagi-Sugeno-Kang
T1FLC	Type-1 Fuzzy Logic Controller
T1FLS	Type-1 Fuzzy Logic System
T1FNN	Type-1 Fuzzy Neural Network
T2FLC	Type-2 Fuzzy Logic Controller
T2FLS	Type-2 Fuzzy Logic System
T2FNN	Type-2 Fuzzy Neural Network

1. INTRODUCTION

1.1. Motivation

If the parameters of a system can be obtained precisely, then its control would be a relatively straightforward problem and model-based approaches such as PID and pole placement could be used. However, in real-time industrial systems, it is often the case that there exist considerable difficulties in obtaining an accurate model. Even when the model is sufficiently accurate, there are many other uncertainties for example due to the precision of the sensors, noise produced by the sensors, environmental conditions of the sensors, and nonlinear characteristics of the actuators. Then, not only does the performance of the model-based approaches drastically decrease, but the complexity of the controller design also increases. In such cases, model-free approaches are generally preferred both for modeling and control purposes. The most common model-free approaches are the use of artificial neural networks (ANNs) and fuzzy logic systems (FLSs).

Zadeh makes the statement 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"* [1]. There are two different approaches to FLSs design: Type-1 FLSs (T1FLSs) and type-2 FLSs (T2FLSs). The latter is proposed as an extension of the former with the intention of being able to model the uncertainties that invariably exist in the rule base of the system [2]. In type-1 fuzzy sets membership functions are totally certain, whereas in type-2 fuzzy sets membership functions are themselves fuzzy. The latter case results in that the antecedents and the consequents of the rules are uncertain. While a type-1 membership grade is a crisp number in [0,1], a type-2 membership grade can be any subset in [0,1] which is called *primary membership*. Additionally, there is a *secondary membership* value corresponding to each primary membership value that defines the possibility for primary member-

ships [3]. Whereas the secondary membership functions can take values in the interval of [0,1] in generalized T2FLSs, they are uniform functions that only take on values of 1 in interval T2FLSs. Since the general T2FLSs are computationally very demanding (this is because the type-reduction is computationally expensive), the use of interval T2FLSs is more commonly seen in literature, due to the fact that the computations are more easily manageable. Due to these factors, interval T2FLSs are investigated in this dissertation.

The noise reduction property of T2FLSs that use a novel type-2 fuzzy membership function (ellipsoidal type-2 membership function) is studied in this dissertation. The proposed type-2 membership function has certain values on both ends of the support and the kernel, and some uncertain values for the other values of the support. In this part of the dissertation, the parameter tuning rules of a T2FLS that uses such a membership function are derived using the gradient descent (GD) learning algorithm. There exists a number of papers in literature which claim that the modeling and control performance of T2FLSs is better than T1FLSs under noisy conditions. This is attempted to be shown via simulation (or real-time) studies only for some specific systems. However in this dissertation, a simpler T2FLS is considered with the novel membership function proposed in which the effect of input noise in the rule base is shown numerically in a general way. The proposed type-2 fuzzy neuro structure is tested on different input-output data sets, and it is shown that the T2FLS in combination with the proposed novel membership function has a better noise reduction property when compared to its type-1 counterpart.

Fuzzy neural networks (FNNs) combine the capability of fuzzy reasoning to handle uncertain information and the capability of ANNs to learn from input-output data sets. Therefore they have become a popular approach in engineering fields [4]. For tuning the parameters of FNNs, the gradient based algorithm 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 [5]. In addition to these drawbacks, the tuning process can easily be trapped into a local minimum [6]. To alleviate the problems mentioned, the use of evolutionary approaches have been suggested [7]. 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, sliding mode control (SMC) theory-based algorithms are proposed for the parameter update rules of ANNs and type-1 fuzzy neural networks (T1FNNs) as robust learning algorithms [8, 9]. SMC theory-based learning algorithms cannot only make the overall system more robust but also ensure faster convergence than the traditional learning techniques in online tuning of ANNs and FNNs [10, 11]. Motivated by the successful results of these learning algorithms on T1FNNs, a further contribution of this dissertation is the derivation of SMC theory-based learning algorithms for the training of type-2 FNNs (T2FNNs).

Prior to 1992, in the design procedure of FLSs, the locations and the spreads of the membership functions were generally chosen by the designer with perhaps some inputs from some experts [2]. Since then, a huge number of papers have been published which are based on the adaptation of the parameters of FLSs using the training data, some early examples being [12] and [13]. In the first stage of the training process, the locations of the membership functions are randomly distributed guaranteeing the coverage of the entire universe of discourse. On the other hand, fuzzy c-means clustering algorithm may be used to determine the initial places of the membership functions to ensure that the GD algorithm used afterwards converges in a shorter time. The use a fuzzy c-means clustering algorithm for T2FLSs is another topic studied in this dissertation.

Gradient-based methods are widely used to train FLSs in literature. In order to make the training procedure faster, some other techniques that use the second order derivatives, e.g. Gauss-Newton's method and Levenberg-Marquardt (LM) algorithm, are used to train FLSs in literature. In this dissertation, a training algorithm based on the LM method is proposed for T2FNNs. This approach is more robust than the other techniques that use the second order derivatives, e.g. Gauss-Newton's method.

1.2. Contributions of This Dissertation

This dissertation concentrates mainly on the adaptation of SMC theory-based learning algorithms to T2FLSs and the analysis of the noise reduction property of T2FLSs. The major and the minor contributions achieved are listed in more detail as follows:

The major contributions to the literature:

- The idea of using SMC theory-based learning algorithms in order to train T1FNNs is extended to T2FNNs. The parameter update rules of T2FNNs using SMC theory are derived (both for Gaussian and triangular type-2 fuzzy membership functions), and the proofs of the learning algorithms are shown using Lyapunov sense of stability.
- The noise reduction property of T2FLSs that use a novel type-2 fuzzy membership function (ellipsoidal type-2 membership function) is studied, and it is shown that the T2FLS with the proposed novel membership function has better noise reduction property as compared to the type-1 counterparts.

The minor contributions to the literature:

- The use of fuzzy c-means classification algorithm in order to determine the initial parameters of the type-2 fuzzy membership functions is proposed.
- A new training approach based on the LM algorithm is proposed for T2FNNs.

1.3. Outline of the Dissertation

In Chapter 2, the basic concepts of type-2 fuzzy sets and type-2 fuzzy logic theory, which are required to explain the proposed algorithms, are presented.

In Chapter 3, type-1 and type-2 Takagi-Sugeno-Kang (TSK) fuzzy models are given. Model II is the most common type-2 TSK model in literature, and it is also called "A2-CO" model. To be able to explain the Model II better, a numerical example is given.

In Chapter 4, the noise reduction property of T2FLSs that use a novel type-2 fuzzy membership function is studied. The proposed type-2 membership function has certain values on both ends of the support and the kernel, and some uncertain values for the other values of the support. The parameter tuning rules of a T2FLS that uses such a membership function are derived using the GD learning algorithm.

In Chapter 5, a type-2 TSK FNN is proposed and its parameter update rules are derived using GD learning algorithm. Fuzzy c-means clustering algorithm is used to determine the initial places of the membership functions to ensure that the GD algorithm used afterwards converges in a shorter time.

In Chapter 6, a new training approach based on the Levenberg-Marquardt algorithm is proposed for T2FNNs.

In Chapter 7, the SMC theory-based approach for on-line learning as applied to T2FNNs are introduced. In the approach, instead of trying to minimize an error function, the parameters of the network are tuned by the proposed algorithm in such a way that the learning error is enforced to satisfy a stable equation. The update rules to achieve this are derived, and the convergence of the parameters is proved by Lyapunov stability method.

In the final Chapter, Chapter 8, the conclusions and remarks are given.

2. FUZZY LOGIC SYSTEMS

2.1. Introduction

The fuzzy theory was first introduced into the scientific literature in 1965 by Professor Lotfi A. Zadeh at the University of California at Berkeley who proposed a set theory that operated over the range [0; 1]. He published a paper titled "Fuzzy Sets" in the journal Information and Control [14]. While Boolean logic results are restricted to 0 and 1, fuzzy logic results are between 0 and 1. In other words, fuzzy logic defines some intermediate values between sharp evaluations like absolute true and absolute false. This means that fuzzy sets can handle some concepts that we commonly meet in daily life, like "very old", "old", "young", "very young". Fuzzy logic is more like human thinking because it is based on degrees of truth and uses linguistic variables.

Fuzzy logic was not an acceptable theory for the scientists at that time because it contained vagueness in the engineering field. However, since 1970s, this approach to set theory has been widely applied to control systems. The principles of fuzzy logic were used to control a steam engine by Ebraham Mamdani of University of London in 1974 [15]. It was a milestone for fuzzy logic. The first industrial application was a cement kiln built in Denmark in 1975. In the 1980s, Fuji Electric applied fuzzy logic theory to the control of a water purification process. As a challenging engineering project, in 1987, Sendai Railway system that had automatic train operation control was built with fuzzy logic principles in Japan. Fuzzy control techniques were used in all the critical operations in the control of the train, such as accelerating, breaking, and stopping operations. In 1987, Takeshi Yamakawa used fuzzy control in an inverted pendulum experiment which is a classical control problem. After these successful applications, not only the engineers but also the social scientists applied fuzzy logic into different areas. In today's technology, many companies use fuzzy logic in their engineering projects like for example air conditioners, video cameras, televisions, washing machines, bus time tables, medical diagnoses, antilock braking system, etc.

Classical control theory, typically PID controller, uses a mathematical model to define the relationship between the inputs and the outputs of a system. The most serious disadvantage of these controllers is that they usually assume the system to be linear or at least that it behaves as a linear system in some range. If an accurate mathematical model of a system is available, a conventional PID controller can make the performance of the system quite acceptable. However, in real life, an accurate mathematical model of a control process is not generally available, even it may not exist. The real world is nonlinear, uncertain and contains always incomplete data. If the mathematical model is not known by the designer, there is no way to come up with a proper PID controller design. Even in those cases, when the mathematical model is known to be relatively accurate, the parameters of the system are likely to change by some external factors, like heat or pressure, etc. In such cases, a model-free approach is preferable. Fortunately, fuzzy logic controllers (FLCs) have the ability to control a system using some limited expert knowledge. In most cases, the design procedure of a FLC tries to imitate an expert or a skilled human operator. Besides, FLCs are low-cost implementations based on cheap sensors.

Type-2 fuzzy sets were introduced by Zadeh in 1975 as an extension of type-1 fuzzy sets. Mendel and Karnik have developed the theory of type-2 fuzzy sets further in [16]. The theoretical background of interval type-2 fuzzy system and its design principles are described in [17]. T2FLSs appear to be a more promising method than their type-1 counterparts for handling uncertainties such as noisy data and changing environments [18, 19]. In [20] and [21] the effects of the measurement noise in type-1 and type-2 FLCs (T2FLCs) and identifiers are simulated to perform a comparative analysis. It is concluded that the use of T2FLCs in real world applications [22] which exhibit measurement noise and modeling uncertainties can be a better option than type-1 FLCs (T1FLCs).

When a system has large amount of uncertainties, T1FLSs may not be able to achieve the desired level of performance with a reasonable complexity of the structure [2]. In such cases, the use of T2FLSs is suggested as the preferable approach in the literature in many areas, such as forecasting of time-series [23], controlling of mobile

robots [24], and the truck backing-up control problem [25] and the VLSI and FPGA implementations of T2FLSs are discussed in [26] and [27]. In [23], it is shown that when the parameters are tuned properly, T2FLSs can result in a better ability to predict as compared to T1FLSs. In [24], T2FLSs are applied to real time mobile robots for indoor and outdoor environments. The real time implementation studies show that a traditional T1FLC cannot handle the uncertainties in the system 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). In [25], the authors construct an interval T2FNN structure, and show that a better performance can be obtained as compared to a conventional T1FNN. The VLSI implementation of T2FLSs is also discussed in literature and it is shown that the inference speed can be sufficiently high for real time applications. In [27], a type-2 self-organizing neural fuzzy 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.

In general, fuzzy logic is a nonlinear mapping of an input data vector into a scalar output [2]. There are two main approaches to design of a FLC in literature:

(i) Type-1 fuzzy sets: Membership functions are totally certain.
(ii) Type-2 fuzzy sets: Membership functions that are themselves fuzzy. This results the antecedents and consequents of the rules are uncertain.

2.2. Type-1 Fuzzy Sets

A type-1 fuzzy set, A, which is in terms of a single variable, $x \in X$, may be represented as:

$$A = \{(x, \mu_A(x))| \qquad \forall x \in X\} \tag{2.1}$$

A can also be defined as:

$$A = \int_{x \in X} \mu_A(x)/x \tag{2.2}$$

where $\int$ denotes union over all admissible x.

As can be seen from Figure 2.1, a type-1 Gaussian membership function, $\mu_A(x)$, is constrained to be between 0 and 1 for all $x \in X$, and is a two-dimensional function. This type of membership function does not contain any uncertainty. In other words, there exits a clear membership value for every input data point.

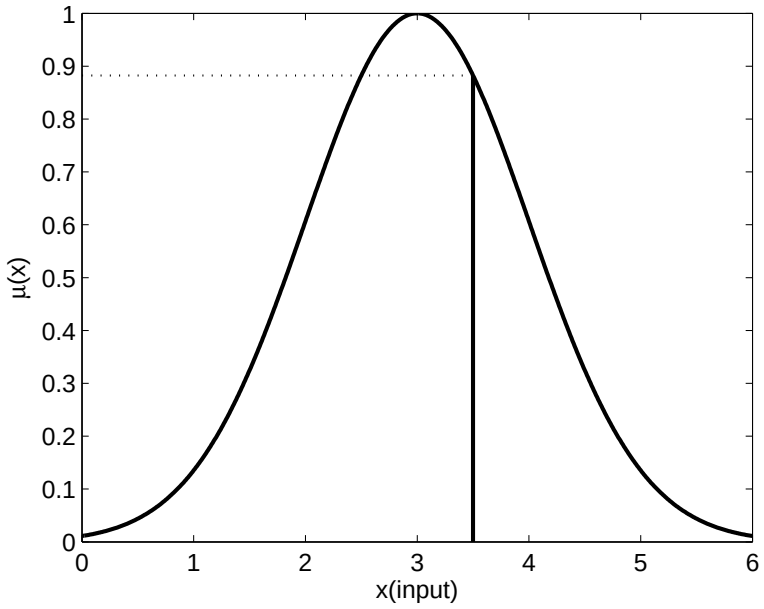


Figure 2.1. A Gaussian type-1 fuzzy membership function.

2.3. Type-2 Fuzzy Sets

A type-2 fuzzy set, $\tilde{A}$, may be represented as [28]:

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u))| \qquad \forall x \in X \qquad \forall u \in J_x \subseteq [0,1]\} \tag{2.3}$$

where $\mu_{\tilde{A}}(x,u))$ is the type-2 fuzzy membership function in which $0 \leq \mu_{\tilde{A}}(x,u)) \leq 1$. $\tilde{A}$ can also be defined as [28]:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \qquad J_x \subseteq [0,1] \tag{2.4}$$

where $\int \int$ denotes union over all admissible x and u [28].

J_x is called primary membership of x [28]. Additionally, there is a *secondary membership* value corresponding to each primary membership value that defines the possibility for primary memberships [3]. Whereas the secondary membership functions can take values in the interval of [0,1] in generalized T2FLSs, they are uniform functions that only take on values of 1 in interval T2FLSs. Since the general T2FLSs are computationally very demanding, the use of interval T2FLSs is more commonly seen in the literature, due to the fact that the computations are more manageable.

If the circumstances are so fuzzy, the places of the membership functions may not be determined precisely. In such cases, the membership grade cannot be determined as a crisp number in [0,1], then the use of type-2 fuzzy sets might be a preferable option.

If the standard deviation of the Gaussian function in Figure 2.1 is blurred, Figure 2.2 can be obtained. In Figure 2.2, the membership function does not have a single value for a specific value of x. The values that the vertical line intersects the membership functions do not need all be weighted same. Moreover, an amplitude distribution can be assigned to all of those points. Hence, a three-dimensional membership function- a type-2 membership function- that characterizes a type-2 fuzzy set is created if the amplitude distribution operation is done for all $x \in X$ [2].

The footprint of uncertainty (FOU), the union of all primary memberships, is said to be the bounded region that represents the uncertainty in the primary memberships of a type-2 fuzzy set (Figure 2.2). An upper membership function and a lower membership function are two type-1 membership functions that are the bounds for the FOU of a type-2 fuzzy set [2].

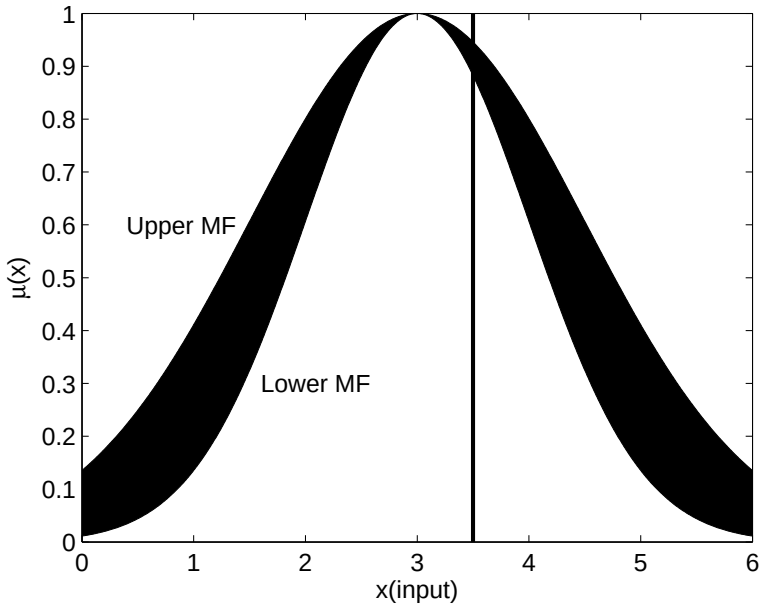


Figure 2.2. A Gaussian type-2 fuzzy membership function (FOU).

2.3.1. Interval Type-2 Fuzzy Sets

When all $\mu_{\tilde{A}}(x,u)$ are equal to 1, then $\tilde{A}$ is an interval T2FLS. The special case of Equation 2.4 might be defined for the interval T2FLSs:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} 1/(x,u) \qquad J_x \subseteq [0,1] \tag{2.5}$$

The researchers are familiar to the computational burden of general T2FLS. Hence, interval T2FLSs are commonly used in literature. Both the general and interval type-2 fuzzy membership functions are three-dimensional. As can be seen from Figure 2.3, the only difference between them is that the secondary membership value of the interval type-2 membership function is always equal to 1.

In this dissertation, the research activities are focused on the interval T2FLSs.

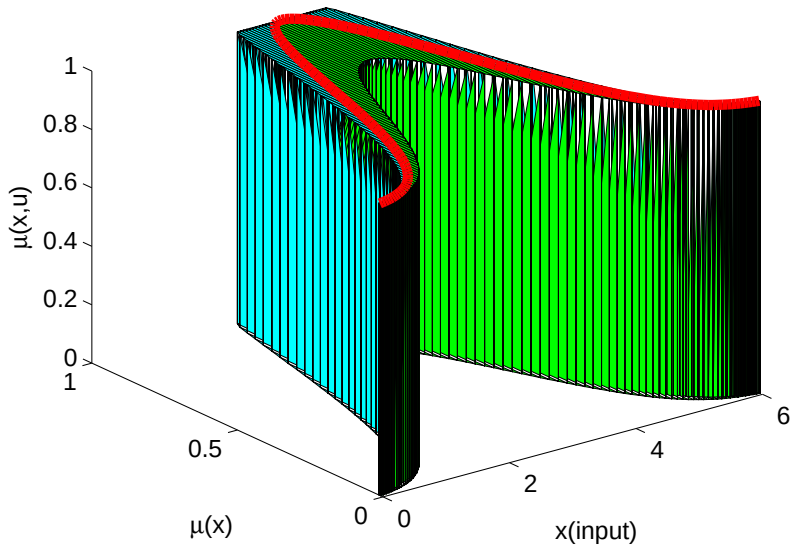


Figure 2.3. Three-dimensional representation of interval type-2 fuzzy membership functions.

2.4. Type-2 Fuzzy Logic System Block Diagram

Both the type-1 and type-2 FLSs are shown in Figure 2.4 and Figure 2.5, respectively. The reader is supposed to have a basic knowledge of T1FLSs, only the similarities and the differences between T1FLSs and T2FLSs will be given in this dissertation. As can seen from Figure 2.5, an additional block (type reduction) is needed in type-2 FLS design. Although the structure in Figure 2.5 brings some advantages when dealing with uncertainties, it also increases the computational burden.

The followings are the basic blocks of a T2FLS:

Fuzzifier: The fuzzifier maps crisp inputs into type-2 fuzzy sets which activates the inference engine.

Rule base: The rules in a T2FLS remains the same as in T1FLS, but antecedents and consequents are represented by interval type-2 fuzzy sets.

Inference: Inference block assigns fuzzy inputs to fuzzy outputs using the rules in the rule base and the operators such as union and intersection. In type-2 fuzzy sets, *join* ($\sqcup$) and *meet* operators ($\sqcap$), which are new concepts in fuzzy logic theory, are used instead of *union* and *intersection* operators. These two new operators are used in secondary membership functions, and they are defined and explained in detail in [29].

Type-reduction: The type-2 fuzzy outputs of the inference engine are transformed into type-1 fuzzy sets that are called *the type-reduced sets*. There are two common methods for the type-reduction operation in the interval T2FLSs: One is the Karnik-Mendel iteration algorithm, and the other is Wu-Mendel uncertainty bounds method. These two methods are based on the calculation of the centroid.

Defuzzification: The outputs of the type reduction block are given to defuzzificaton block. The type-reduced sets are determined by their left end point and right end point, the defuzzified value is calculated the average of these points.

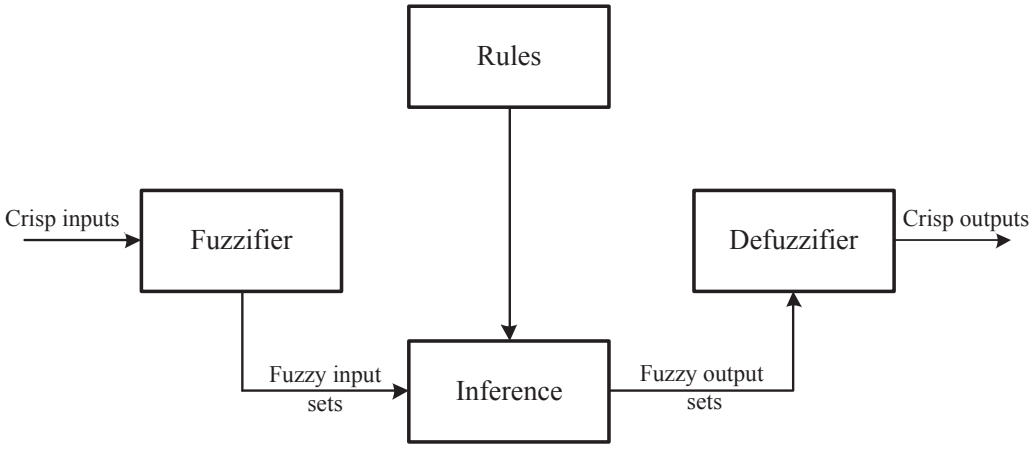


Figure 2.4. T1FLS block diagram.

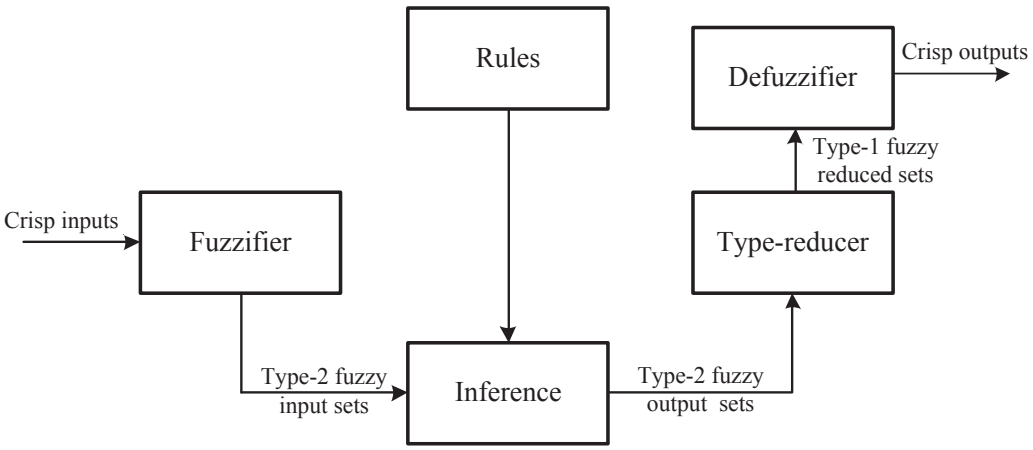


Figure 2.5. T2FLS block diagram.

3. TSK FUZZY LOGIC SYSTEMS

3.1. Introduction

TSK fuzzy logic models are introduced in 1984 by T. Takagi, M. Sugeno and K. T. Kang. Instead of using fuzzy sets in the consequent part (like in Mamdani models), TSK model uses a function of the input variables. The order of the function determines the order of the model, e.g. zeroth-order TSK model, first-order TSK model, etc.

3.2. Type-1 TSK FLS

A type-1 TSK model can be described by fuzzy IF-THEN rules. For instance, in a first-order type-1 TSK model, the rule base is as follows:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{j1} \text{ and } x_2 \text{ is } A_{j2} \text{ and } \ldots \text{and } x_n \text{ is } A_{jn} \\ &\qquad\qquad \text{THEN } u_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \end{aligned} \tag{3.1}$$

where x_1, x_2, ... ,x_n are the input variables, u_j's are the output variables, $\mathrm{A_{ji}}$'s are type-1 membership functions for j^{th} rule and the i^{th} input. The parameters in the consequent part of the rules are w_{ij} and b_j $(i = 1, \ldots, n, j = 1, \ldots, M)$.

The final output of the system can be written as:

$$u = \frac{\sum_{j=1}^{M} f_j u_j}{\sum_{j=1}^{M} f_j} \tag{3.2}$$

where f_j is given by:

$$f_j(x) = \mu_{A_{j1}}(x_1) * \ldots * \mu_{A_{jn}}(x_n) \tag{3.3}$$

in which $*$ represents the t-norm which is the *prod* operator in this study.

3.3. Other TSK Models

Other TSK models (shown in Table 3.1) can be classified into three groups [30]:

(i) Model I: Antecedents are type-2 fuzzy sets, and consequents are type-1 fuzzy sets (A2-C1)

(ii) Model II: Antecedents are type-2 fuzzy sets, and consequents are crisp numbers (A2-C0)

(iii) Model III: Antecedents are type-1 fuzzy sets, and consequents are type-1 fuzzy sets (A1-C1)

Table 3.1. Classification of other TSK Models.

Other TSK FLSs	**Model I**	**Model II**	**Model III**
Antecedent	type-2 fuzzy sets	type-2 fuzzy sets	type-1 fuzzy sets
Consequent	type-1 fuzzy sets	crisp numbers	type-1 fuzzy sets

3.3.1. Type-2 FLS Model I

Type-2 Model I can be described by fuzzy IF-THEN rules. The antecedent part is type-2 fuzzy sets. In the consequent part, the structure is similar to that of type-1 TSK fuzzy system, however, the parameters are type-1 fuzzy sets rather than numbers. They are therefore named as "Type-2 TSK Model I" systems. In the Model I, the rule base is as follows:

$$\begin{aligned} &\text{IF } x_1 \text{ is } \tilde{A}_{j1} \text{ and } x_2 \text{ is } \tilde{A}_{j2} \text{ and } \ldots \text{and } x_n \text{ is } \tilde{A}_{jn} \\ &\text{THEN } U_j = \sum_{i=1}^{n} W_{ij} x_i + B_j \end{aligned} \tag{3.4}$$

where x_1, x_2, ... ,x_n are the input variables, U_j's are the output variables, $\tilde{\text{A}}_{\text{ji}}$'s are type-2 membership functions for j^{th} rule and the i^{th} input. The parameters in the consequent part of the rules are W_{ij} and B_j $(i = 1, \ldots, n, j = 1, \ldots, M)$ which are type-1 fuzzy

sets. The final output of the first-order type-2 TSK Model I is as follows [30]:

$$U(U_1, ...U_M, F_1, ...F_M) = \int_{u_1} ... \int_{u_M} \int_{f_1} ... \int_{f_M} \tau_{j=1}^{M} \mu_{U_j}(u_j) \star \tau_{j=1}^{M} \mu_{F_j}(f_j) / \frac{\sum_{j=1}^{M} f_j u_j}{\sum_{j=1}^{M} f_j} \tag{3.5}$$

where M is the number of rules fired, $u_j \in U_j$, $f_j \in F_j$, and τ and $\star$ indicate the t-norm.

F_j is the firing strength which is defined as:

$$F_j = \mu_{\tilde{A}_{j1}}(x_1) \sqcap \mu_{\tilde{A}_{j2}}(x_2) \sqcap ... \sqcap \mu_{\tilde{A}_{jn}}(x_n) \tag{3.6}$$

where $\sqcap$ shows the meet operation.

Although the calculation of Equation 3.5 is very difficult, some general concepts are explained in [31]. When interval type-2 sets are used in the antecedent part and type-1 sets are used in the consequent parts, it is shown in [30] that the output of an interval T2FLS is:

$$f(x) = \frac{u_l + u_r}{2} \tag{3.7}$$

where u_r and u_l are the maximum and minimum values of u, respectively.

The reader is encouraged to refer [30] and [31] for further information about the calculation process of u_r and u_l.

3.3.2. Type-2 TSK FLS Model II

Model II can be regarded as a special case of Model I where the antecedents are type-2 fuzzy sets and the consequents are polynomials. A type-2 TSK Model II can be described by fuzzy IF-THEN rules. For instance, in a first-order type-2 TSK Model II,

the rule base is as follows [30]:

$$\text{IF } x_1 \text{ is } \tilde{A}_{j1} \text{ and } x_2 \text{ is } \tilde{A}_{j2} \text{ and } \ldots \text{and } x_n \text{ is } \tilde{A}_{jn} \quad \text{THEN } u_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \tag{3.8}$$

where x_1, x_2, ... ,x_n are the input variables, u_j's are the output variables, $\tilde{A}_{ji}$'s are type-2 membership functions for j^{th} rule and the i^{th} input. The parameters in the consequent part of the rules are w_{ij} and b_j $(i = 1, \ldots, n, j = 1, \ldots, M)$.

The final output of the model is as follows [30]:

$$U(F_1, ...F_M) = \int_{f_1} ... \int_{f_M} \tau_{j=1}^{M} \mu_{F_j}(f_j) / \frac{\sum_{j=1}^{M} f_j u_j}{\sum_{j=1}^{M} f_j} \tag{3.9}$$

where M is the number of rules fired, $f_j \in F_j$, and τ indicates the t-norm.

It is to be noted that Equation 3.9 is the special case of Equation 3.5, because each U_j is a crisp value.

The firing strength is the same as Equation 3.6.

3.3.2.1. Interval Type-2 TSK FLS. In the structure of interval type-2 TSK FLS, Equation 3.9 is given as follows [32]:

$$Y_{TSK/A2-C0} = \int_{f^1 \in [\underline{f}^1, \overline{f}^1]} ... \int_{f^M \in [\underline{f}^M, \overline{f}^M]} 1 / \frac{\sum_{j=1}^{M} f_j u_j}{\sum_{j=1}^{M} f_j} \tag{3.10}$$

where $\underline{f}^i$ and $\overline{f}^i$ are given by:

$$\underline{f}_j(x) = \underline{\mu}_{A_{j1}}(x_1) * ... * \underline{\mu}_{A_{jn}}(x_n) \qquad \overline{f}_j(x) = \overline{\mu}_{A_{j1}}(x_1) * ... * \overline{\mu}_{A_{jn}}(x_n) \tag{3.11}$$

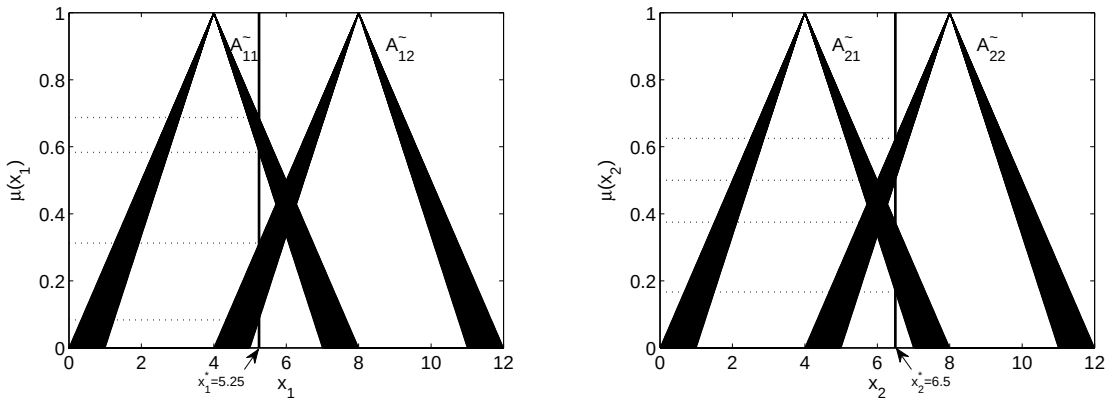
in which $*$ represents the t-norm which is the *product* operator in this study.

The output of the fuzzy system in closed form is obtained by [32]:

$$Y_{TSK/A2-C0} = \frac{\sum_{j=1}^{M} \underline{f}_j u_j}{\sum_{j=1}^{M} \underline{f}_j + \sum_{j=1}^{M} \overline{f}_j} + \frac{\sum_{j=1}^{M} \overline{f}_j u_j}{\sum_{j=1}^{M} \underline{f}_j + \sum_{j=1}^{M} \overline{f}_j} \tag{3.12}$$

3.3.2.2. Numerical Example for the Interval Type-2 TSK FLS. In this Ph.D. dissertation, A2-CO TSK model is used. In order to be able to give a clear explanation about the inference of this type FLS, a numerical example has been given:

Let's assume a T2FLS (A2-CO) with two inputs and two type-2 fuzzy membership functions for each. While the antecedent type-2 fuzzy membership functions are given in Figure 3.1, the consequent part of the rules are given as follows: $u_1 = 4x_1 + x_2$ and $u_2 = 2x_1 + 3x_2$.



(a) input 1 (x_1) (b) input 2 (x_2)

Figure 3.1. Two rules each having two type-2 triangular fuzzy membership functions.

The mathematical form of triangular membership functions are as follows:

$$\tilde{\mu}(x) = \begin{cases} 1 - \frac{|x-c|}{d} & \text{if } c-d < x < c+d \\ 0 & else \end{cases} \tag{3.13}$$

where $\underline{c}_{11} = \underline{c}_{21} = \overline{c}_{11} = \overline{c}_{21} = 4$, $\underline{c}_{12} = \underline{c}_{22} = \overline{c}_{12} = \overline{c}_{22} = 8$, and $\underline{d}_{11} = \underline{d}_{21} = \underline{d}_{12} = \underline{d}_{22} = 3$, $\overline{d}_{11} = \overline{d}_{21} = \overline{d}_{12} = \overline{d}_{22} = 4$.

The input 1 (x_1^*) and the input 2 (x_2^*) are selected as 5.25 and 6.5, respectively. The firing strengths are as follows:

$$\begin{aligned} \overline{f}_1 &= 0.6875 * 0.6250 = 0.4297 \\ \underline{f}_1 &= 0.5833 * 0.5000 = 0.2917 \\ \overline{f}_2 &= 0.3125 * 0.3750 = 0.1172 \\ \underline{f}_2 &= 0.0833 * 0.1667 = 0.0139 \end{aligned} \tag{3.14}$$

$$\begin{aligned} u_1 &= 4x_1 + x_2 = 4 * 5.25 + 6.5 = 27.50 \\ u_2 &= 2x_1 + 3x_2 = 2 * 5.25 + 3 * 6.5 = 30.00 \end{aligned} \tag{3.15}$$

$$\begin{aligned} u_l &= \frac{0.4297 * 27.50 + 0.0139 * 30.00}{0.4297 + 0.0139} = 27.5783 \\ u_r &= \frac{0.2917 * 27.50 + 0.1172 * 30.00}{0.2917 + 0.1172} = 28.2166 \end{aligned} \tag{3.16}$$

The final output is calculated as follows:

$$u^* = \frac{u_l + u_r}{2} = 27.8974 \tag{3.17}$$

3.3.3. TSK FLS Model III

The TSK Model III can be described by fuzzy IF-THEN rules. For instance, in a first-order type-2 TSK Model III rule base is as follows [30]:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{j1} \text{ and } x_2 \text{ is } A_{j2} \text{ and } \dots \text{and } x_n \text{ is } A_{jn} \\ &\text{THEN } U_j = \sum_{i=1}^{n} W_{ij} x_i + B_j \end{aligned} \tag{3.18}$$

where x_1, x_2, ... ,x_n are the input variables, U_j's are the output variables, $\mathrm{A_{ji}}$'s are type-1 membership functions for j^{th} rule and the i^{th} input. The parameters in the consequent part of the rules are W_{ij} and B_j $(i = 1, \dots, n, j = 1, \dots, M)$. It is to be noted that both the consequent parameters and the outputs of the rules above are type-1 fuzzy sets. Also, A_{jk}'s are type-1 fuzzy sets $(k = 1, ...n)$.

The final output of the model is as follows:

$$U(U_1, ...U_M) = \int_{u_1} \dots \int_{u_M} \tau_{j=1}^{M} \mu_{U_j}(u_j) \frac{\sum_{j=1}^{M} f_j u_j}{\sum_{j=1}^{M} f_j} \tag{3.19}$$

where M is the number of rules fired, $u_j \in U_j$.

f_j is the firing strength which is defined as:

$$f_j = \mu_{A_{j1}}(x_1) \star \mu_{A_{j2}}(x_2) \star ... \star \mu_{A_{jn}}(x_n) \tag{3.20}$$

4. ANALYSIS OF THE NOISE REDUCTION PROPERTY OF T2FLSs

4.1. Introduction

There are (at least) eight sources of uncertainties in FLSs:

(i) The precision of the measurement devices,
(ii) The noise of the measurement devices,
(iii) The environmental conditions of the measurement devices,
(iv) The unknown nonlinear characteristics of the actuators,
(v) A real-time system cannot be modeled accurately, and there are always some modeling uncertainties,
(vi) The meanings of the words that are used in the antecedents and consequents of rules can be uncertain (words mean different things to different people) [28],
(vii) Consequents may have a histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not all agree [28],
(viii) Uncertainty caused by some unvisited data which the fuzzy system does not have any predefined rules for.

T2FLSs appear to be a more promising method than their type-1 counterparts for handling uncertainties such as noisy data and changing environments [33]. The performance of the type-2 fuzzy sets in the presence of measurement noise has been considered in a number of papers. In [34] the effects of measurement noise in T1FLCs and T2FLCs are simulated to perform a comparative analysis of the responses of the systems in the presence of uncertainty. It is concluded that using a T2FLC in real world applications which exhibit measurement noise can be a good option. In [35] type-2 fuzzy logic theory is applied to predict Mackey-Glass chaotic time-series with uniform noise. The comparison between type-1 and type-2 fuzzy systems shows the superiority of type-2 fuzzy systems in the presence of noise at the inputs. In [36] a type-2 fuzzy system is used with a sliding mode controller to control a nonlinear dynamical plant.

The effect of measurement noise is considered in that paper, and type-2 fuzzy system outperforms its type-1 counterpart. Although all these papers above claim that the performance of the T2FLSs is superior over their type-1 counterparts especially under noisy conditions, the justifications offered for the claim made about the noise reduction property is generally limited to some simulation studies carried out for some specific systems. To our best knowledge, there does not exist any paper in the literature which makes such a general comment about the noise reduction property of T2FLSs. In this dissertation, we propose a novel type-2 membership function that enables us to come up with some metrics. The parameters of this function that represent uncertainty are de-coupled from the parameters that determine the center and the support of the membership function. This allows us to analyze the distortion of the rule base by the uncertainties in the inputs to the rule-base. To be able to make this analysis, a simple T2FLS with the proposed novel membership function is considered in which the effect of input noise in the rule base can be shown numerically in a general way.

4.2. Type-2 Fuzzy Neural System Structure

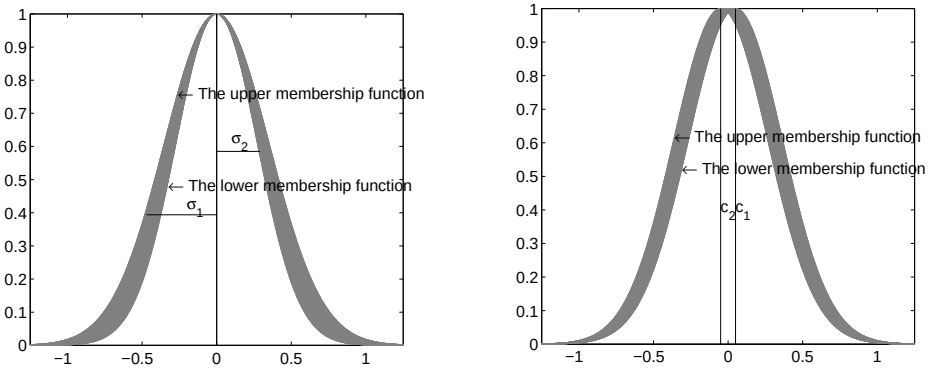
4.2.1. Existing Type-2 Membership Functions in the Literature

There exists a number of type-2 fuzzy membership functions in literature, *i.e.* triangular, gaussian, trapezoidal, sigmoidal, pi-shaped, etc. Gaussian type membership functions are widely used in the literature in which uncertainties can be associated to the mean and the standard deviation.

In Figure 4.1a and 4.1b, Gaussian type-2 fuzzy sets with uncertain standard deviation and uncertain mean are shown. The mathematical expression for the membership function is expressed as:

$$\tilde{\mu}(x) = \exp\left(-\frac{1}{2}\frac{(x-c)^2}{\sigma^2} \right) \tag{4.1}$$

where c and σ are the center and the width of the membership function, x is the input vector.



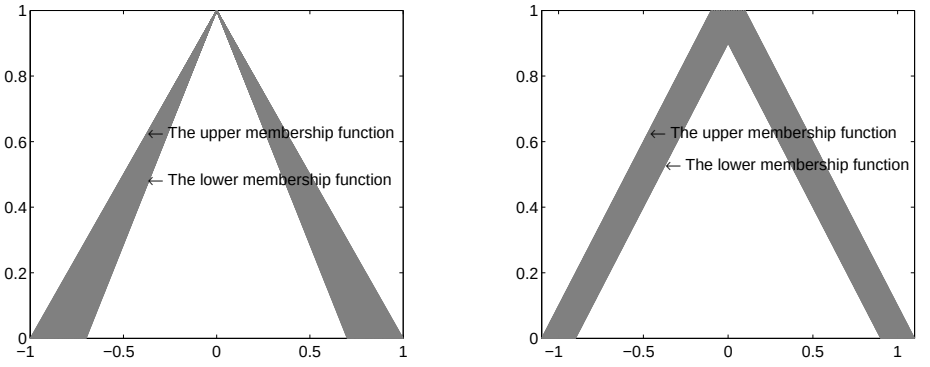
(a) (b)

Figure 4.1. Type-2 fuzzy set with uncertain standard deviation (a) and uncertain mean (b).

In Figure 4.2a and 4.2b, triangular type-2 fuzzy sets with uncertain width and uncertain center are shown. The mathematical expression for the membership function is expressed as:

$$\tilde{\mu}(x) = \begin{cases} 1 - \frac{|x-c|}{d} & \text{if } c-d < x < c+d \\ 0 & else \end{cases} \tag{4.2}$$

where c and d are the center and the width of the membership function, x is the input vector.



(a) (b)

Figure 4.2. Type-2 fuzzy set with uncertain width (a) and center (b).

4.2.2. A Novel Type-2 Membership Function: Ellipsoidal Membership Function

A novel type-2 fuzzy membership function is introduced in this subsection. It has certain values on both ends of the support and the kernel, and some uncertain values on the other values of the support. The lower ($\underline{\mu}$) and the upper ($\bar{\mu}$) membership functions with the parameters c, d, a_1 and a_2 are defined as follows:

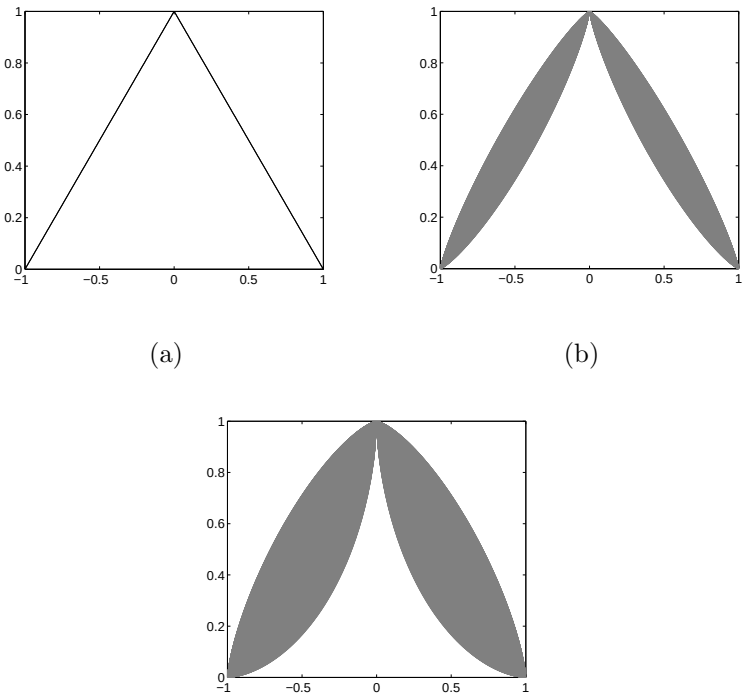
$$\bar{\mu}(x) = \begin{cases} \left(1 - |\frac{x-c}{d}|^{a_1}\right)^{1/a_1} & \text{if } c-d < x < c+d \\ 0 & else \end{cases} \tag{4.3}$$

$$\underline{\mu}(x) = \begin{cases} \left(1 - |\frac{x-c}{d}|^{a_2}\right)^{1/a_2} & \text{if } c-d < x < c+d \\ 0 & else \end{cases} \tag{4.4}$$

where c and d are the center and the width of the membership function, x is the input vector. The parameters a_1 and a_2 determine the width of the uncertainty of the proposed membership function, and these parameters should be selected in the following form:

$$\begin{aligned} a_1 &> 1 \\ 0 < a_2 &< 1 \end{aligned} \tag{4.5}$$

Figures 4.3a, 4.3b and 4.3c show the shapes of the proposed membership function for $a_1 = a_2 = 1$, $a_1 = 1.2, a_2 = 0.8$ and $a_1 = 1.4, a_2 = 0.6$, respectively. As can be seen from Figure 4.3a, the shape of the proposed type-2 membership function is changed to a type-1 triangular membership function when its parameters are selected as $a_1 = a_2 = 1$. These parameters can be selected as some constants or they can be tuned adaptively.



(a) (b)

(c)

Figure 4.3. The shapes of the proposed type-2 membership function with different values for a_1 and a_2.

4.2.3. The Structure of T2FLS

The interval T2FLS considered in this part of the dissertation benefits from type-2 membership functions in the premise part and crisp numbers for the consequent part. Such a structure is called A2-C0 fuzzy system in the literature [32], and the rule base is as follows:

$$\begin{aligned} &\text{IF } x_1 \text{ is } \tilde{A}_{j1} \text{ and } x_2 \text{ is } \tilde{A}_{j2} \text{ and } \ldots \text{and } x_n \text{ is } \tilde{A}_{jn} \\ &\qquad\qquad \text{THEN } u_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \end{aligned} \tag{4.6}$$

where x_1, x_2, ... ,x_n are the input variables, $u_j (j = 1, \ldots, M)$'s are the output variables, $\tilde{\text{A}}_{\text{ji}}$'s are type-2 membership functions for j^{th} rule and the i^{th} input. The parameters in the consequent part of the rules are w_{ij} and b_j $(i = 1, \ldots, n, j = 1, \ldots, M)$.

The final output of the system can be written as [32]:

$$Y_{TSK/A2-C0} = \int_{f^1 \in [\underline{f}^1, \overline{f}^1]} ... \int_{f^M \in [\underline{f}^M, \overline{f}^M]} 1 / \frac{\sum_{j=1}^{M} f^j u_j}{\sum_{j=1}^{M} f^j} \tag{4.7}$$

where $\underline{f}^j$ and $\overline{f}^j$ are given by:

$$\begin{aligned} \underline{f}^j(x) &= \underline{\mu}_{\tilde{F}_1^j}(x_1) * ... * \underline{\mu}_{\tilde{F}_n^j}(x_n) \\ \overline{f}^j(x) &= \overline{\mu}_{\tilde{F}_1^j}(x_1) * ... * \overline{\mu}_{\tilde{F}_n^j}(x_n) \end{aligned} \tag{4.8}$$

in which $*$ represents the t-norm which is the *prod* operator in this study. The computational output of the fuzzy system in closed form is achieved by [32]:

$$Y_{TSK1} = \frac{\sum_{j=1}^{M} \underline{f}^j u_j}{\sum_{j=1}^{M} \underline{f}^j + \sum_{j=1}^{M} \overline{f}^j} + \frac{\sum_{j=1}^{M} \overline{f}^j u_j}{\sum_{j=1}^{M} \underline{f}^j + \sum_{j=1}^{M} \overline{f}^j} \tag{4.9}$$

$$Y_{TSK1} = \frac{\sum_{j=1}^{M} (\underline{f}^j + \overline{f}^j) u_j}{\sum_{j=1}^{M} \underline{f}^j + \sum_{j=1}^{M} \overline{f}^j} \tag{4.10}$$

In this way, the firing of each rule is defined as follows:

$$r_j = \frac{\underline{f}^j + \overline{f}^j}{\sum_{j=1}^{M} \underline{f}^j + \sum_{j=1}^{M} \overline{f}^j} \tag{4.11}$$

Figure 4.4 shows the general structure of the T2FLS used in this part of the dissertation.

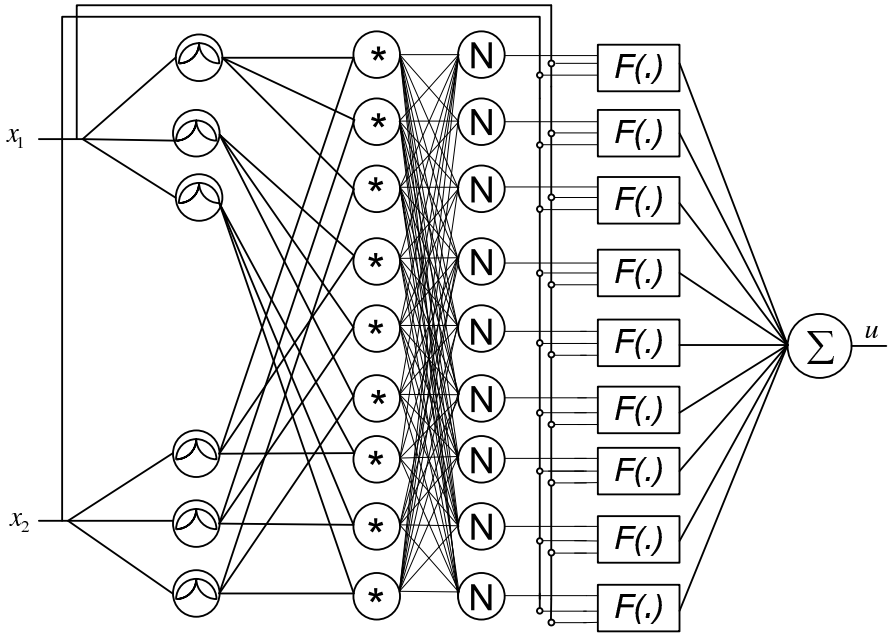


Figure 4.4. The structure of T2FLS with the proposed membership function.

4.2.4. Noise Reduction Property of the Proposed Type-2 Membership Function

In an effort to prove the noise reduction property of T2FLSs, we consider two different FLSs with the proposed membership functions. The first one is a T2FLS with one input with two membership functions, and the other one is with two inputs with two membership functions for each. As the parameters responsible for the width of uncertainty and the parameters responsible for the center and the support of the proposed membership function (c and d respectively) are decoupled from each other in the type-2 membership function, it is possible to analyze how the width of uncertainty of the membership function and the distortion caused by noise (DCN) in the rule base of the fuzzy system are related.

4.2.4.1. Case I. Let us consider a single input T2FLS that uses two membership functions $\tilde{\mu}_1$ and $\tilde{\mu}_2$ such that:

$$\bar{\mu}_2 = 1 - \underline{\mu}_1 \tag{4.12}$$

$$\underline{\mu}_2 = 1 - \bar{\mu}_1 \tag{4.13}$$

The lower ($\underline{\mu}_1$) and the upper ($\bar{\mu}_1$) membership functions with the parameters c_1, d_1, a_1 and a_2 are defined as follows:

$$\bar{\mu}_1(x) = \begin{cases} \left(1 - |\frac{x-c_1}{d_1}|^{a_1}\right)^{1/a_1} & \text{if } c_1 - d_1 < x < c_1 + d_1 \\ 0 & else \end{cases} \tag{4.14}$$

$$\underline{\mu}_1(x) = \begin{cases} \left(1 - |\frac{x-c_1}{d_1}|^{a_2}\right)^{1/a_2} & \text{if } c_1 - d_1 < x < c_1 + d_1 \\ 0 & else \end{cases} \tag{4.15}$$

The fuzzy system has two rules. Using Equation 4.11, the firing strength of the first rule is calculated as:

$$r_1(x) = \frac{\bar{\mu}_1(x) + \underline{\mu}_1(x)}{2} \tag{4.16}$$

The firing strength will be distorted by the noise in the data as:

$$r_1(x+n) = \frac{\bar{\mu}_1(x+n) + \underline{\mu}_1(x+n)}{2} \tag{4.17}$$

where n indicates the noise added on the signal.

The total DCN over the support set can be found by the following integral:

$$\text{DCN} = \int_{n=-n_1}^{n=n_1} \int_{x=-d_1+c_1}^{x=d_1+c_1} \Big[r_1(x) - r_1(x+n) \Big]^2 dx dn \tag{4.18}$$

To simplify the limits in Equation 4.18, the following definitions are done:

$$t = \frac{x - c_1}{d_1},\ n' = \frac{n}{d_1} \tag{4.19}$$

Using the definitions in Equation 4.19, the following equation is derived:

$$\mathrm{DCN} = d_1^2 \int_{n'=-n_1/d_1}^{n'=n_1/d_1} \int_{t=-1}^{t=1} \Big[r_1(x) - r_1(x+n) \Big]^2 dt dn' \tag{4.20}$$

In above, the parameter n_1 is the magnitude of the amplitude of the noise added onto the input of the fuzzy system.

The integral in Equation 4.20 cannot be calculated explicitly. Therefore a numerical solution of this integral is obtained for each pair of a_1 and a_2, and the distortion is drawn with respect to a_1 and a_2 parameters. Figure 4.5 shows the numerical solution of the integral above for the noise level signal to noise ratio (SNR)= $0dB$. It is to be noted that to achieve SNR=0dB, n_1 is selected as being equal to d_1. As can be seen from that figure, the parameter a_2 is more critical for noise reduction as compared to the parameter a_1. We can also see that the distortion caused by the noise in the case of $a_1 = a_2 = 1$ which corresponds to type-1 membership function is higher than the other values of a_1 and a_2 which correspond to the case of type-2 membership functions. The figure indicates that by a proper selection of the parameters a_1 and a_2, it is possible to achieve better performance in the presence of noise. Although it has already been stated that the parameter a_1 should be selected bigger than 1 and the parameter a_2 between 0 and 1, Figure 4.5 indicates what values of a_1 and a_2 are better to be used. Although the figure shows that it is better to select the parameter a_2 bigger than 0.3 and any value for a_1, a very small value for a_2 results in such a lower membership function that the membership grade is close to zero for a large portion of its support. Similarly a very large value for a_1 results in such an upper membership function that the membership grade is close to one for a large portion of its support. Such membership functions are not very desirable. A proper selection for a_1 and a_2 would therefore be $1 < a_1 < 2$ and $1 > a_2 > 0.5$

Figure 4.6 shows the numerical solution of the integral for a low level of noise (SNR=100dB). This figure indicates that the noise reduction property of T1FLSs is comparable with T2FLSs in the presence of very low levels of noise.

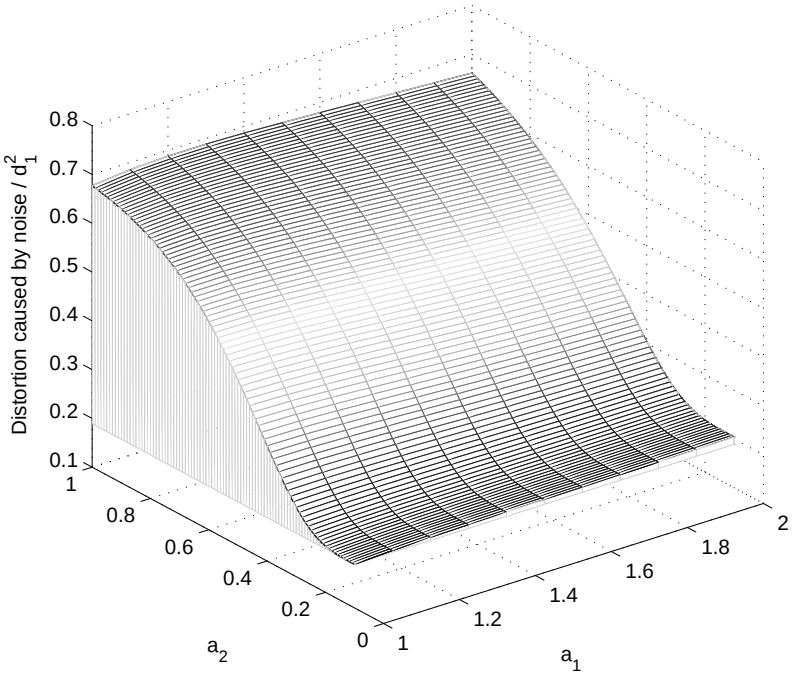


Figure 4.5. The three-dimensional figure of DCN in Equation 4.20 w.r.t. a_1 and a_2 for high levels of noise (SNR=0dB).

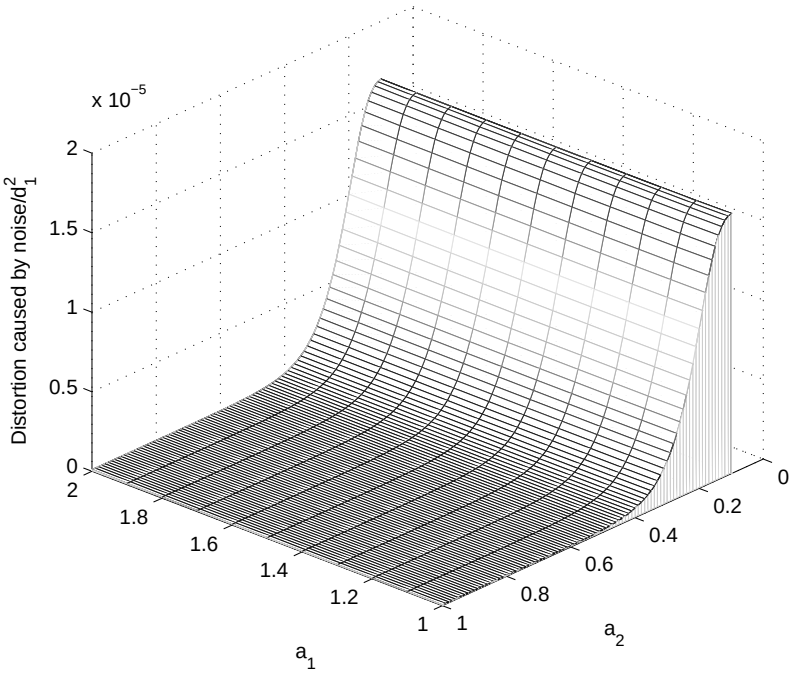


Figure 4.6. The three-dimensional figure of DCN in Equation 4.20 w.r.t. a_1 and a_2 for low levels of noise (SNR=100dB).

4.2.4.2. Case II. Consider a T2FLS with two inputs and two type-2 membership functions for each input. The membership functions for the first input are selected as $\tilde{\mu}_{11}$ and $\tilde{\mu}_{12}$, and the membership function selected for the second input are as: $\tilde{\mu}_{21}$ and $\tilde{\mu}_{22}$. The type-2 fuzzy membership function $\tilde{\mu}_{11}$ is defined as:

$$\bar{\mu}_{11}(x) = \begin{cases} \left(1 - |\frac{x-c_{11}}{d_{11}}|^{a_{1_{11}}}\right)^{1/a_{1_{11}}} & \text{if } |x - c_{11}| < d_{11} \\ 0 & else \end{cases} \tag{4.21}$$

$$\underline{\mu}_{11}(x) = \begin{cases} \left(1 - |\frac{x-c_{11}}{d_{11}}|^{a_{2_{11}}}\right)^{1/a_{2_{11}}} & \text{if } |x - c_{11}| < d_{11} \\ 0 & else \end{cases} \tag{4.22}$$

The type-2 fuzzy membership function $\tilde{\mu}_{21}$ is defined as:

$$\bar{\mu}_{21}(x) = \begin{cases} \left(1 - |\frac{x-c_{21}}{d_{21}}|^{a_{1_{21}}}\right)^{1/a_{1_{21}}} & \text{if } |x - c_{21}| < d_{21} \\ 0 & else \end{cases} \tag{4.23}$$

$$\underline{\mu}_{21}(x) = \begin{cases} \left(1 - |\frac{x-c_{21}}{d_{21}}|^{a_{2_{21}}}\right)^{1/a_{2_{21}}} & \text{if } |x - c_{21}| < d_{21} \\ 0 & else \end{cases} \tag{4.24}$$

In order to see the effect of $a_{1_{11}}$ and $a_{2_{11}}$ on the DCN, $a_{1_{21}}$ and $a_{2_{21}}$ are set to 1. The other membership functions are considered as:

$$\begin{aligned} \bar{\mu}_{12} &= 1 - \underline{\mu}_{11} \\ \underline{\mu}_{12} &= 1 - \bar{\mu}_{11} \\ \bar{\mu}_{22} &= 1 - \underline{\mu}_{21} \\ \underline{\mu}_{22} &= 1 - \bar{\mu}_{21} \end{aligned} \tag{4.25}$$

This fuzzy system has four rules. The firing strength for the first rule is written as:

$$r_1(x_1, x_2) = \frac{1}{2}\Big(\bar{\mu}_{11}(x_1)\bar{\mu}_{21}(x_2) + \underline{\mu}_{11}(x_1)\underline{\mu}_{21}(x_2)\Big) \tag{4.26}$$

The firing strength will be distorted by the noise in the data as:

$$r_1(x_1 + n, x_2) = \frac{1}{2}\Big(\bar{\mu}_{11}(x_1 + n)\bar{\mu}_{21}(x_2) + \underline{\mu}_{11}(x_1 + n)\underline{\mu}_{21}(x_2)\Big) \tag{4.27}$$

The total DCN over the support set can be calculated by the following integral:

$$\begin{aligned}\text{DCN} = \int_{n=-n_1}^{n=n_1}\int_{x_2=-d_{21}+c_{21}}^{x_2=d_{21}+c_{21}}\int_{x_1=-d_{11}+c_{11}}^{x_1=d_{11}+c_{11}}\Big[r_1(x_1, x_2) \\ -r_1(x_1 + n, x_2)\Big]^2 dx_1 dx_2 dn\end{aligned} \tag{4.28}$$

To simplify the limits in Equation 4.28, the following definitions are done:

$$t_1 = \frac{x_1 - c_{11}}{d_{11}},\ t_2 = \frac{x_2 - c_{21}}{d_{21}},\ n' = \frac{n}{d_{11}} \tag{4.29}$$

Using the definitions in Equation 4.29, the following equation is achieved:

$$\begin{aligned}\text{DCN} = d_{11}^2 d_{21}\int_{n'=-n_1/d_{11}}^{n'=n_1/d_{11}}\int_{t_2=-1}^{t_2=1}\int_{t_1=-1}^{t_1=1}\Big[r_1(x_1, x_2) \\ -r_1(x_1 + n, x_2)\Big]^2 dt_1 dt_2 dn'\end{aligned} \tag{4.30}$$

Similar to Case I, the integral in Equation 4.30 cannot be calculated explicitly. Therefore a numerical solution of this integral is obtained for each pair of a_1 and a_2,

and the distortion is drawn with respect to a_1 and a_2 parameters in Figure 4.7. This figure is obtained with n_1 is equal to d_1 which corresponds to SNR=0dB. Figure 4.8, which is the contour diagram of Figure 4.7, shows that there is a proper selection area for a_1 and a_2. Based on similar arguments, a proper selection for a_2 and a_1 would therefore be $1 < a_1 < 2$ and $1 > a_2 > 0.5$.

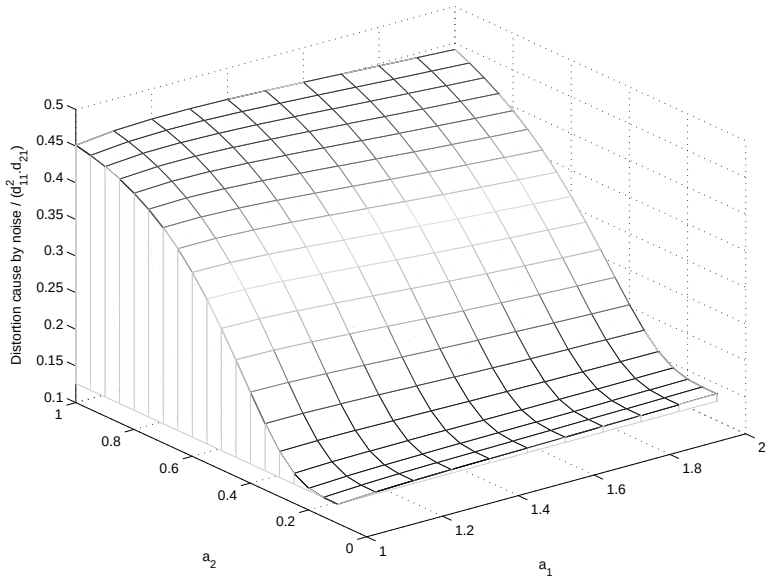


Figure 4.7. The three-dimensional figure of DCN in Equation 4.30 w.r.t. a_1 and a_2 for high levels of noise (SNR=0dB).

Similar to Figure 4.6, the numerical solution of the integral for a low level of noise (SNR=100dB) is shown in Figure 4.9. This figure indicates that the noise reduction property of T1FLSs is comparable with T2FLSs in the presence of very low levels of noise. Therefore, in the presence of low levels of noise, T1FLS is more preferable as compared to T2FLS because of its computational simplicity.

4.3. Parameter Update Rules

The design of T2FLS (Figure 4.4) includes the determination of the unknown parameters that are the parameters of the antecedent and the consequent parts of the fuzzy *If-Then* rules. The gradient algorithm is applied to design the antecedent

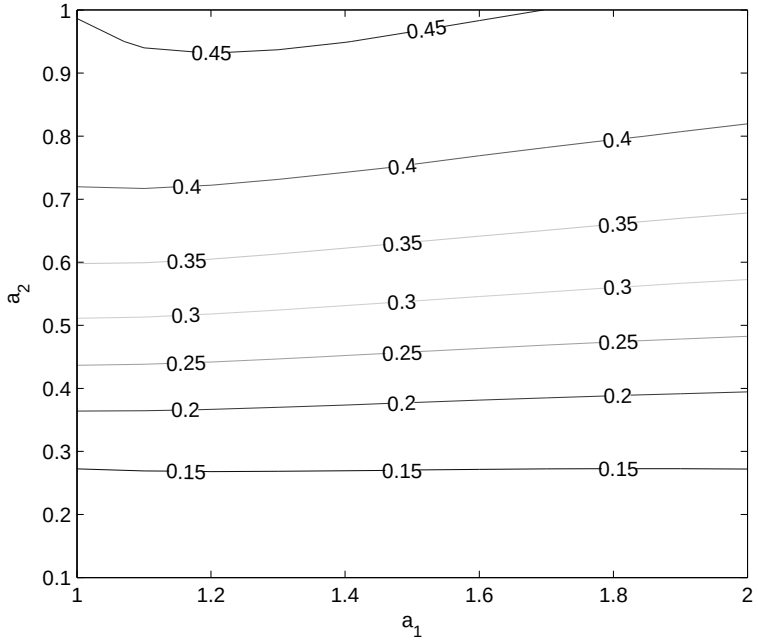


Figure 4.8. The contour figure of DCN in Equation 4.30 w.r.t. a_1 and a_2 (SNR = 0dB).

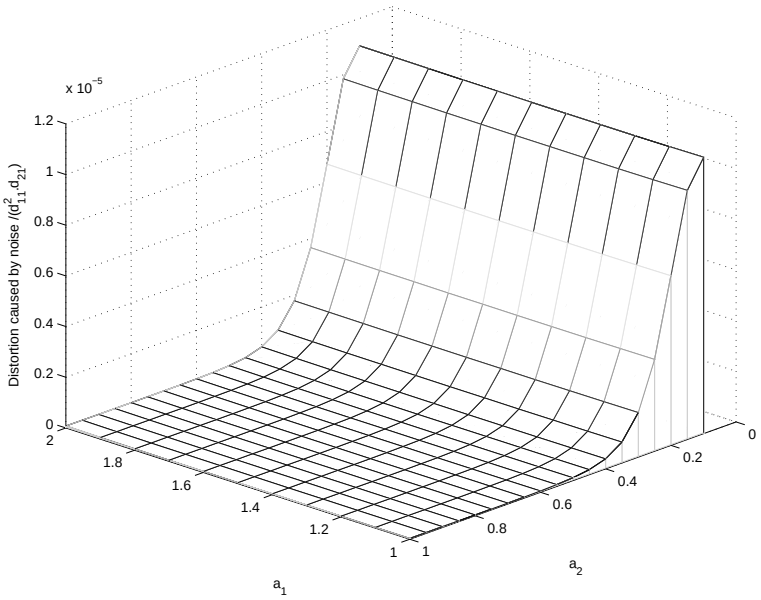


Figure 4.9. The three-dimensional figure of DCN in Equation 4.30 w.r.t. a_1 and a_2 for low levels of noise (SNR=100dB).

(premise) and the consequent parts of the fuzzy rules.

At the first step, the cost function is defined as:

$$E = \frac{1}{2}(u^d - u)^2 \tag{4.31}$$

where u^d and u are the desired and the current output values of the network, respectively.

The parameters w_{ij}, b_j and a_{1ij}, a_{2ij}, c_{ij} and d_{ij} are adjusted using the GD method as follows:

$$w_{ij}(t+1) = w_{ij}(t) - \gamma \frac{\partial E}{\partial w_{ij}} \tag{4.32}$$

$$b_j(t+1) = b_j(t) - \gamma \frac{\partial E}{\partial b_j} \tag{4.33}$$

$$c_{ij}(t+1) = c_{ij}(t) - \gamma \frac{\partial E}{\partial c_{ij}} \tag{4.34}$$

$$a_{1ij}(t+1) = a_{1ij}(t) - \gamma \frac{\partial E}{\partial a_{1ij}} \tag{4.35}$$

$$a_{2ij}(t+1) = a_{2ij}(t) - \gamma \frac{\partial E}{\partial a_{2ij}} \tag{4.36}$$

$$d_{ij}(t+1) = d_{ij}(t) - \gamma \frac{\partial E}{\partial d_{ij}} \tag{4.37}$$

where γ is the learning rate, $a_{1_{ij}}$, $a_{2_{ij}}$, c_{ij} and d_{ij} are the i^{th} parameter of j^{th} rule. The derivatives in Equation 4.32 and Equation 4.33 are determined by the following formulas:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \frac{\partial E}{\partial u}\frac{\partial u}{\partial u_j}\frac{\partial u_j}{\partial w_{ij}} \\ \frac{\partial E}{\partial b_j} &= \frac{\partial E}{\partial u}\frac{\partial u}{\partial u_j}\frac{\partial u_j}{\partial b_j} \end{aligned} \tag{4.38}$$

where u_j is as defined in Equation 4.6

The derivatives in Equation 4.34 through Equation 4.37 are determined by the following formulas:

$$\frac{\partial E}{\partial c_{ij}} = \sum_j \frac{\partial E}{\partial u}\left[\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial c_{ij}} + \frac{\partial u}{\partial \overline{f}_j}\frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}}\frac{\partial \overline{\mu_{ij}}}{\partial c_{ij}}\right] \tag{4.39}$$

$$\frac{\partial E}{\partial d_{ij}} = \sum_j \frac{\partial E}{\partial u}\left[\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial d_{ij}} + \frac{\partial u}{\partial \overline{f}_j}\frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}}\frac{\partial \overline{\mu_{ij}}}{\partial d_{ij}}\right] \tag{4.40}$$

$$\begin{aligned} \frac{\partial E}{\partial a_{1ij}} &= \sum_j \frac{\partial \mathrm{E}}{\partial u}\frac{\partial u}{\partial \overline{f}_j}\frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}}\frac{\partial \overline{\mu}_{ij}}{\partial a_{1ij}} \\ \frac{\partial E}{\partial a_{2ij}} &= \sum_j \frac{\partial \mathrm{E}}{\partial u}\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial a_{2ij}} \end{aligned} \tag{4.41}$$

The parameters of the T2FLS can thus be updated using Equation 4.32 through Equation 4.37 together with Equation 4.39 through Equation 4.41.

The derivatives in Equation 4.38 are calculated as:

$$\frac{\partial E}{\partial u} = u(t) - u^d(t), \quad \frac{\partial u}{\partial u_j} = 1, \quad \frac{\partial u_j}{\partial w_{ij}} = r_j x_i \text{ and } \frac{\partial u}{\partial b_j} = r_j \tag{4.42}$$

$$\begin{aligned}\frac{\partial \bar{\mu}_{ij}}{\partial a_{1ij}} &= -\frac{1}{\mathrm{a_{1ij}}^2}\ln\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{1ij}}}\right)\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{1ij}}}\right)^{\frac{1}{\mathrm{a_{1ij}}}} \\ &- \frac{1}{\mathrm{a_{1ij}}}\ln\left|\frac{x_i-c_{ij}}{d_{ij}}\right|\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{1ij}}}\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{1ij}}}\right)^{\frac{1}{\mathrm{a_{1ij}}}-1} \\ & c_{ij}-d_{ij}<x_i<c_{ij}+d_{ij} \end{aligned} \tag{4.43}$$

$$\begin{aligned}\frac{\partial \bar{\mu}_{ij}}{\partial c_{ij}} &= \frac{1}{|d_{ij}|}\mathrm{sign}(x_i-c_{ij})\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{1ij}}-1}\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a1ij}}\right)^{\frac{1}{\mathrm{a_{1ij}}}-1} \\ & c_{ij}-d_{ij}<x_i<c_{ij}+d_{ij} \end{aligned} \tag{4.44}$$

$$\begin{aligned}\frac{\partial \bar{\mu}_{ij}}{\partial d_{ij}} &= \frac{1}{|d_{ij}|^2}\mathrm{sign}(d_{ij})\ |x_i-c_{ij}|\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{1ij}}-1}\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{1ij}}}\right)^{\frac{1}{\mathrm{a_{1ij}}}-1} \\ & c_{ij}-d_{ij}<x_i<c_{ij}+d_{ij} \end{aligned} \tag{4.45}$$

$$\begin{aligned}\frac{\partial \underline{\mu}_{ij}}{\partial a_{2ij}} &= -\frac{1}{\mathrm{a_{2ij}}^2}\ln\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{2ij}}}\right)\ \left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{2ij}}}\right)^{\frac{1}{\mathrm{a_{2ij}}}} \\ &- \frac{1}{\mathrm{a_{2ij}}}\ln\left|\frac{x_i-c_{ij}}{d_{ij}}\right|\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{2ij}}}\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{2ij}}}\right)^{\frac{1}{\mathrm{a_{2ij}}}-1} \\ & c_{ij}-d_{ij}<x_i<c_{ij}+d_{ij} \end{aligned} \tag{4.46}$$

$$\begin{aligned}\frac{\partial \underline{\mu}_{ij}}{\partial c_{ij}} &= \frac{1}{|d_{ij}|}\mathrm{sign}(x_i-c_{ij})\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{2ij}}-1}\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a2ij}}\right)^{\frac{1}{\mathrm{a_{2ij}}}-1} \\ & c_{ij}-d_{ij}<x_i<c_{ij}+d_{ij} \end{aligned} \tag{4.47}$$

$$\begin{aligned}\frac{\partial \underline{\mu}_{ij}}{\partial d_{ij}} &= \frac{1}{|d_{ij}|^2}\mathrm{sign}(d_{ij})\ |x_i-c_{ij}|\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{2ij}}-1}\left(1-\left|\frac{x_i-c_{ij}}{d_{ij}}\right|^{\mathrm{a_{2ij}}}\right)^{\frac{1}{\mathrm{a_{2ij}}}-1} \\ & c_{ij}-d_{ij}<x_i<c_{ij}+d_{ij} \end{aligned} \tag{4.48}$$

4.4. Simulation Studies

4.4.1. Prediction of Chaotic Mackey-Glass Time Series

The proposed T2FLS is used to predict the noisy chaotic Mackey-Glass time series. This chaotic system is a well-known benchmark problem in literature, and it can be described by the following dynamic equation [37]:

$$\dot{x}(t) = 0.2\frac{x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{4.49}$$

The numerical values selected for the chaotic system above are $\tau = 17$, $x(0) = 1.2$ in this study. The following approximation is used:

$$\dot{x}(t) = \frac{x(k+1) - x(k)}{T_s} \tag{4.50}$$

where $T_s = 1$. The number of the training and the test data is equal to 500.

The predictor goal is to predict $x(t+1)$ using the inputs $x(t-3), x(t-2), x(t-1)$ and $x(t)$. For each input two membership functions are used. In this way the number of the rules of the system are equal to 16. In order to study the effect of noise in the proposed system, a number of experiments are performed in which the data are corrupted with different levels noise of SNR. The well-known formula below for SNR is used as:

$$SNR = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_n^2}\right) \tag{4.51}$$

where σ_s^2 is the variance of the signal and σ_n^2 is the variance of the noise.

Figure 4.10 shows the actual and the noisy Mackey-Glass chaotic time series data with SNR equal to 0dB. Six experiments are done using different levels of noise (from 0dB to 10dB), but only the one with SNR=0dB is shown in Figure 4.10.

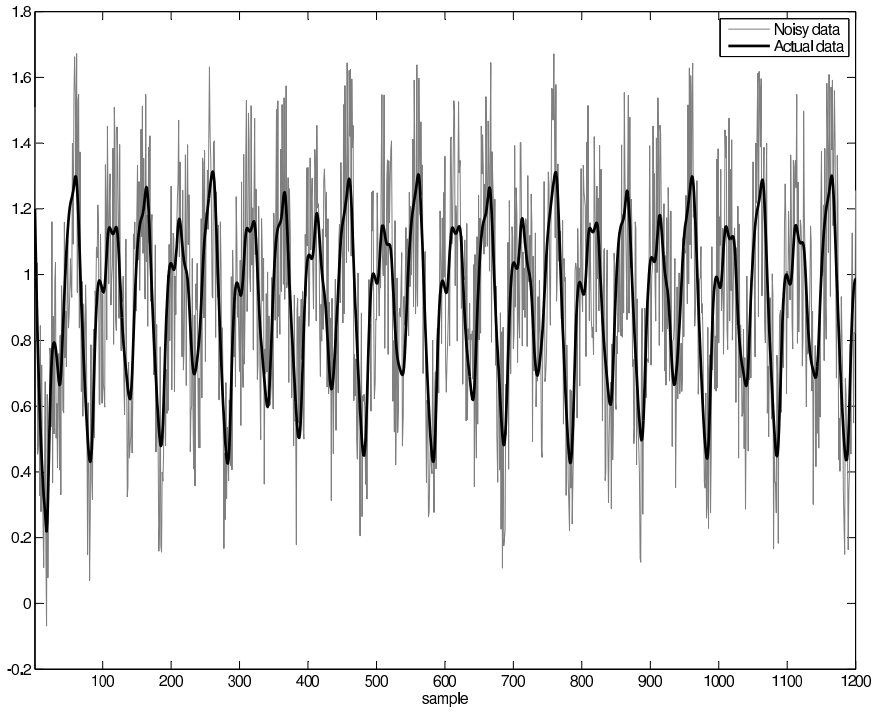


Figure 4.10. The actual Mackey-Glass time series and the noisy data (SNR = 0dB).

In order to tune the parameters (c, d, a_1 and a_2) of type-2 membership function proposed in this part of the dissertation, the gradient method is used. Figure 4.11 shows the result of the tuning operation on these parameters. As can be seen, the lower membership function varies more than the upper one. It is to be noted that Figure 4.5 also indicates that the lower membership function is more critical in the noise reduction property of the T2FLSs.

Figure 4.12 shows the mean value of $\sum_{i=1}^{8} |a_{1i} - a_{2i}|$ over ten times of simulations for Mackey-Glass system. As can be seen, the width of the proposed membership function decreases as the power of the input noise decreases. Again, this fact is also expected from the Figure 4.5. This means that the more noise injected into the T2FLS, the fatter the proposed type-2 membership function will become. This property of the proposed type-2 membership can be regarded as a critical advantage for real-time systems.

Figure 4.13 indicates that the prediction accuracy difference between the type-1 and the type-2 fuzzy predictors with the proposed membership function. It can be

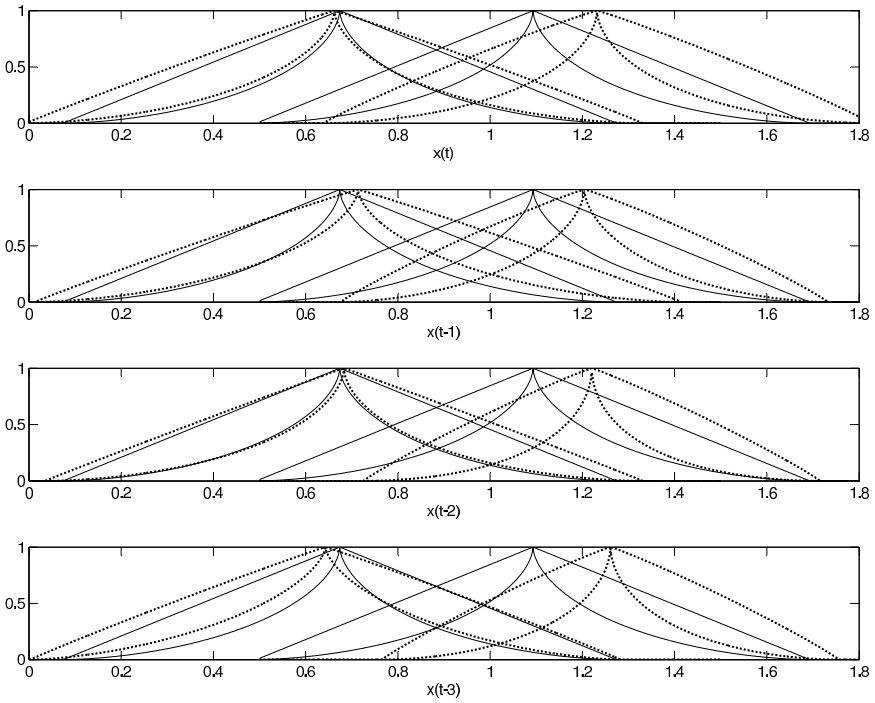


Figure 4.11. The membership functions of the T2FLS before (dotted line) and after (solid line) the training (SNR=0dB).

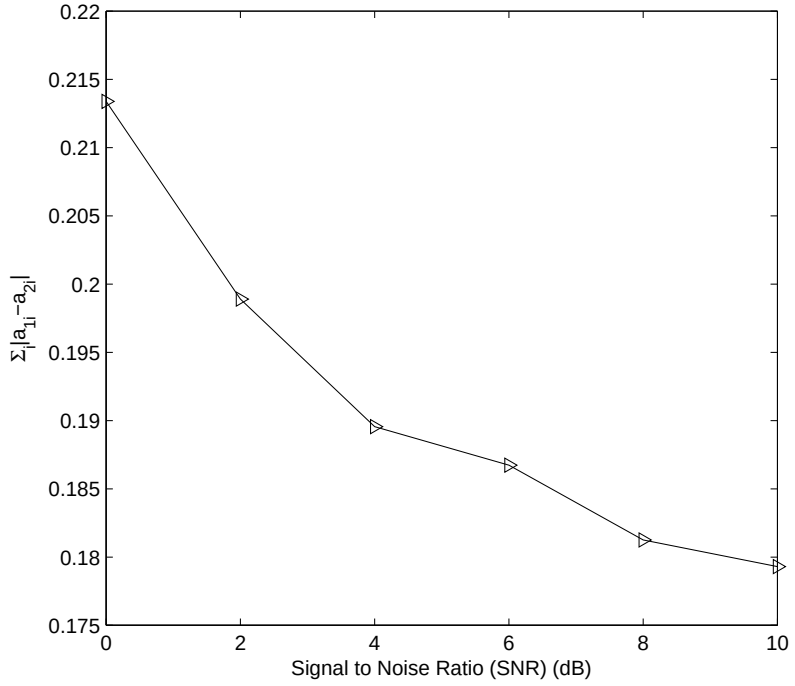


Figure 4.12. The width of the membership function versus the power of input noise.

seen that the performance accuracy of T2FLS is higher than its type-1 counterpart in the presence of the higher noise. A similar conclusion has previously be reached in [38]. Consequently, it can be stated that T2FLSs should be preferred when there exists a large amount of uncertainties in the system considered. In other words, if the uncertainties in the system are relatively small, T1FLS might be a better choice because of having less number of parameters to be tuned.

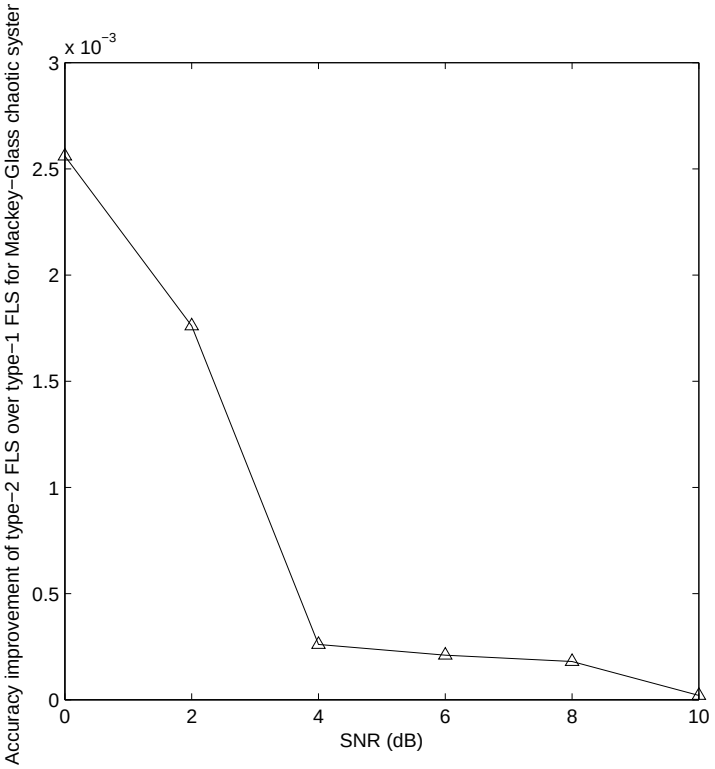


Figure 4.13. The improvement of identification accuracy of T2FLS over T1FLS.

The output of the T2FLS with the proposed novel membership function and the actual data can be seen in Figure 4.14 with SNR=0dB. This figure shows that the prediction accuracy of the proposed T2FLS is quite satisfactory considering amount of noise in the data used for training. The accuracy of the results shown in this part of the dissertation is comparable with the results reported in [35] in which the same data set is used, although it is used interval values in the consequent part in that paper. For example, the root-mean-squared-error (RMSE) quoted in [35] for $SNR = 0dB$ is 0.1429.

Table 4.1 shows the mean values of the RMSE for triangular type-1, Gaussian

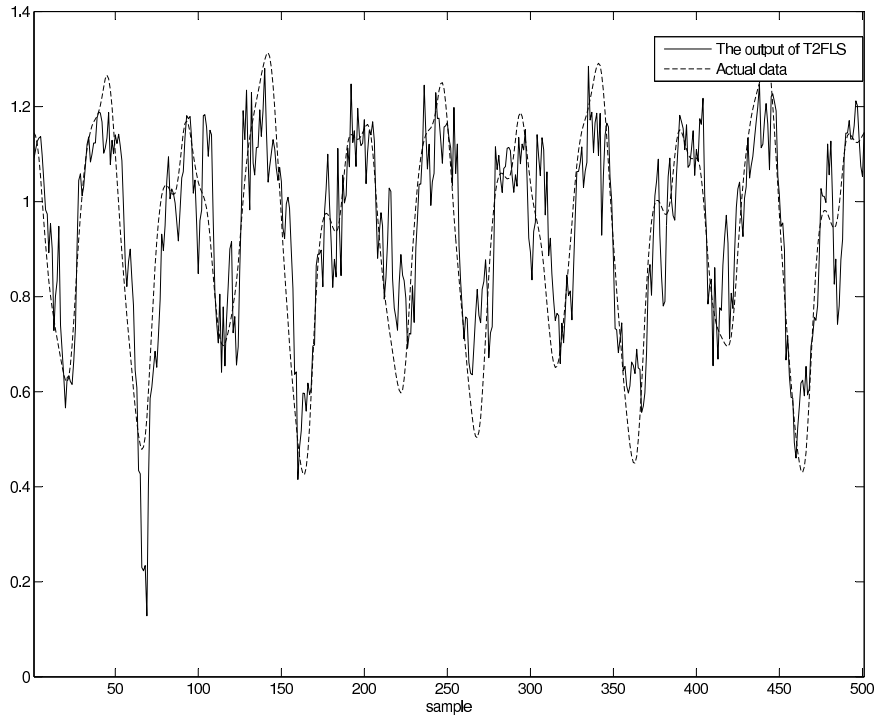


Figure 4.14. The response of the T2FLS with SNR=0dB and the actual noise-free Mackey-Glass data.

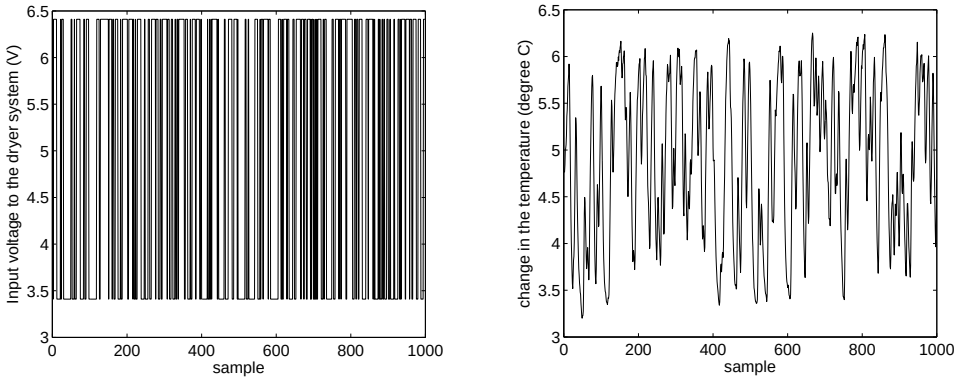
type-2 with uncertain σ, Gaussian type-2 with uncertain center, triangular type-2 with uncertain parameters, proposed ellipsoidal type-2 with constant a_1 and a_2 and proposed ellipsoidal type-2 with adaptive a_1 and a_2 over ten times of simulations, the lowest figures being indicated in bold characters. As can be seen from the table, the prediction accuracy of the T2FLS with the proposed novel membership function is better than others when the noise level is high. In order to be able to make a fair comparison, we have determined the initial values of the type-1 membership functions similar to type-2 membership functions. The parameters for the type-1 fuzzy membership function are as follows: The mean value of the first membership function of all inputs are selected as 0.7 and the second membership function as 1.3. In addition, the width of all membership functions are selected as 0.6. Moreover, a random value in the interval $[-0.2; 0.2]$ is added to each value in every simulation. It is to be noted that even when a_1 and a_2 are kept constant (at 1.2 and 0.8 as suggested by Figure 4.7) the performance of T2FLS is better although the number of parameters to be tuned is the same as T1FLS. If the parameters are tuned during the simulation, the performance is better than the constant case.

Table 4.1. The prediction accuracies of the T2FLS with the proposed type-2 membership function and its type-1 counterpart.

SNR	Triangular Type-1	Gaussian Type-2 with uncertain σ	Gaussian Type-2 with uncertain center	Triangular Type-2 with uncertain parameters	Ellipsoidal Type-2 with constant a_1 and a_2	Ellipsoidal Type-2 with adaptive a_1 and a_2
0dB	0.1257	0.1235	0.1233	0.1250	0.1250	**0.1231**
2dB	0.1178	0.1167	0.1163	0.1167	0.1172	**0.1161**
4dB	0.1097	0.1098	0.1099	**0.1093**	0.1104	0.1098
6dB	0.1032	0.1033	0.1036	**0.1027**	0.1036	0.1032
8dB	0.0971	0.0971	0.0975	**0.0967**	0.0968	**0.0967**
10dB	0.0917	0.0912	0.0916	**0.0907**	0.0909	0.0908

4.4.2. Identification of a Laboratory Setup Acting Like a Hair Dryer Data

The T2FLS with the proposed novel membership function is used to identify the data set of a laboratory setup acting like a hair dryer. In the system, air is fanned through a tube and heated at the inlet. The air temperature is measured by a thermocouple at the output. The input is the voltage over the heating device and the output is the air temperature. The input voltage of the heating device is discrete and in the form of pulse-width-modulation (PWM). Figure 4.15 shows the input-output data set of the real-time setup available on line [39]. Since the PWM signal switches between the values of 3.4 and 6.4 V, the membership functions for the input should be chosen with these center values. In the identification process, $u(t), y(t-1)$ and $y(t-2)$ are chosen as the inputs of the T2FLS proposed, and $y(t)$ is the target value. Two membership functions are considered for $u(t)$ and three membership functions are used for $y(t-1)$ and $y(t-2)$. Consequently, the resulting fuzzy system has 18 rules.



(a) (b)

Figure 4.15. The input (a) - output (b) data set of a real time hair dryer laboratory setup.

Similar to the previous experiment done in Case 1, six different experiments are done using different levels of noise (from 0dB to 10dB), but only the one with SNR=10dB is shown in Figure 4.16. As can be seen from Figure 4.16, the identification accuracy of the T2FLS with the proposed membership function is quite good.

Figure 4.17 indicates that the prediction accuracy difference between the type-1 and the type-2 fuzzy identifiers with the proposed ellipsoidal type-2 fuzzy membership function used in this part of the dissertation decreases as the noise power decreases. The performance accuracy of T2FLS is higher than its type-1 counterpart in the presence of the higher noise. The observation suggests that if the uncertainties in the system are relatively small, T1FLS may be a better choice because of having less number of parameters to be tuned.

Similar to the Figure 4.12, Figure 4.18 shows the mean value of $\sum_{i=1}^{8} |a_{1i} - a_{2i}|$ over ten time of simulations for hair dryer system. As can be seen from Figure 4.18, the width of the proposed membership function decreases as the power of the input noise decreases.

Similar to Table 4.1, Table 4.2 shows the mean values of the RMSE for FLSs given in Table 4.1 over ten times of simulations. As can be seen from this table, the prediction accuracy of the T2FLS with the proposed novel membership function

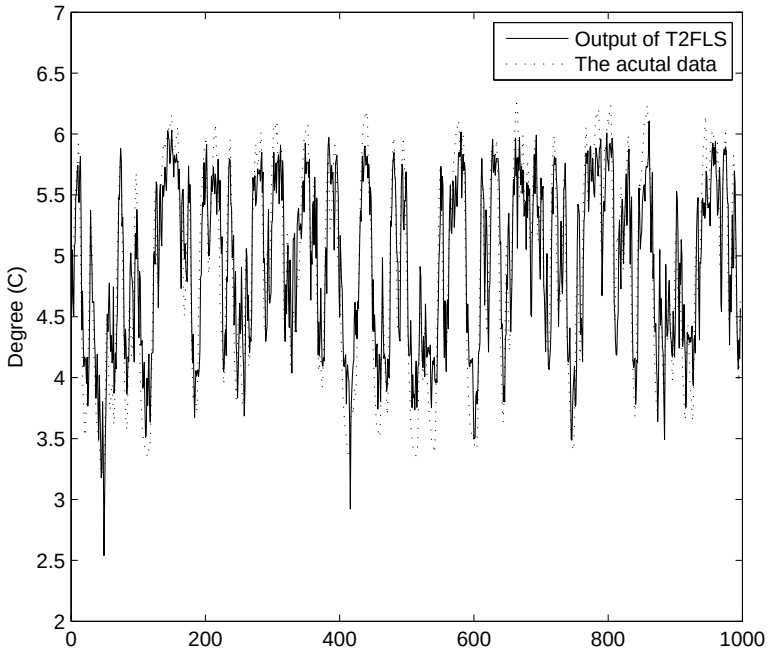


Figure 4.16. The response of the T2FLS with SNR=10dB and the actual noise-free data set.

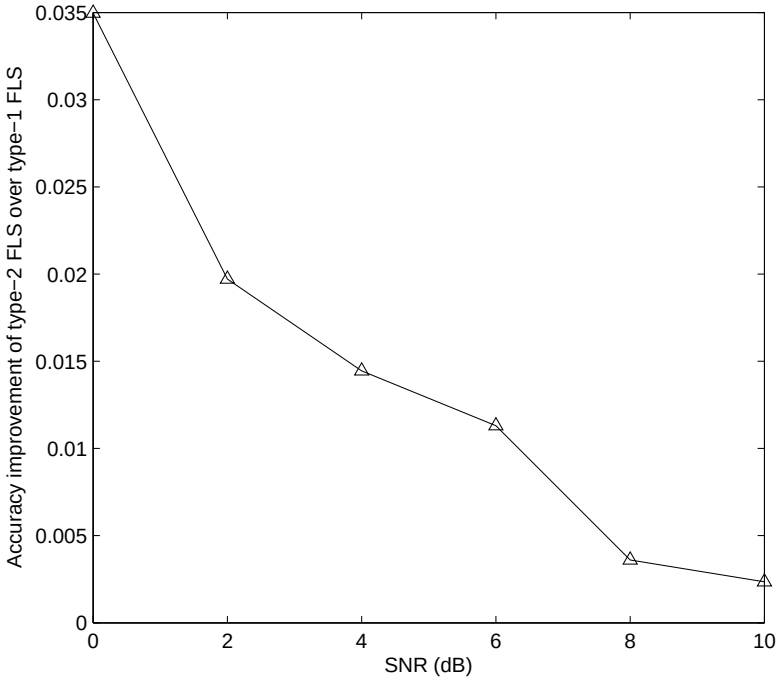


Figure 4.17. The improvement of identification accuracy of T2FLS over T1FLS.

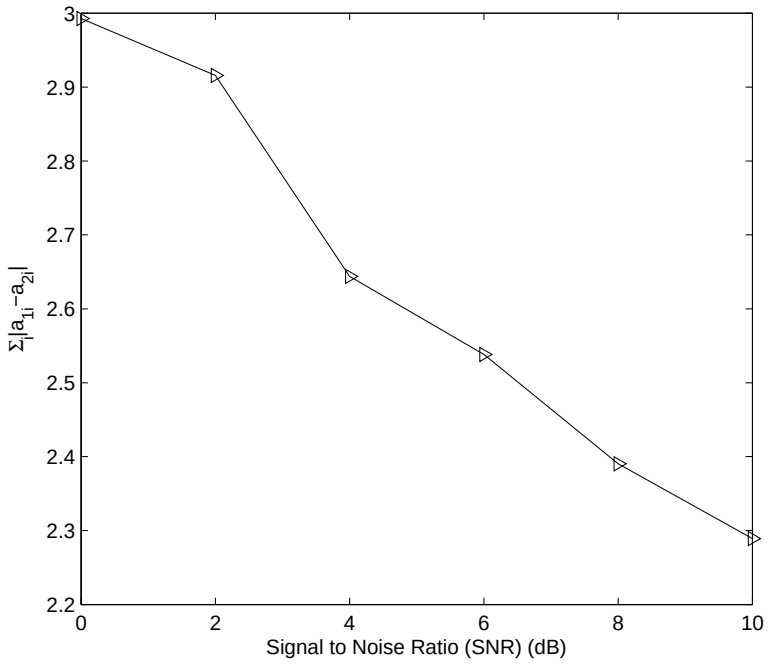


Figure 4.18. The width of the membership function versus the power of input noise for hair dryer data set.

is again comparable with the previously known membership functions. To have a better comparison, the parameters of the novel membership function a_1 and a_2 are kept constant in one case (selected as 1.2 and 0.8, respectively as suggested by Figure 4.9, and tuned in the other case. It is to be noted that with constant a_1 and a_2, the number of the parameters of type-2 and type-1 membership functions are the same. Even in this case, there is a performance improvement in T2FLS in the presence of higher levels of input noise. If the parameters are tuned during the simulation, the performance is better than the constant case.

4.4.3. Control of a Non-BIBO Nonlinear Plant

The T2FLS with the proposed novel membership function is used to control a non-bounded-input-bounded-output (non-BIBO) nonlinear plant. The dynamical

Table 4.2. The identification accuracies of the T2FLS with the proposed type-2 membership function and its type-1 counterpart.

SNR	Triangular Type-1	Gaussian Type-2 with uncertain σ	Gaussian Type-2 with uncertain center	Triangular Type-2 with uncertain parameters	Ellipsoidal Type-2 with constant a_1 and a_2	Ellipsoidal Type-2 with adaptive a_1 and a_2
0dB	0.5719	0.5379	0.5506	0.5372	0.5521	**0.5369**
2dB	0.5272	**0.5009**	0.5165	0.5082	0.5163	0.5075
4dB	0.4897	**0.4693**	0.4860	0.4827	0.4805	0.4753
6dB	0.4570	0.4461	0.4567	0.4527	0.4577	**0.4457**
8dB	0.4207	**0.4164**	0.4245	0.4258	0.4202	0.4171
10dB	0.3964	0.3969	0.4092	0.4006	0.3954	**0.3941**

plant model is as follows [40]:

$$\begin{aligned} y(k+1) &= 0.2y^2(k) + 0.2y(k-1) \\ &\quad +0.4sin\left[0.5(y(k)+y(k-1))\right] \\ &\quad .cos\left[0.5(y(k)+y(k-1))\right] + 1.2u(k) \end{aligned} \tag{4.52}$$

The plant is a non-BIBO nonlinear plant. It means that for a given uniformly bounded input signal, the plant output may diverge. The plant output diverges when the step input $u(k) = 0.83, \forall k \geq 0$, is applied to the plant [41].

In the first case (in Figure 4.19), the reference signal is chosen as:

$$r(k) = \begin{cases} 0.25 & \text{if } k \leq 750 \\ 0.50 & \text{if } 750 < k \leq 1500 \\ 0.25 & \text{if } 1500 < k \leq 2250 \\ 0.50 & \text{if } 2250 < k \leq 3000 \\ 0.25 & \text{if } 3000 < k \leq 3750 \\ 0.50 & \text{if } 3750 < k \leq 4500 \end{cases} \tag{4.53}$$

In the second case (in Figure 4.20), the reference signal is chosen as:

$$r(k) = 0.4 + 0.2cos\big(2k\pi/2000\big) \tag{4.54}$$

An output noise in the interval of $[-0.1; 0.1]$ is added to the output of the nonlinear system described above. The sum of the squared of the error (SSE) is calculated for the cases of triangular type-1, Gaussian type-2 with uncertain σ, Gaussian type-2 with uncertain center, triangular type-2 with uncertain parameters, the proposed ellipsoidal type-2 with constant a_1 and a_2 and the proposed ellipsoidal type-2 with adaptive a_1 and a_2 over ten times of simulations. Table 4.3 shows the performances of the different controllers mentioned above. As can be seen from Table 4.3 , the performance of the T2FLC with the proposed novel membership function is comparable with the other types of T2FLCs in the presence of output noise. In addition, it can seen that T2FLC with constant values for a_1 and a_2 has better performance than the type-1 counterpart although the number of its parameters is the the same as type-1 case. The control performances of T2FLC with the proposed novel membership function for a step input and sinusoidal input can be seen in Figure 4.19 and Figure 4.20, respectively.

Table 4.3. The control accuracies of the T1FLS and T2FLS with the conventional and the proposed type-2 membership functions (SSE).

The controller type	**Step input**	**Sinusoidal input**
T1FLC with triangular membership function	5.1585	4.9236
T2FLC with Gaussian membership function with uncertain σ	5.0739	4.3351
T2FLC with Gaussian membership function with uncertain center	5.1031	4.6946
T2FLC with triangular membership function	**4.9768**	**4.3268**
T2FLC with ellipsoidal membership function with constant a_1 and a_2	5.1228	4.7217
T2FLC with ellipsoidal membership function with adaptive a_1 and a_2	5.0195	4.3923

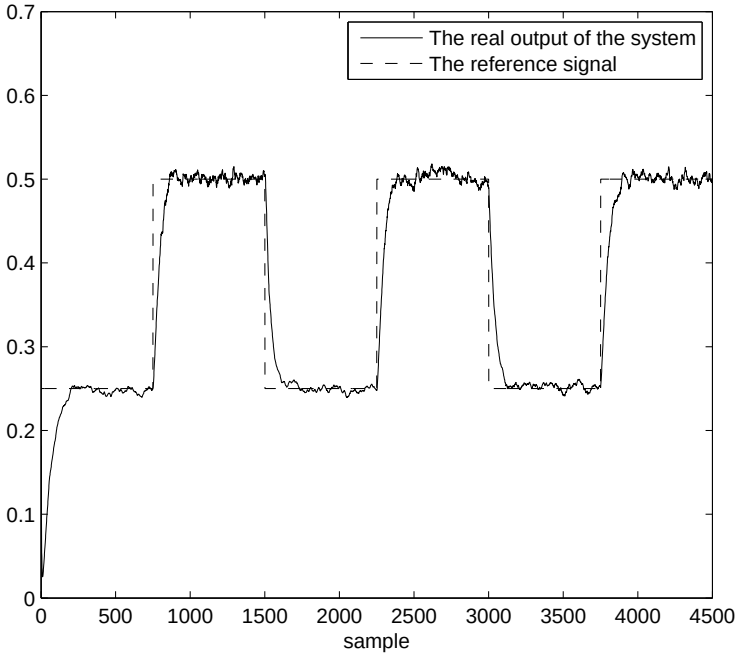


Figure 4.19. The step response of the system for the T2FLC with the proposed novel membership function.

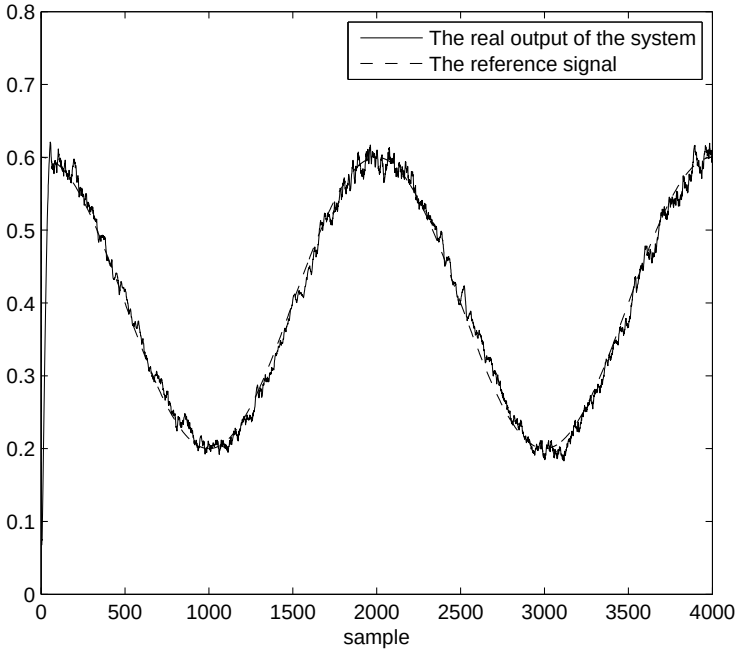


Figure 4.20. The sinusoidal response of the system for the T2FLC with the proposed novel membership function.

4.5. Conclusion

We propose a novel type-2 membership function in this study that enables us to come up with some metrics. This membership function has certain values on both ends of the support and the kernel, and some uncertain values on other points of the support. With such a function, the parameters responsible for the width of uncertainty are de-coupled from the parameters responsible for the center and the support of the membership function. This allows us to analyze how the uncertainty in the input distorts the inference of the T2FLS. The parameter update rules are derived for the proposed type-2 fuzzy membership function using the GD method. Three different applications (identification, prediction and control) are considered and the simulation results obtained indicate that the width of the proposed membership function increases as more noise power is injected to the system. The performance improvement of T2FLS over type-1 counterparts is much higher for the higher levels of noise. This fact shows that T2FLSs should be used when needed, i.e. in the presence of noise and uncertainties in the system.

5. GRADIENT-BASED LEARNING AS APPLIED TO T2FNNs

5.1. Introduction

In this section of this Ph.D. dissertation, the learning rules are derived for the adaptation of the T2FNN parameters using fuzzy clustering and gradient algorithms. The knowledge base of the designed structures is based on type-2 TSK fuzzy rules, the antecedent parts of which are characterized by type-2 fuzzy sets, and the consequent parts use crisp linear functions. Fuzzy c-means clustering algorithm is proposed for T2FLSs to determine the initial places of the membership functions to ensure that the GD algorithm used afterwards converges in a shorter time.

5.2. Type-2 Fuzzy Neural System Structure

The rules are such that the consequent parts are TSK type as indicated below:

$$\begin{aligned} &\text{IF } x_1 \text{ is } \tilde{A}_{j1} \text{ and } x_2 \text{ is } \tilde{A}_{j2} \text{ and } \ldots \text{and } x_m \text{ is } \tilde{A}_{jm} \\ &\qquad\qquad \text{THEN } y_j = \sum_{i=1}^{m} w_{ij} x_i + b_j \end{aligned} \tag{5.1}$$

where x_1, x_2, ... ,x_m are input variables, $y_j (j = 1, \ldots, n)$ are output variables, $\tilde{\text{A}}_{\text{ji}}$ denotes a type-2 membership function for j^{th} rule of the i^{th} input defined as a Gaussian membership function. The parameters w_{ij} and b_j are the parameters in the consequent part of the rules.

Due to the uncertainties in the antecedent part, the output of the type-2 fuzzy rules will have uncertainties. Let us first assume that the parameters of membership functions are represented by an uncertain mean and a fixed STD. Each membership function of the antecedent part is represented by an upper and a lower membership

function. These are denoted with $\overline{\mu}(x)$ and $\underline{\mu}(x)$, or $\overline{A}(x)$ and $\underline{\mathrm{A}}(x)$.

$$\mu_{\tilde{A}_k^i}(x_k) = [\underline{\mu}_{\tilde{A}_k^i}(x_k), \overline{\mu}_{\tilde{A}_k^i}(x_k)] = [\underline{\mu}^i, \overline{\mu}^i] \tag{5.2}$$

The inference engine of type-2 TSK FNS is proposed in [32,42]. The final output of T2FNS is determined as:

$$u(x) = [u_L(x), u_R(x)] = \sum_{f_1 \epsilon [\underline{f}_1, \overline{f}_1]} \cdots \sum_{f_1 \epsilon [\underline{f}_1, \overline{f}_1]} 1 \Big/ \frac{\sum_{i=1}^{N} f_i(x) y_i}{\sum_{i=1}^{N} f_i(x)} \tag{5.3}$$

where $u_L(x)$ and $u_R(x)$ can be computed using iterative Karnik-Mendel algorithm [2]. However the application of Karnik-Mendel algorithm may not be convenient for the control structure of a real time implementation. For this reason [42] proposes an alternative approach. We use inference engine for type-2 TSK system proposed in [42], which is given as:

$$u = \frac{p \sum_{j=1}^{N} \underline{f}_j y_j}{\sum_{j=1}^{N} \underline{f}_j} + \frac{q \sum_{j=1}^{N} \overline{f}_j y_j}{\sum_{j=1}^{N} \overline{f}_j} \tag{5.4}$$

where p and q are the design parameters that weight the sharing of lower and upper firing levels of each fired rule, N is the number of active rules. These parameters can be tuned during the design of the TSK system. For example, [43] uses the least mean square method for finding the values of p and q parameters.

5.3. Parameter Update Rules

The design of T2FNS includes the determination of the unknown parameters in the antecedent and the consequent parts of the fuzzy If-Then rules. The parameter update rules are derived for a type-2 TSK FNS with fixed mean and uncertain STD. Fuzzy clustering technique is applied for the determination of the parameters in the an-

tecedent part for modeling purposes and the gradient algorithm is used for the learning of the parameters in the antecedent and the consequent part for modeling and control purposes.

In the antecedent part, the input space is divided into a set of fuzzy regions, and in the consequent part the system behavior in those regions is described. Recently a number of different approaches have been used for designing fuzzy If-Then rules based on clustering [44–47], table look-up scheme [48,49], least-squares method [32,50], recursive least-squares method [51], gradient algorithms [19, 43, 50, 52–59], and genetic algorithms [59–61]. The fuzzy clustering is applied to design the antecedent parts, and the gradient algorithm is applied to design the consequent parts of the fuzzy rules.

The aim of clustering methods is to identify a certain group of data from a large data set, such that a concise representation of the behavior of the system is produced. Each cluster center can be translated into a fuzzy rule for identifying the class. Clustering has been used for type-1 fuzzy systems [46, 47]. For type-2 fuzzy systems, subtractive clustering and fuzzy clustering have been developed [47]. Subtractive clustering is an unsupervised clustering method, in which the number of clusters for input data points is determined by the clustering algorithm. Sometimes the number of clusters in the input space should be determined in advance. In these cases, the supervised clustering algorithms are of primary concern. Fuzzy c-means clustering is one of them. It can efficiently be used for type-1 fuzzy systems [46] owing to its simple structure and sufficient accuracy. However there are still uncertainties in the determination of the model structure, such as the number of rules and the number of variables involved in the rule premises. When fuzzy clustering is used in type-2 fuzzy systems, the computational complexity is reduced and a better distribution of the cluster centers is obtained. In this part of the dissertation, the clustering technique proposed in [44] is used for structuring the premise part of the fuzzy system. It is to be noted that fuzzy c-means does not guarantee classification for a data set that contains clusters with different densities, that have different volumes with different number of data points.

Because of the imperfect information about the data points, there exists uncertainties in the various parameters that are used for the assignment of fuzzy memberships. In [44], the representation and the management of these uncertainties are handled by varying the fuzzifier (parameter m) in the membership function. The fuzzifier m controls the amount of fuzziness in fuzzy classification. Consequently, the input data set is extended into interval type-2 fuzzy sets. For an input data point x_i, the highest and the lowest memberships are defined by using different fuzzy degrees m_1 and m_2 and the footprint of uncertainty is thus created. The primary memberships that extend data point x_i by interval type-2 fuzzy sets are determined as [44]

$$\underline{\mu}_j(x_i) = \begin{cases} \frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m_1-1}}}, \text{ if } \frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ik}}{d_{jk}}\right)} \geq \frac{1}{C} \\ \frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m_2-1}}}, \text{ otherwise} \end{cases}$$

$$\overline{\mu}_j(x_i) = \begin{cases} \frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m_1-1}}}, \text{ if } \frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ik}}{d_{jk}}\right)} < \frac{1}{C} \\ \frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m_2-1}}}, \text{ otherwise} \end{cases} \tag{5.5}$$

where $d_{jk}(d_{ik})$ denotes the distance between cluster $j(k)$ and data point x_i, C is the number of clusters and $\underline{\mu}_j(x_i)$ and $\overline{\mu}_j(x_i)$ are the lower and the upper membership of the data point. Updating of the cluster centers are performed by the extension of interval type-2 fuzzy sets while executing fuzzy c-means algorithm. The interval type-2 fuzzy set is incorporated into fuzzy c-means clustering.

The use of the fuzzifiers m_1 and m_2 result in two different objective functions to be minimized.

$$J_{m_1} = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m_1} d_{ji}^2, \text{ and } J_{m_2} = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m_2} d_{ji}^2 \tag{5.6}$$

where $1 \leq m_1 \leq m_2 \leq \infty$.

Fuzzy partitioning is carried out through an iterative optimization of the objective functions Equation 5.6, with the update of membership function and the cluster centers. For a detailed analysis, the reader can refer to [44]. After the design of the antecedents parts by fuzzy clustering, the GD algorithm is applied to design the consequent parts of the fuzzy rules. In what follows, the parameter update rules are derived for a type-2 TSK FNS. At the first step, the output error is calculated:

$$E = \frac{1}{2}\sum_{i=1}^{O}(u_i^d - u_i)^2 \tag{5.7}$$

where O is number of output signals of the network (in the given case $O = 1$), u_i^d and u_i are the desired and the current output values of the network, respectively.

The parameters w_{ij}, b_j and $c1_{ij}$, $c2_{ij}$ and σ_{ij} $(i = 1, .., m, j = 1, .., n)$ are adjusted using the following formulas:

$$w_{ij}(t+1) = w_{ij}(t) - \gamma\frac{\partial E}{\partial w_{ij}}; \quad b_j(t+1) = b_j(t) - \gamma\frac{\partial E}{\partial b_j} \tag{5.8}$$

$$c1_{ij}(t+1) = c1_{ij}(t) - \gamma\frac{\partial E}{\partial \underline{c}_{ij}}; \quad c2_{ij}(t+1) = c2_{ij}(t) - \gamma\frac{\partial E}{\partial \overline{c}_{ij}} \tag{5.9}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \gamma\frac{\partial E}{\partial \sigma_{ij}} \tag{5.10}$$

where γ is the learning rate, m is the number of input signals of the network (input neurons) and n is the number of rules (hidden neurons). The derivatives in Equation 5.8 are determined by the following formulas:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \frac{\partial E}{\partial u}\frac{\partial u}{\partial y_j}\frac{\partial y_j}{\partial w_{ij}} \\ \frac{\partial E}{\partial b_j} &= \frac{\partial E}{\partial u}\frac{\partial u}{\partial y_j}\frac{\partial y_j}{\partial b_j} \end{aligned} \tag{5.11}$$

The derivatives in Equation 5.9 and Equation 5.10 are determined by the following formulas:

$$\frac{\partial E}{\partial \sigma_{ij}} = \sum_j \frac{\partial E}{\partial u}\left[\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial \sigma_{ij}} + \frac{\partial u}{\partial \overline{f}_j}\frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}}\frac{\partial \overline{\mu_j}}{\partial \sigma_{ij}}\right] \tag{5.12}$$

$$\begin{aligned}\frac{\partial E}{\partial c1_{ij}} &= \sum_j \frac{\partial \mathrm{E}}{\partial u}\left[\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial c1_{ij}} + \frac{\partial u}{\partial \overline{f}_j}\frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}}\frac{\partial \overline{\mu}_{ij}}{\partial c1_{ij}}\right] \\ \frac{\partial E}{\partial c2_{ij}} &= \sum_j \frac{\partial \mathrm{E}}{\partial u}\left[\frac{\partial u}{\partial \underline{f}_j}\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}}\frac{\partial \underline{\mu}_{ij}}{\partial c2_{ij}} + \frac{\partial u}{\partial \overline{f}_j}\frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}}\frac{\partial \overline{\mu}_{ij}}{\partial c2_{ij}}\right]\end{aligned} \tag{5.13}$$

The parameters of the T2FNS can thus be updated using Equations 5.8-5.10 together with Equations 5.11-5.13.

As mentioned above, the parameter p and q in Equation 5.4 enable us to adjust the lower or the upper portions in the final output. The values of p and q are optimized during learning from an initial value of 0.5 using:

$$p(t+1) = p(t) - \gamma\frac{\partial E}{\partial p} \tag{5.14}$$

$$q(t+1) = q(t) - \gamma\frac{\partial E}{\partial q} \tag{5.15}$$

where

$$\frac{\partial E}{\partial p} = (u - u^d)\frac{\underline{f}_j}{\sum_{j=1}^{n}\underline{f}_j} \tag{5.16}$$

$$\frac{\partial E}{\partial q} = (u - u^d)\frac{\overline{f}_j}{\sum_{j=1}^{n}\overline{f}_j} \tag{5.17}$$

where

$$\frac{\partial E}{\partial u} = u(t) - u^d(t); \frac{\partial u}{\partial \underline{f}_j} = p\frac{y_j - \underline{u}}{\sum_{j=1}^{n} \underline{f}_j}; \frac{\partial u}{\partial \overline{f}_j} = q\frac{(y_j - \overline{u})}{\sum_{j=1}^{n} \overline{f}_j}$$

$$\underline{\mathrm{u}} = \frac{\sum_{j=1}^{n} \underline{f}_j y_j}{\sum_{j=1}^{n} \underline{f}_j}; \quad \overline{u} = \frac{\sum_{j=1}^{n} \overline{f}_j y_j}{\sum_{j=1}^{n} \overline{f}_j} \tag{5.18}$$

If t-norm *prod* operator is used, then

$$\frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}} = \prod_{k=1,k\neq i}^{N1} \underline{\mu}_{kj}; \frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}} = \prod_{k=1,k\neq i}^{N1} \overline{\mu}_{kj} \tag{5.19}$$

where $i = 1, \ldots, N1$, $k = 1, \ldots, N1$, and $j = 1, \ldots, N2$.

For t-norm *min* operator:

$$\text{if } (\underline{\mu}_{kj} - \underline{\mu}_{ij}) \geq 0, \text{ then } \frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}} = 1, \text{ else } \frac{\partial \underline{f}_j}{\partial \underline{\mu}_{ij}} = 0,$$

$$\text{if } (\overline{\mu}_{kj} - \overline{\mu}_{ij}) \geq 0, \text{ then } \frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}} = 1, \text{ else } \frac{\partial \overline{f}_j}{\partial \overline{\mu}_{ij}} = 0;$$

where $i = 1, \ldots, N1$, $k = 1, \ldots, N1$, and $j = 1, \ldots, N2$.

Upper and lower membership functions can be written as follows:

$$\underline{\mu}_{ij}(x) = \begin{cases} G(c2_{ij}, \sigma_{ij}, x_i), & x_i \leq \frac{c1_{ij}+c2_{ij}}{2} \\ G(c1_{ij}, \sigma_{ij}, x_i), & x_i > \frac{c1_{ij}+c2_{ij}}{2} \end{cases}$$

$$\overline{\mu}_{ij}(x) = \begin{cases} G(c1_{ij}, \sigma_{ij}, x_i), & x_i < c1_{ij} \\ 1, & c1_{ij} \leq x_i \leq c2_{ij} \\ G(c2_{ij}, \sigma_{ij}, x_i), & x_i > c2_{ij} \end{cases} \tag{5.20}$$

where $G(c_{ij}, \sigma_{ij}, x_i)$ is determined as:

$$G(c_{ij}, \sigma_{ij}, x_i) = \exp\left(-\frac{1}{2}\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right) \tag{5.21}$$

Then

$$\frac{\partial \overline{\mu}_j(x_i)}{\partial c1_{ij}} = \begin{cases} G(c1_{ij}, \sigma_{ij}, x_i)\frac{(x_i - c1_{ij})}{\sigma_{ij}^2}, & x_i < c1_{ij} \\ 0, & c1_{ij} \leq x_i \leq c2_{ij} \\ 0, & x_i > c2_{ij} \end{cases}$$

$$\frac{\partial \underline{\mu}_j(x_i)}{\partial c1_{ij}} = \begin{cases} 0, x_i \leq \frac{c1_{ij}+c2_{ij}}{2} \\ G(c1_{ij}, \sigma_{ij}, x_i)\frac{(x_i - c1_{ij})}{\sigma_{ij}^2}, x_i > \frac{c1_{ij}+c2_{ij}}{2} \end{cases} \tag{5.22}$$

$$\frac{\partial \overline{\mu}_j(x_i)}{\partial c2_{ij}} = \begin{cases} 0, x_i < c1_{ij} \\ 0, c1_{ij} \leq x_i \leq c2_{ij} \\ G(c2_{ij}, \sigma_{ij}, x_i)\frac{(x_i - c2_{ij})}{\sigma_{ij}^2}, & x_i > c2_{ij} \end{cases}$$

$$\frac{\partial \underline{\mu}_j(x_i)}{\partial c2_{ij}} = \begin{cases} G(c2_{ij}, \sigma_{ij}, x_i)\frac{(x_i - c2_{ij})}{\sigma_{ij}^2}, & x_i \leq \frac{c1_{ij}+c2_{ij}}{2} \\ 0, & x_i > \frac{c1_{ij}+c2_{ij}}{2} \end{cases} \tag{5.23}$$

$$\frac{\partial \overline{\mu}_j(x_i)}{\partial \sigma_{ij}} = \begin{cases} G(c1_{ij}, \sigma_{ij}, x_i)\frac{(x_i - c1_i)^2}{\sigma_{ij}^3}, x_i < c1_{ij} \\ 0, c1_{ij} \leq x_i \leq c2_{ij} \\ G(c2_{ij}, \sigma_{ij}, x_i)\frac{(x_i - c2_{ij})^2}{\sigma_{ij}^3}, \ x_i > c2_{ij} \end{cases}$$

$$\frac{\partial \underline{\mu}_j(x_i)}{\partial \sigma_{ij}} = \begin{cases} G(c2_{ij}, \sigma_{ij}, x_i)\frac{(x_i - c2_{ij})^2}{\sigma_{ij}^3}, \ x_i \leq \frac{c1_{ij}+c2_{ij}}{2} \\ G(c1_{ij}, \sigma_{ij}, x_i)\frac{(x_i - c1_{ij})^2}{\sigma_{ij}^3}, \ \ x_i > \frac{c1_{ij}+c2_{ij}}{2} \end{cases} \tag{5.24}$$

One important problem in the learning algorithms is the convergence. The convergence of the GD method depends on the selection of the initial value of the learning rate. Usually, this value is selected within the interval $[0, 1]$. A large value of the learning rate may lead to unstable learning, while a small value of the learning rate results in a slow learning speed. In this study, an adaptive approach is used to update this parameters. That is, the learning of the T2FNS parameters is started with a small value of the learning rate γ. During learning, γ is increased if the value of the change of error $\Delta E = E(k) - E(k+1)$ is positive, and decreased if negative. This strategy ensures a stable learning for the T2FNS, guarantees the convergence and speeds up the learning.

5.4. Simulation and Experimental Studies

5.4.1. Mathematical Model of the Permanently Excited DC Motor

The experimental setup used in this study [62] 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).

The block diagram of the servo system is shown in Figure 5.2, the nomenclature of the symbols used being given in Table 1. From the block diagram of the system

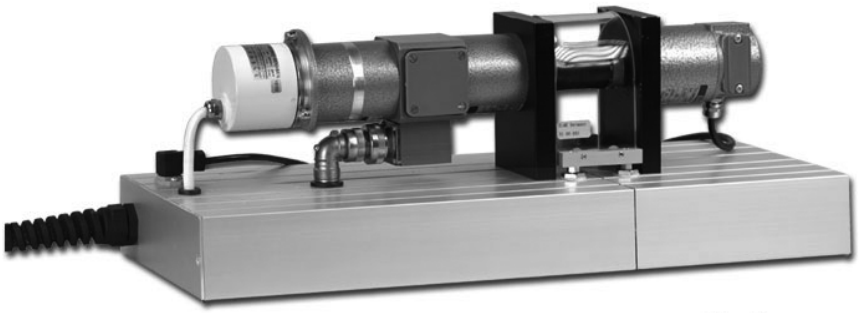


Figure 5.1. A servo system setup.

depicted in Figure 5.2, the transfer function of the overall system can readily be derived as follows:

$$\begin{aligned} \omega(s) &= \frac{1}{C\Phi}\frac{1}{1+T_M s+T_M T_A s^2}U_A(s) \\ &- \frac{R_A}{K_M C\Phi}\frac{1+T_A s}{1+T_M s+T_M T_A s^2}M_L(s) \end{aligned} \tag{5.25}$$

where

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

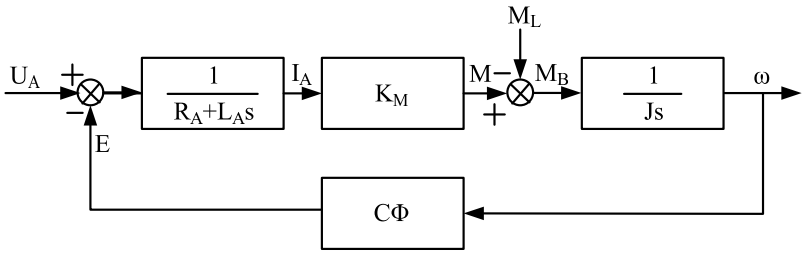


Figure 5.2. Block diagram of the motor with load.

The numerical values used in this study are: Armature terminal voltage= $24V$, Rated torque= $0.096Nm$, Moment of inertia of the system= $80.45\text{x}10^{-6}kgm^2$, Armature inductance= $3mH$, Armature resistance= 3.13Ω, Back emf constant= $0.06Vs$, Torque constant= $0.06Nm/A$.

Table 5.1. Nomenclature.

Name	Description
U_A	Armature terminal voltage
E	Induced electromotive force
I_A	Armature current
R_A	Armature winding resistance
L_A	Armature winding inductance
C	Motor constant
Φ	Magnetic excitation
ω	Speed of the rotor
M_L	Load torque
M_B	Acceleration torque
M	Torque produced by the motor
J	Moment of inertia of the motor
T_A	Electrical time constant
T_M	Mechanical time constant

5.4.2. Identification of the Dynamical Plants

The dynamical process of finding the input-output relations for a system is called *identification* in the literature. In identification, the clustering involves the determination of clusters in data space and the translation of these clusters into fuzzy rules such that the model obtained is close to the identified system. For this purpose, first the fuzzy clustering approach is applied to the input data points of the plant in order to determine the cluster centers. The obtained cluster centers are then used in order to organize the premise parts of the fuzzy rules. It is to be noted that it would be possible not to use a clustering algorithm and spread the membership functions onto the universe of discourse randomly. However, in this case, the GD algorithm would require more time to find the final places of the membership functions. In this study, the fuzzy c-means clustering algorithm is used to determine the initial places of the

membership functions so that the GD algorithm converges in a shorter time.

In order to prevent the system from a possible saturation, the input-output data set is collected in a closed loop fashion (Figure 5.3) using a conventional PI controller. The coefficients of the PI controller are $K_P = 1$ and $K_I = 5$ which are tuned by trial-and-error method. The sampling period of the real-time system is equal to 0.01 seconds which satisfies the Nyquist–Shannon sampling theorem for the real-time set up. This value is also proper for the stability of the digital control system. The period of the reference signal is 750 samples, and the mathematical expression for the reference signal can be seen in Equation 5.26:

$$\text{Reference}(k) = \begin{cases} 0.4\text{pu} & \text{if } 0 < k \leq 150 \\ 0.5\text{pu} & \text{if } 150 < k \leq 300 \\ 0.6\text{pu} & \text{if } 300 < k \leq 450 \\ 0.55\text{pu} & \text{if } 450 < k \leq 600 \\ 0.45\text{pu} & \text{if } 600 < k \leq 750 \end{cases} \tag{5.26}$$

The inputs to the T2FNS based identifier are the input signal $u(k)$, and 1-step delayed values of the real plant outputs $y(k)$. The problem is to find the proper parameter values for the T2FNS structure such that the difference between the plant output $y(k)$ and the identifier output $y_n(k)$ will be minimized for all input values $u(k)$.

The training of the T2FNS is carried out for 200 epochs with 1000 time steps in each epoch. As a performance criterion the RMSE defined in Equation 5.27 is used.

$$RMSE = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(x_i^d - x_i)^2} \tag{5.27}$$

where K represents the total number of the samples.

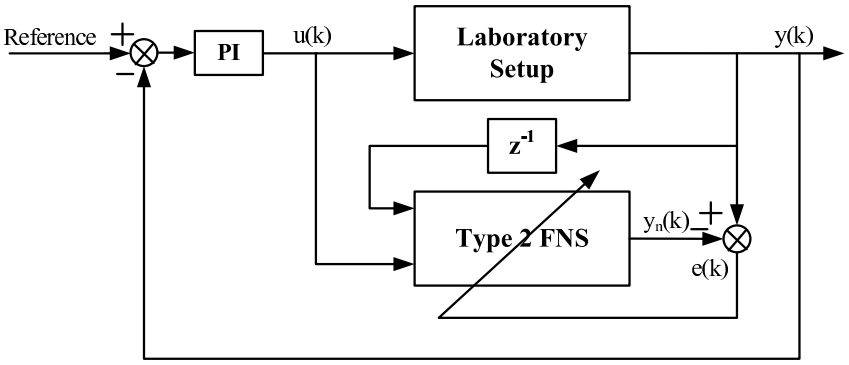


Figure 5.3. The block diagram of the identifier.

Figure 5.4 shows the input to the identifier. While Figure 5.5 shows the RMSE values versus epoch number, which indicates a stable learning with the gradient based algorithm, Figure 5.6 demonstrates the output of the model and the real-time system. As can be seen from Figure 5.6, the T2FNS gives accurate modeling results.

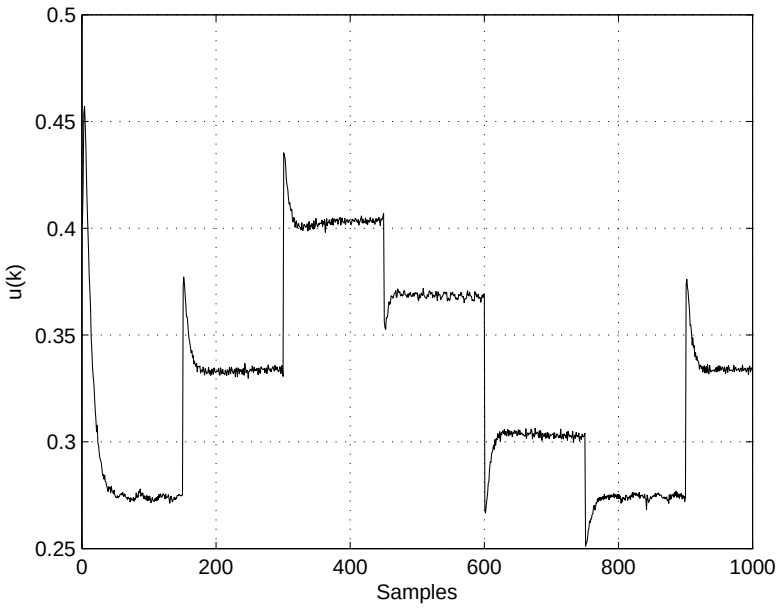


Figure 5.4. The input of the identifier.

Identification is performed with 4 fuzzy rules. These fuzzy rules are constructed using two clusters that are determined for each input variables. Different combinations of these clusters are considered for the organization of the premise parts of the fuzzy

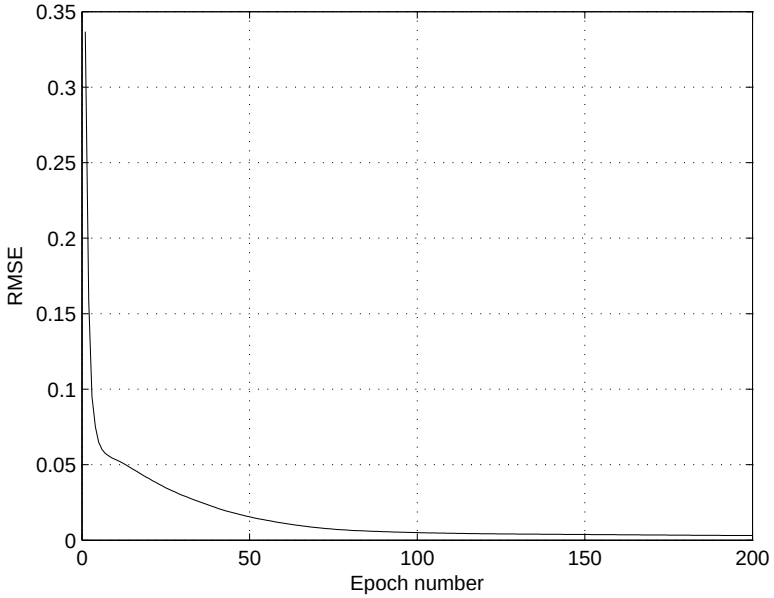


Figure 5.5. RMSE versus epoch number.

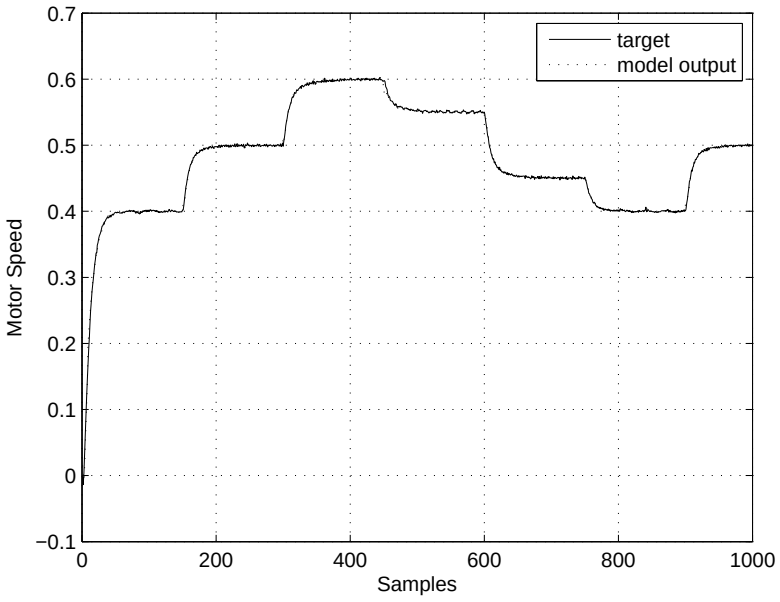


Figure 5.6. Output of the model and the real-time system.

rules. As a result of clustering of premise parts and training of consequent parts, the four fuzzy rules are generated and the parameters of the T2FNS are determined.

After the determination of the parameters in the antecedent part, the gradient algorithm is applied for the learning of the parameters in the antecedent and the consequent part. The initial values of w, a and b are selected randomly within the interval [-1, 1].

Table 5.2 shows the centers of the type-2 membership functions after the clustering process. The number of rows shows the number of membership functions for each input.

Table 5.2. The values of the centers of the input membership functions for the identification case after clustering.

	input 1	**input 2**
$c1_{ij}$	0.2811	0.3620
	0.4074	0.5393
$c2_{ij}$	0.2920	0.3688
	0.4079	0.5498

5.4.3. Control of the Dynamical System

The structure of the control scheme is shown in Figure 5.7. The variable $y(k)$ is the output signal of the plant, $g(k)$ is the set-point signal, $e(k)$, $\Delta e(k)$ and $\sum e(k)$ are the error, the change in the error and the sum of the error, respectively. D indicates differentiation operation and $\sum$ indicates integration operation.

AMIRA DR300 DC motor experimental setup is used to test and compare the performances of the T1FNS and T2FNS controllers. The sampling time is set to 20 ms for all the experiments. The speed of the motor and the load torque are scaled to [-1,1]. In order to determine the efficiency and the accuracy of the proposed controllers,

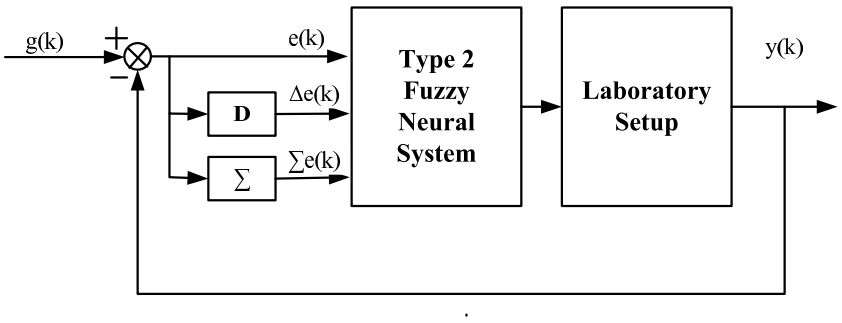


Figure 5.7. Structure of type-2 TSK fuzzy neural system.

three different types of load conditions are considered. Figure 5.8 represents the speed responses of the motor for T1FNS and T2FNS controllers. The corresponding load condition starts with a value of 0.2pu, and increases suddenly to 0.5pu on 7^{th}s, and then comes back to 0.2pu on 14^{th}s. As can be seen from Figure 5.8, T1FNS and T2FNS adapt their parameters when the load on the motor changes suddenly in 7^{th}s and in 14^{th}s. In order to initialize w_{ij}, b_j, $c1_{ij}$, $c2_{ij}$ and σ_{ij}, first the simulation model of the setup is run for 200 epochs.

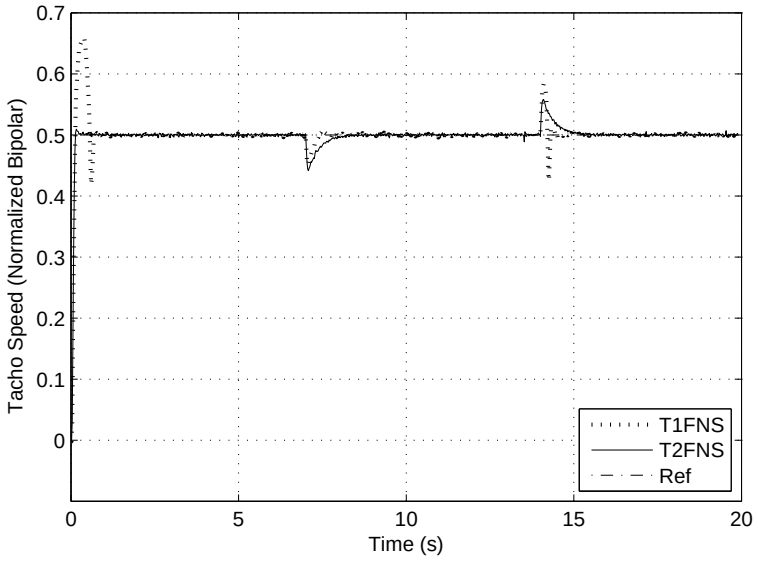


Figure 5.8. The speed responses of the motor for T1FNS and T2FNS.

Figure 5.9 shows the speed response of the motor under the sinusoidal load condition, i.e. $M_L(t) = 0.35 + 0.15sin(0.75t)$. Besides, Figure 5.10 shows the speed response

of motor under load condition which is proportional to the square of the speed, i.e. $M_L(t) = 1.5(Speed)^2$. This type of load corresponds to the load-torque characteristics of centrifugal fans, pumps and blowers. It can be inferred from Table 5.3 that T2FNS controller gives better performance than the T1FNS controller in all load conditions. In this table, load type 1, load type 2, and load type 3 correspond to the step change, sinusoidal, and proportional to the square of the speed of load torque, respectively.

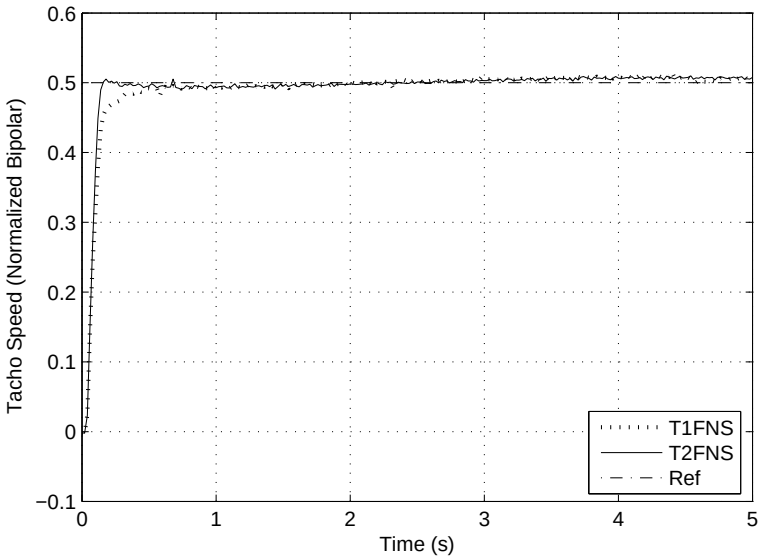


Figure 5.9. The speed responses of the motor for T1FNS and T2FNS.

Table 5.4-5.5 show the centers and standard deviations before and after control process, respectively.

Table 5.3. Square of the error values at each time step.

	Load type 1	**Load type 2**	**Load type 3**
T1FNN	1.330	1.021	0.856
T2FNN	0.956	0.953	0.841

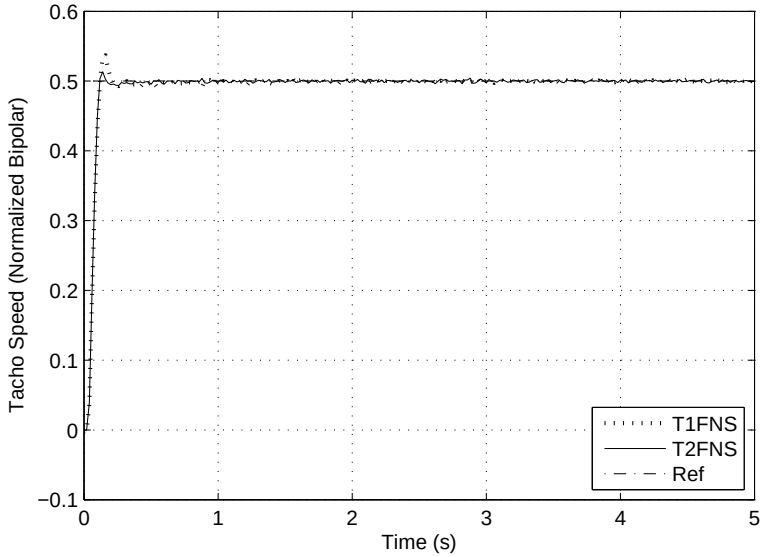


Figure 5.10. The speed responses of the motor for T1FNS and T2FNS.

Table 5.4. Initial values of the centers and standard deviation of the input membership functions for the control case.

$c1_{ij}$			$c2_{ij}$			o_{ij}		
e(k)	$\Delta e(k)$	$\sum e(k)$	e(k)	$\Delta e(k)$	$\sum e(k)$	e(k)	$\Delta e(k)$	$\sum e(k)$
-0.3	-0.1	-0.5	-0.2	-0.05	-0.3	0.2	0.1	0.3
0	0	0.5	0.1	0.05	0.7	0.2	0.1	0.3
0.3	0.1	1.5	0.4	0.15	1.7	0.2	0.1	0.3

Table 5.5. Final values of the centers and standard deviation of the input membership functions for the control case.

$c1_{ij}$			$c2_{ij}$			o_{ij}		
-0.3006	-0.1006	-0.5004	-0.2631	-0.0896	-0.3241	0.1022	0.0735	0.2812
0.0875	0.0596	0.6010	-0.1253	-0.0215	0.6474	-0.0959	0.1276	0.3118
0.3021	0.2550	1.5050	0.3543	0.1442	1.6992	0.1740	-0.0418	0.2922

5.5. Conclusion

In this part of the dissertation a T2FNS is proposed for the identification and the control of a servo system with time-varying and nonlinear load conditions. The structure of the T2FNS is presented and the parameter update rules of the structure are derived based on fuzzy clustering and GD algorithms. In order to improve the performance of the type-2 fuzzy structure, a fuzzy c-means clustering algorithm is used to determine the initial positions of the membership functions. The proposed control algorithm is tested on a real time DC motor system under different nonlinear load conditions. The experimental results obtained indicate the potential of the proposed T2FNS structure. It is seen that type-2 fuzzy control algorithm can handle the uncertainties in the system effectively with a much better transient performance and smaller RMSE values compared to a conventional T1FNS controller.

6. LEVENBERG-MARQUARDT METHOD AS APPLIED TO T2FNNs

6.1. Introduction

A new training approach based on the LM algorithm is proposed for T2FNNs. While conventional GD algorithms use only the first order derivative, the proposed algorithm used in this thesis benefits from the first and the second order derivatives which makes the training procedure faster. Besides, this approach is more robust than the other techniques that use the second order derivatives, e.g. Gauss-Newton's method. The training algorithm proposed is tested on the training of a T2FNN used for the prediction of a chaotic Mackey-Glass time series. The results show that the learning algorithm proposed not only results in faster training but also in a better forecasting accuracy.

The most commonly used training algorithms are based on first derivatives, such as the GD based one. If higher derivatives are used, the resulting iterative algorithm may perform better. For instance, Gauss-Newton's method uses first and second order derivatives and it gives a better performance when compared with the steepest descent method. On the other hand, if the Hessian matrix, F, is not positive definite, then the search direction may not point in a descent direction. To guarantee that the search direction is a descent direction, a simple modification, namely LM modification, can be introduced. The resulting algorithm is known to be more robust than the Gauss-Newton's method but slower.

LM algorithm is a promising method to train FNNs, and there exists number of papers based on that algorithm in the literature. New training algorithms for B-spline neural network and fuzzy rule-based systems using LM approach are discussed in [63], and it has been shown that the standard error-back propagation algorithm exhibits a very poor performance when compared to LM approach. In [64], FNNs trained with LM algorithm exhibits faster training speed and better forecasting capability. A

modified version of LM algorithm is introduced in [65], and it is shown that the modified algorithm is more robust. The novel training algorithm also describes a simple method to compute the Jacobian matrix which is considered to be the most difficult step in implementing LM algorithm.

6.1.1. Levenberg-Marquardt Training Method

The well known back-propagation (BP) algorithm based on GD is very slow because it must use small learning rates for stable learning and suffers from some other drawbacks [66]. Logically, one would like to take large steps down the gradient at locations where the gradient is small (the slope is gentle) and conversely, take small steps when the gradient is large, so as not to rattle out of the minima. With the simple BP update rule, just the opposite of this is done [67]. Another issue is that the curvature of the error surface may not be the same in all directions. For example, if there is a long and narrow valley in the error surface, the component of the gradient in the direction that points along the base of the valley is very small while the component along the valley walls is quite large. This results in motion more in the direction of the walls even though we have to move a long distance along the base and a small distance along the walls.

There are different approaches in the literature to improve the performance of BP. Speed and reliability of BP can be increased by techniques called momentum and adaptive learning rates. The momentum technique helps the network to get out, if stacked in a local minima. By the use of adaptive learning rates it is possible to decrease the learning time and improve the accuracy. In this paper we used a LM modification of BP algorithm. By using LM, it is possible to achieve the results much faster. The parameter update rule for LM is as follows:

$$\triangle\, w = (J^T J + \mu I)^{-1} J^T e \tag{6.1}$$

in which J is the Jacobian matrix of derivatives of network function error, e is the error and is calculated as the difference between desired and estimated net outputs, I is the

identity matrix with proper dimensions. The number of columns in J matches the number of training patterns. μ is a scalar which plays an important role in the update process of LM. If it is very large, the above expression approximates GD, while if it is small this expression becomes the Gauss-Newton method. The coefficient is changed in such a way as to combine good features of both algorithms: GD algorithm (which does not require that the initial values of parameters are well chosen), and Gauss-Newton algorithm (which has quadratic convergence near an error minimum). As long as the error gets smaller, μ is made bigger, but, once the error starts increasing, it is made smaller. The LM is much faster than the GD algorithm, on which the standard BP algorithm is based. However, it is computationally more expensive than the GD algorithm. The pseudo code for LM and the selection of μ can be summarized below [68]:

(i) Initialize the weights and the parameter μ ($\mu = 0.01$ is appropriate)
(ii) Compute the sum of squared errors over all inputs F(w)
(iii) Compute the Jacobian matrix J
(iv) Solve Equation 6.1 to obtain the increment of weights Δw
(v) Recompute the sum of squared error using the $w + \Delta w$ and Judge
If $F(w + \Delta w) < F(w)$ Then $w = w + \Delta w$ $\mu = \mu\beta$, $0 < \beta < 1$ and go to Step 2
Else
$\mu = \mu/\beta$, $0 < \beta < 1$ and go to step 4
End If ($\beta = 0.1$ is appropriate)

6.2. Simulation Results

6.2.1. Prediction of Chaotic Mackey-Glass Time Series

In this section the LM algorithm is used to train the parameters of T2FLS, and it is used to predict the noisy chaotic Mackey-Glass time series. This chaotic system is a well-known benchmark problem in the literature, and it can be described by the

following dynamic equation [37]:

$$\dot{x}(t) = 0.2\frac{x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{6.2}$$

The numerical values selected for the chaotic system above are $\tau = 17$, $x(0) = 1.2$ in this study. The following approximation is used:

$$\dot{x}(t) = \frac{x(k+1) - x(k)}{T_s} \tag{6.3}$$

where $T_s = 1$.

The predictor goal is to predict $x(t+1)$ using the inputs $x(t-3), x(t-2), x(t-1)$ and $x(t)$. For each input two membership functions are used. Consequently the number of rules that the predictor uses is 16. The number of training data is selected as 750, and the number of test data is 335. In order to study the effect of noise in the proposed system, six experiences are performed in which the data are corrupted with different levels of signal to noise ratio (SNR) using three different optimization methods. The well-known formula below is used as SNR:

$$SNR = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_n^2}\right) \tag{6.4}$$

where σ_s^2 is the variance of the signal and σ_n^2 is the variance of the noise. Figure 6.1 shows the actual and the noisy Mackey-Glass chaotic time series data with SNR equals to 0dB.

Figure 6.2 shows the primitive data set and the output of the T2FLS trained with LM algorithm. As can be seen, the prediction accuracy of T2FLS using LM is quite satisfactory for such a noisy data. The prediction error values between the model and the T2FLS can be observed in Figure 6.3. Table 6.1 shows the prediction accuracies of the T2FLS with the proposed type-2 membership function and T1FLS using LM. To be able to make a fair comparison, triangular type membership function is used in

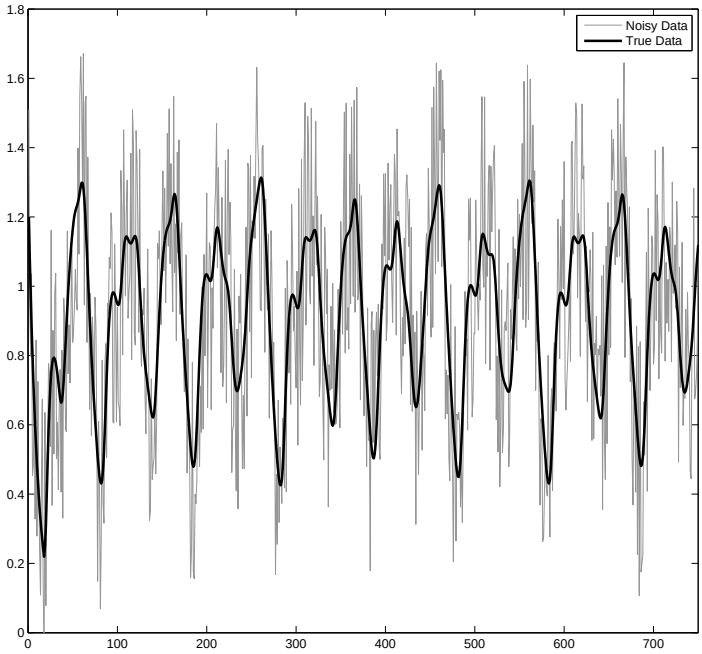


Figure 6.1. The actual Mackey-Glass time series and the noisy data (SNR = 0dB).

T1FLS. The prediction performance of T2FLS is better than T1FLS for all levels of noise on the data.

Figures 6.4a-6.4d show the proposed novel type-2 fuzzy membership functions before (dotted line) and after (solid line) the training using LM. The related figures show the type-2 fuzzy membership functions for $x(t-3), x(t-2), x(t-1)$ and $x(t)$, respectively. The parameters of the type-2 membership functions are initially set to $a_1 = a_2 = 1$ which means that they are type-1. Once the training is finished, it is seen that the widths of the proposed membership functions are grown which means that they become type-2.

Figure 6.5 shows $\sum |a_1 - a_2|$ (the sum being over the eight membership functions used for the 4 inputs) after the training of the T2FLS by LM algorithm. It can be seen that the widths of the proposed membership functions are larger for higher levels of noise which is an expected behavior from the T2FLS. Table 6.2 shows the prediction accuracies of the T2FLS with the proposed type-2 membership function using LM and

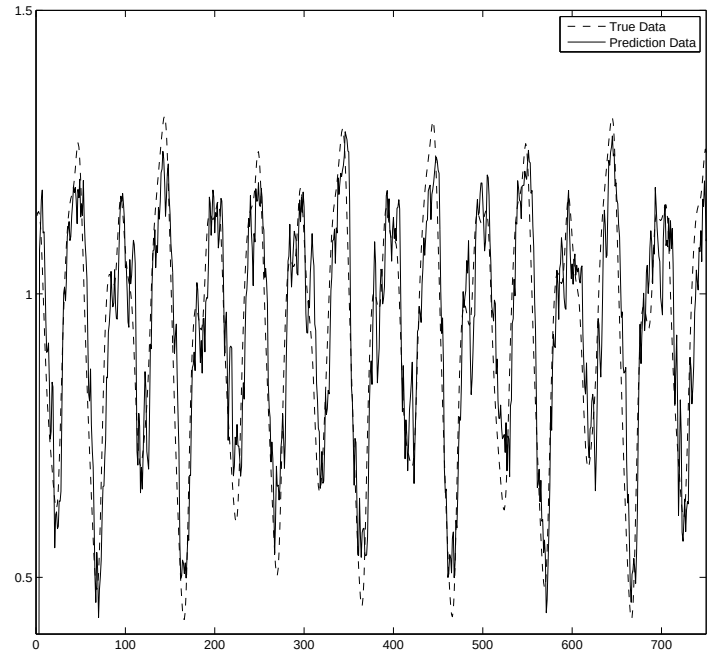


Figure 6.2. The actual Mackey-Glass time series and the output of the T2FLS (SNR=10dB).

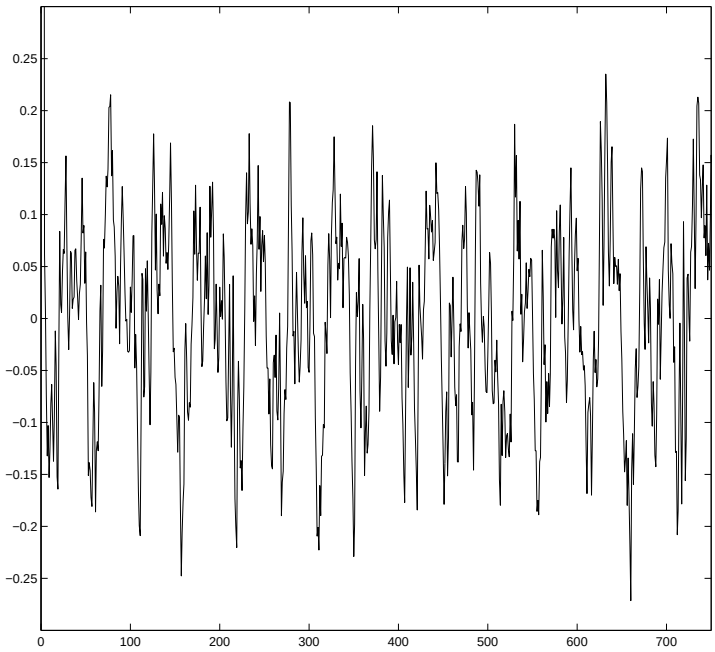


Figure 6.3. The prediction error of the model and the original predicted data (SNR=10dB).

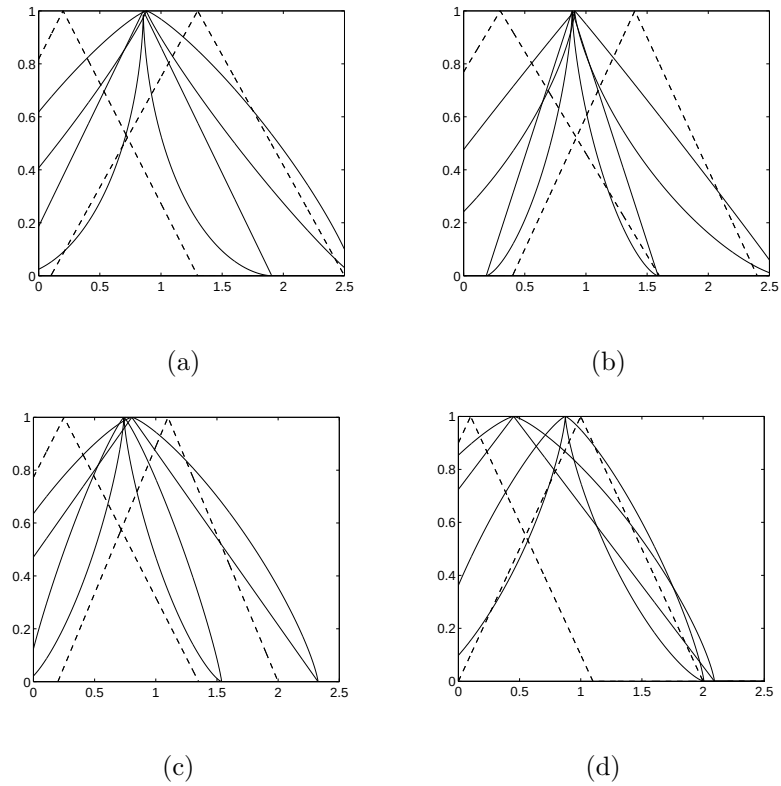
Table 6.1. The prediction accuracies of the T2FLS with the proposed type-2 membership function and T1FLS using LM.

	Train		**Test**	
	T2FLS	**T1FLS**	**T2FLS**	**T1FLS**
0dB	0.1256	0.1264	0.1234	0.1244
2dB	0.1177	0.1194	0.1172	0.1181
4dB	0.1108	0.1122	0.1104	0.1106
6dB	0.1036	0.1056	0.1047	0.1042
8dB	0.0976	0.0995	0.0969	0.0984
10dB	0.0914	0.0934	0.0913	0.0927

GD. The figures indicate that the prediction performance of T2FLS trained with LM is always better than T2FLS trained with GD for the training data set and is mostly better for the testing data set. In order to have a comparison of the computational time of LM and GD algorithms, the computation time is recorded when the error decreases a specific value. It is seen that this time is 70 seconds for LM and 84 seconds for GD. This shows that the LM algorithm is a computationally more effective algorithm when compared to GD.

Table 6.2. The prediction accuracies of the T2FLS with the proposed type-2 membership function using GD and LM.

	Train		**Test**	
	LM	**GD**	**LM**	**GD**
0dB	0.1256	0.1298	0.1234	0.1254
2dB	0.1177	0.1192	0.1172	0.1171
4dB	0.1108	0.1120	0.1104	0.1103
6dB	0.1036	0.1054	0.1047	0.1040
8dB	0.0976	0.0992	0.0969	0.0982
10dB	0.0914	0.0930	0.0913	0.0924



(a) (b)

(c) (d)

Figure 6.4. The proposed novel type-2 fuzzy membership functions before (dotted line) and after (solid line) the training using LM algorithm (a) shows the membership function for $x(t)$, (b) shows the membership function for $x(t-1)$, (c) shows the membership function for $x(t-2)$ and (d) shows the membership function for $x(t-3)$.

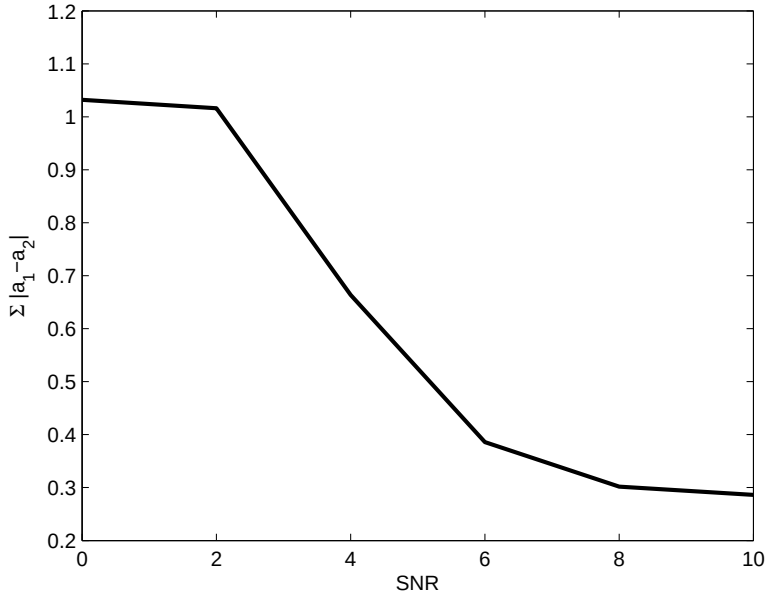


Figure 6.5. The width of the proposed novel type-2 fuzzy membership function after training using LM w.r.t. the noise level.

6.3. Conclusions

A novel type-2 membership function with certain values on both ends of the support and the kernel, and some uncertain values on other values of the support has been proposed. The parameter update rules of a T2FLS with such a membership function are derived using LM algorithm. In order to show the effectiveness of the use of LM for training the parameters of T2FLSs, a gradient based training method is also used. The results show that LM outperforms the gradient-based method, while its computation time is less than the computational time of GD algorithm.

T2FLS trained by LM and GD methods are used for the prediction of Mackey-Glass chaotic system in the presence of measurement noise. It is shown that the performance of T2FLS is better than the performance of type-1 counterpart in the presence of higher levels of noise and uncertainties. The simulation results also show that the width of the proposed membership function increases as more noise power is injected to the system which means that the performance improvement of T2FLS over type-1 counterpart is much higher for the higher levels of noise. This fact suggests that T2FLSs should be preferred in the presence of higher levels of uniform noise and uncertainties in the system.

7. SMC THEORY-BASED ONLINE LEARNING AS APPLIED TO T2FNNs

7.1. Introduction

A novel SMC theory-based learning algorithm is proposed to train an interval T2FNN using triangular and Gaussian membership functions. The structure considered is an interval type-2 TSK FLS in which the antecedents are type-2 fuzzy sets, and consequents are crisp numbers (A2-C0). In the proposed learning algorithm, instead of trying to minimize an error function as is generally done, the weights of the FNN are tuned by the proposed algorithm in a way that the error is enforced to satisfy a stable equation. The update rules to achieve this are derived and the convergence of the parameters is proved by the use of Lyapunov stability method. To illustrate the applicability and the efficacy of the proposed method, it is tested on the velocity control of an electro hydraulic servo system (EHSS) in presence of flow nonlinearities and internal friction for the triangular membership function case and on a Duffing oscillator model for the Gaussian membership function case. The simulation studies indicate that the type-2 fuzzy neuro structure with the proposed learning algorithm results in a better performance than its type-1 fuzzy counterpart. Moreover, the proposed learning algorithm is easy to implement due to its simple structure which makes it faster than the other existing learning algorithms seen in the literature.

A number of different methods, *e.g* GD [69] and evolutionary algorithms [70, 71] have been used for the parameter optimization of FNNs. The GD method 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 repetitive algorithms, a number of numerical robustness issues may emerge when they are applied over long periods of time [5]. In addition to these drawbacks, the tuning process can easily be trapped into a local minimum [6].

To alleviate the problems mentioned, the use of evolutionary approaches have been suggested [71]. 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.

SMC is an approach that guarantees the robustness of a system in the case of external disturbances, parameter variations and uncertainties and as such has attracted the attention of many researchers to guarantee robustness in computationally intelligent architectures [72,73]. 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 the predefined function of the error is zero [74]. SMC-based learning algorithms can not only make the overall system more robust but also ensure faster convergence than the traditional learning techniques in online tuning of T1FNNs. There are various studies in the literature that aim to use the robustness property of SMC in the learning process of ANNs and T1FNNs [11]. Conversely, the robustness and the stability properties of soft computing-based control strategies can also be analyzed through the use of SMC theory [75].

A SMC theory-based learning algorithm is proposed to train T2FNNs using both triangular and Gaussian type-2 membership functions (to the knowledge of the authors, this is the first time such an approach has been used). The proposed learning algorithm establishes a sliding motion in terms of the T2FNN parameters, leading the learning error toward zero. The convergence of the algorithm is established and the stability conditions are given. As compared to the gradient-based learning methods which aim to minimize an error function, the learning parameters are tuned by the proposed algorithm in a way to enforce the error to satisfy a stable equation.

In this study, a T2FNN is assumed to have two inputs, each being modeled by triangular or Gaussian type-2 membership functions. The parameter update rules are derived for such a structure, and the stability of the learning algorithm proposed is proved by using a Lyapunov function. The learning algorithm proposed is tested on the velocity control of an EHSS in the presence of flow nonlinearities and internal friction and on a Duffing oscillator system.

7.2. The Adaptive Fuzzy Neuro Control Approach

7.2.1. The FEL Control Scheme and Fuzzy Neural Network Structure

The control scheme proposed is presented on Figure 7.1 where the FNN block with two inputs and one output can be a T1FNN or T2FNN. Its structure implements a TSK fuzzy model as presented on Figure 7.2 for the T2FNN case. 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 [76].

When a PD controller is used in the block diagram shown on Figure 7.1, it 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 PD control law is described as follows:

$$\tau_c = k_P e + k_D \dot{e} \tag{7.1}$$

where $e = \theta_d - \theta$ is the feedback error, θ_d is the target value, k_P and k_D are the controller gains and η_1 is the noise added to the output of the system.

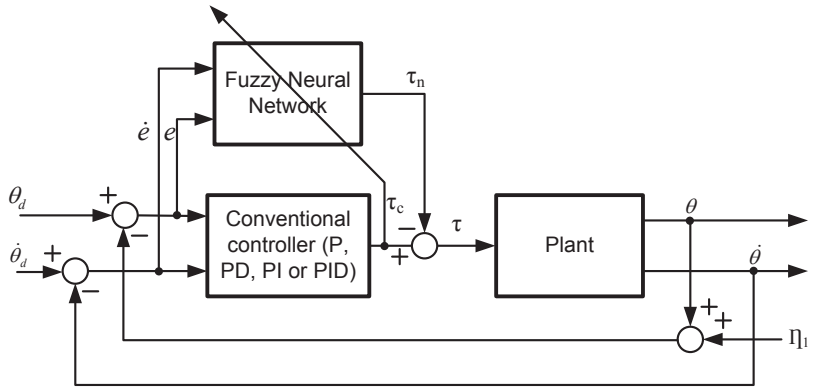


Figure 7.1. Block diagram of the proposed adaptive fuzzy neuro scheme.

7.2.2. Type-2 Fuzzy Neural Network

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 is defined as follows:

$$R_{ij}: \quad \text{If } x_1 \text{ is } \widetilde{1i} \text{ and } x_2 \text{ is } \widetilde{2j}, \quad \text{then } f_{ij} = d_{ij} \tag{7.2}$$

where x_1 and x_2 are the inputs of the type-2 TSK model, $\widetilde{1i}$ and $\widetilde{2j}$ are the type-2 fuzzy sets corresponding to the input 1 and input 2, respectively. The zeroth order function, f_{ij} is the consequent part of the rules where I $(i = 1, \ldots, I)$ and J $(j = 1, \ldots, J)$ being the number of membership functions used for the input 1 and input 2, respectively.

The T2FNN considered in this thesis uses type-2 membership functions in the premise part and crisp numbers in the consequent part (Figure 7.2). This structure is called "A2-C0" fuzzy system [42]. It is to be noted that the number of rules are equal to $K = IxJ$ in Figure 7.2.

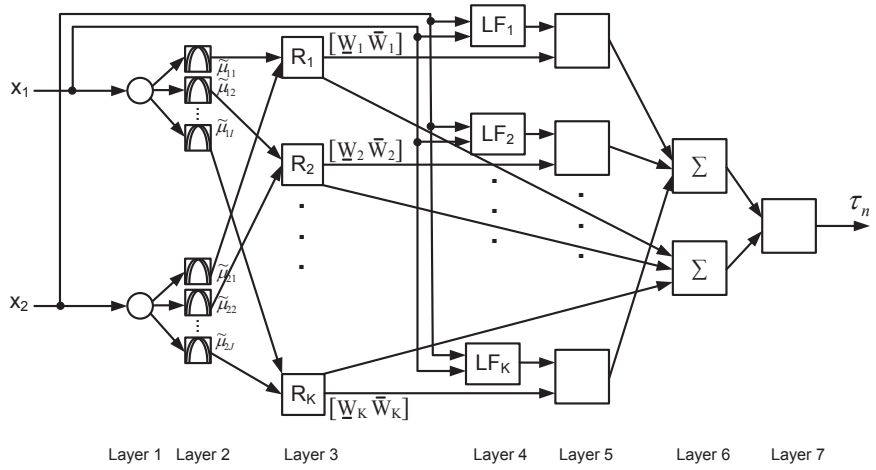


Figure 7.2. Structure of T2FNN for two inputs.

In the first layer of Figure 7.2, the input signals are fed into the system. In 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.

$$\underline{W_{ij}} = \underline{\mu}_{1i}(x_1)\underline{\mu}_{2j}(x_2) \tag{7.3}$$

$$\overline{W_{ij}} = \overline{\mu}_{1i}(x_1)\overline{\mu}_{2j}(x_2) \tag{7.4}$$

The fourth layer determines the outputs of the linear functions f_{ij}, in the consequent parts. In this study, as has been stated earlier, a zeroth-order system is assumed and therefore:

$$f_{ij} = d_{ij} \tag{7.5}$$

The fifth, the sixth and the seventh layers perform the type-reduction and the defuzzification operations. The defuzzified output of the type-2 TSK fuzzy neural system is determined as follows:

$$\tau_n = \int_{W_{11}\in[\underline{W}_{11},\overline{W}_{11}]} \cdots \int_{W_{IJ}\in[\underline{W}_{IJ},\overline{W}_{IJ}]} 1 \Big/ \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}(x) f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}(x)} \tag{7.6}$$

where f_{ij} is given by the *if-then* rule. Equation 7.6 above is the general expression to obtain the output of an interval type-2 TSK model. In order to obtain a crisp output for a practical implementation, we use the inference engine proposed in [42]. By the use of this approach, Equation 7.6 simplifies to Equation 7.7 :

$$\tau_n = \frac{q\sum_{i=1}^{M}\sum_{j=1}^{N}\underline{W}_{ij} f_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\underline{W}_{ij}} + \frac{(1-q)\sum_{i=1}^{M}\sum_{j=1}^{N}\overline{W}_{ij} f_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\overline{W}_{ij}} \tag{7.7}$$

The design parameter, q, weights the sharing of the lower and the upper firing

levels of each fired rule [42]. $\overline{W}_{ij}$ and $\underline{W}_{ij}$ are determined using Equation 7.3 and Equation 7.4, and f_{ij} is determined using Equation 7.5.

In Figure 7.2, 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 numerator part of Equation 7.7) and the other block computes the sum of the output signal of Layer 3 (the denominator part of Equation 7.7). Layer 7 calculates output of the network using Equation 7.7.

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.

Gaussian type-2 fuzzy membership functions $\underline{\mu_{1i}(x_1)}$, $\overline{\mu_{1i}(x_1)}$, $\underline{\mu_{2j}(x_2)}$, and $\overline{\mu_{2j}(x_2)}$ for the inputs x_1 and x_2 are shown for uncertain standard deviation and uncertain mean. Their mathematical expressions are as follows:

$$\underline{\mu_{1i}(x_1)} = \exp\left[-\left(\frac{x_1 - \underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)^2\right] \tag{7.8}$$

$$\overline{\mu_{1i}(x_1)} = \exp\left[-\left(\frac{x_1 - \overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)^2\right] \tag{7.9}$$

$$\underline{\mu_{2j}(x_2)} = \exp\left[-\left(\frac{x_2 - \underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)^2\right] \tag{7.10}$$

$$\overline{\mu_{2j}(x_2)} = \exp\left[-\left(\frac{x_2 - \overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)^2\right] \tag{7.11}$$

where the real constants $\underline{c}, \overline{c}, \underline{\sigma}, \overline{\sigma} > 0$ are among the tunable parameters of the T2FNN. Hence, Equation 7.3 and Equation 7.4 can be rewritten as follows:

$$\underline{W_{ij}} = \exp\left[-\left(\frac{x_1 - \underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)^2 - \left(\frac{x_2 - \underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)^2 \right] \tag{7.12}$$

$$\overline{W_{ij}} = \exp\left[-\left(\frac{x_1 - \overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)^2 - \left(\frac{x_2 - \overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)^2 \right] \tag{7.13}$$

After the normalization of Equation 7.7, the output signal of the T2FNN will acquire the following form:

$$\tau_n = q \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \widetilde{\underline{W_{ij}}} + (1-q) \sum_{i=1}^{I} \sum_{j=1}^{J} f_{ij} \widetilde{\overline{W_{ij}}} \tag{7.14}$$

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

$$\widetilde{\underline{W_{ij}}} = \frac{\underline{W_{ij}}}{\sum_{i=1}^{I} \sum_{j=1}^{J} \underline{W_{ij}}} \tag{7.15}$$

$$\widetilde{\overline{W_{ij}}} = \frac{\overline{W_{ij}}}{\sum_{i=1}^{I} \sum_{j=1}^{J} \overline{W_{ij}}} \tag{7.16}$$

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

$$\tau = \tau_c - \tau_n \tag{7.17}$$

where τ_c and τ_n are the control signals generated by the PD controller and the FNN, respectively.

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

$$|\tau(t)| < B_\tau, \quad |\dot{\tau}(t)| < B_{\dot{\tau}} \quad \forall t \tag{7.18}$$

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

7.2.3. The Sliding Mode Learning Algorithm

Using the principles of SMC theory [77], the time-varying sliding surface is defined:

$$S_c(\tau_n, \tau) = \tau_c(t) = \tau_n(t) + \tau(t) = 0 \tag{7.19}$$

This is the condition that the T2FNN is trained to become the nonlinear regulator to obtain the desired response and the contribution of the PD controller is driven to zero.

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

$$S_p(e, \dot{e}) = \dot{e} + \chi e \tag{7.20}$$

with χ being a constant determining the slope of the sliding surface. It is obvious that higher value of χ implies faster response of the system. On the other hand, the selection of this parameter is not arbitrary. The reader is encouraged to refer [78] for a detailed explanation of the selection process of χ.

Definition: 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 T2FNN parameters such that the sliding mode condition of the above definition is enforced.

7.2.3.1. The Parameter Update Rules for Gaussian Membership Functions.

The T2FLS is assumed to have two inputs, each being modeled by type-2 Gaussian membership functions. The parameter update rules are derived for such a structure.

Theorem 7.1. *If the adaptation laws for the parameters of the considered T2FNN are chosen as:*

$$\dot{\underline{c}}_{1i} = \dot{\bar{c}}_{1i} = \dot{x_1} \tag{7.21}$$

$$\dot{\underline{c}}_{2j} = \dot{\bar{c}}_{2j} = \dot{x_2} \tag{7.22}$$

$$\dot{\underline{\sigma_{1i}}} = -\frac{(\underline{\sigma_{1i}})^3}{(x_1 - \underline{c_{1i}})^2}\alpha sgn(\tau_c) \tag{7.23}$$

$$\dot{\underline{\sigma_{2j}}} = -\frac{(\underline{\sigma_{2j}})^3}{(x_2 - \underline{c_{2j}})^2}\alpha sgn(\tau_c) \tag{7.24}$$

$$\dot{\overline{\sigma_{1i}}} = -\frac{(\bar{\sigma_{1i}})^3}{(x_1 - \bar{c_{1i}})^2}\alpha sgn(\tau_c) \tag{7.25}$$

$$\dot{\overline{\sigma_{2j}}} = -\frac{(\bar{\sigma_{2j}})^3}{(x_2 - \bar{c_{2j}})^2}\alpha sgn(\tau_c) \tag{7.26}$$

$$\dot{d_{ij}} = -\frac{q\widetilde{\underline{W_{ij}}} + (1-q)\widetilde{\overline{W_{ij}}}}{(q\underline{\widetilde{W}} + (1-q)\overline{\widetilde{W}})^T(q\underline{\widetilde{W}} + (1-q)\overline{\widetilde{W}})}\alpha sgn(\tau_c) \tag{7.27}$$

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

$$\alpha > B_{\dot{\tau}} \tag{7.28}$$

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

Proof. The reader is referred to Appendix A. □

In Equation 7.27, $\underline{\widetilde{W}}$ and $\overline{\widetilde{W}}$ are the vectors defined as follows:

$\underline{\widetilde{W}}(t) = \left[\underline{\widetilde{W_{11}}}(t)\ \underline{\widetilde{W_{12}}}(t) ... \underline{\widetilde{W_{21}}}(t)\ ...\ \underline{\widetilde{W_{ij}}}(t)\ ...\ \underline{\widetilde{W_{IJ}}}(t)\right]^T$: vector of the normalized lower output signals of the neurons from the second hidden layer;

$\overline{\widetilde{W}}(t) = \left[\overline{\widetilde{W_{11}}}(t)\ \overline{\widetilde{W_{12}}}(t) ... \overline{\widetilde{W_{21}}}(t)\ ...\ \overline{\widetilde{W_{ij}}}(t)\ ...\ \overline{\widetilde{W_{IJ}}}(t)\right]^T$: vector of the normalized upper output signals of the neurons from the second hidden layer;

In the adaptation laws of Equations 7.21-7.27, 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 [79]:

(i) The use of a saturation function to replace the signum function.
(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 the second method is used, 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 7.29 has been used in this investigation instead of the sign function in the dynamic strategy

described in Equations 7.23-7.27.

$$\text{sign}(\tau_c) := \frac{\tau_c}{|\tau_c| + \delta} \tag{7.29}$$

where $\delta = 0.05$.

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

$$S_c = \tau_c = k_D\dot{e} + k_P e = k_D\left(\dot{e} + \frac{k_p}{k_D}e\right) = k_D S_p \tag{7.30}$$

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

$$V_p = \frac{1}{2}S_p^2 \tag{7.31}$$

Theorem 7.2. *If the adaptation strategy for the adjustable parameters of the T2FNN is chosen as in (7.21)-(7.27), then the negative definiteness of the time derivative of the Lyapunov function in Equation 7.31 is ensured.*

Proof. The reader is referred to Appendix B. □

7.2.3.2. The Parameter Update Rules for Triangular Membership Functions.

The parameter update rules for a T2FNN which uses triangular membership functions in the antecedent part and has two inputs are given in the following theorem.

Theorem 7.3. *If the adaptation laws for the parameters of the considered T2FNN are chosen as:*

$$\underline{\dot{c}_{1i}} = \begin{cases} \underline{\dot{c}_{1i}} = \dot{x}_1 & \left|x_1 - \underline{c_{1i}}\right| < \underline{d_{1i}} \\ 0 & otherwise \end{cases} \tag{7.32}$$

$$\dot{\overline{c_{1i}}} = \begin{cases} \dot{\overline{c_{1i}}} = \dot{x}_1 & |x_1 - \overline{c_{1i}}| < \overline{d_{1i}} \\ 0 & otherwise \end{cases} \tag{7.33}$$

$$\dot{\underline{c_{2j}}} = \begin{cases} \dot{\underline{c_{2j}}} = \dot{x}_2 & \left|x_2 - \underline{c_{2j}}\right| < \underline{d_{2j}} \\ 0 & otherwise \end{cases} \tag{7.34}$$

$$\dot{\overline{c_{2j}}} = \begin{cases} \dot{\overline{c_{2j}}} = \dot{x}_2 & |x_2 - \overline{c_{2j}}| < \overline{d_{2j}} \\ 0 & otherwise \end{cases} \tag{7.35}$$

$$\dot{\underline{d_{1i}}} = \underline{\mu_{1i}} \frac{-\alpha \underline{d_{1i}}^2}{x_1 - \underline{c_{1i}}} sgn(\tau_c) sgn\left(\frac{x_1 - \underline{c_{1i}}}{\underline{d_{1i}}}\right) \tag{7.36}$$

$$\dot{\overline{d_{1i}}} = \overline{\mu_{1i}} \frac{-\alpha \overline{d_{1i}}^2}{x_1 - \overline{c_{1i}}} sgn(\tau_c) sgn\left(\frac{x_1 - \overline{c_{1i}}}{\overline{d_{1i}}}\right) \tag{7.37}$$

$$\dot{\underline{d_{2j}}} = \underline{\mu_{2j}} \frac{-\alpha \underline{d_{2j}}^2}{x_2 - \underline{c_{2j}}} sgn(\tau_c) sgn\left(\frac{x_2 - \underline{c_{2j}}}{\underline{d_{2j}}}\right) \tag{7.38}$$

$$\dot{\overline{d_{2j}}} = \overline{\mu_{2j}} \frac{-\alpha \overline{d_{2j}}^2}{x_2 - \overline{c_{2j}}} sgn(\tau_c) sgn\left(\frac{x_2 - \overline{c_{2j}}}{\overline{d_{2j}}}\right) \tag{7.39}$$

$$\dot{f}_{ij} = -\frac{(q\widetilde{\underline{W_{ij}}} + (1-q)\widetilde{\overline{W_{ij}}})}{(q\widetilde{\underline{W}} + (1-q)\widetilde{\overline{W}})^T (q\widetilde{\underline{W}} + (1-q)\widetilde{\overline{W}})} \alpha sgn(\tau_c) \tag{7.40}$$

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

$$\alpha > B_{\dot{\tau}} \tag{7.41}$$

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

Proof. The reader is referred to Appendix C. □

Similar to the previous case, in order to avoid division to zero in the adaptive laws Equations 7.32-7.40 an algorithm is written to make the denominator equal to 0.001 always when its calculated value is smaller than this threshold.

In order to reduce the chattering effect the function in Equation 7.42 has been used in this investigation instead of the signum function in the dynamic strategy described in Equation 7.36-Equation 7.40.

$$\text{sign}(\tau_c) := \frac{\tau_c}{|\tau_c| + \delta} \tag{7.42}$$

where $\delta = 0.05$.

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

$$S_c = \tau_c = k_D\dot{e} + k_P e = k_D\left(\dot{e} + \frac{k_p}{k_D}e\right) = k_D S_p \tag{7.43}$$

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

$$V_p = \frac{1}{2}S_p^2 \tag{7.44}$$

Theorem 7.4. *If the adaptation strategy for the adjustable parameters of the T2FNN is chosen as in Equations 7.32-7.40, then the negative definiteness of the time derivative of the Lyapunov function in Equation 7.44 is ensured.*

Proof. The reader is referred to Appendix D. □

7.3. Simulation Studies With Gaussian Membership Functions for Duffing Oscillator

The chaotic dynamic systems are nonlinear deterministic systems which have sensitive dependence on initial conditions, they exhibit unpredictable, noisy-like and complex behaviors [80]. Since they have complex system dynamics, they are useful for being a benchmark problem for control engineering applications [81, 82]. The performance of the proposed learning algorithms is tested on the control of a Duffing oscillator which is a second-order chaotic system, described by the following differential equation:

$$\ddot{\theta} = f(\theta) + \tau \tag{7.45}$$

where $f(\theta) = -p_1\theta - p_2\theta^3 - p\dot{\theta} + qcos(\omega_d t)$, t is the time variable, τ is the control force, and p, p_1, p_2 and q are real constants. The numerical values (p=0.2, p_1=-1, p_2=1, q=0.3 and w_d=1) are selected such that the system in Equation 7.45 is forced to exhibit a chaotic behavior. The sampling time for all the experiments is set to to 1ms, and the learning rate α is set to 0.3. Figure 7.3 shows the open-loop phase portrait of the system for the numerical values given above in which the control input $\tau(t) = 0$, and the initial condition of the system is $(0, 1)$.

When the PD controller acts on its own, the best control performance obtainable is as shown in Figure 7.4. Here the desired trajectory is $\ddot{\theta}_d = 1.25sin(0.75\theta_d)$ with $\theta_d(0) = 1$ and $\dot{\theta}_d(0) = 0$) and the PD controller coefficients are tuned to $k_P = 750$ and $k_D = 275$ with trial-and-error method.

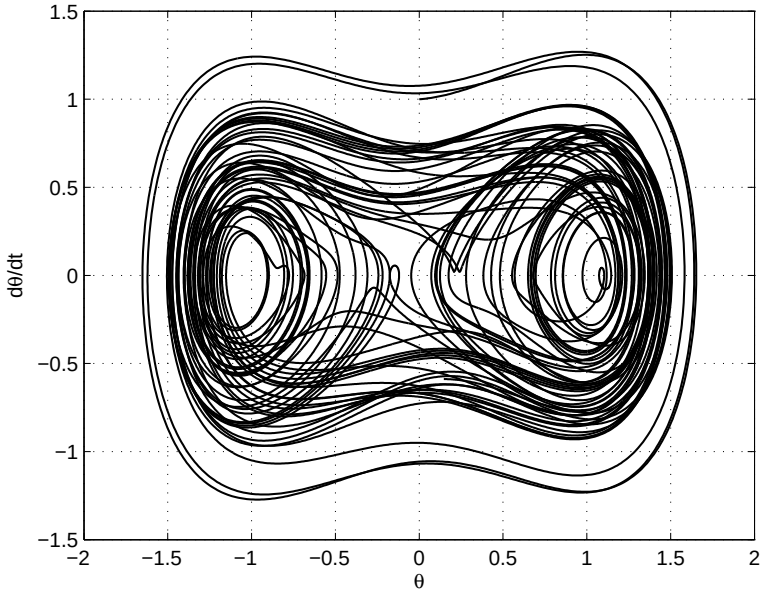


Figure 7.3. Open-loop phase portrait of Duffing oscillator.

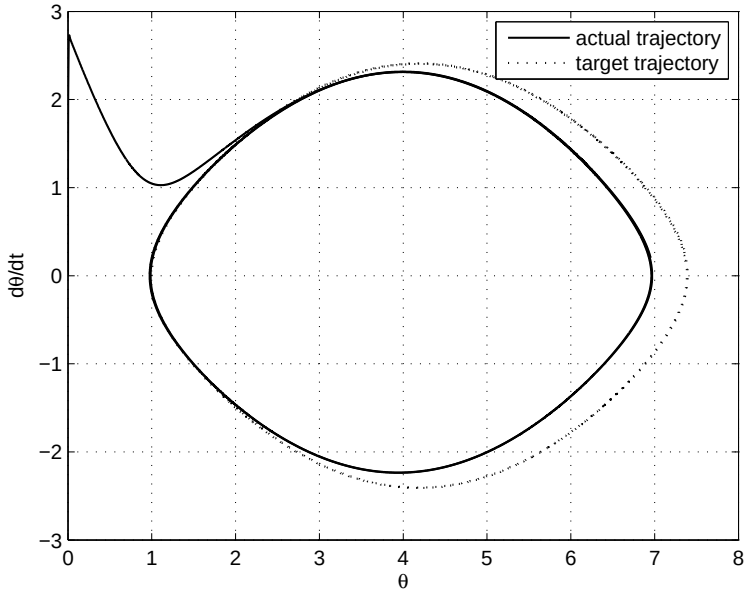


Figure 7.4. Duffing oscillator with only the conventional PD controller.

The T2FLSs should be used when they are needed, because a number of studies in the literature state that the performance of T1FLSs and T2FLSs are more or less the same when there are no uncertainties in the system to be controlled. In other words, in order to see some performance improvement between type-1 and type-2 FLCs, the system to be controlled should have some parameter uncertainties, noise, time-varying parameters or other sources of uncertainties. Figure 7.5-7.7 show the phase portrait, the output ($\theta(t)$) and the time derivative of the output ($d/dt\theta(t)$) of the actual and the target trajectories of the system respectively, with the conventional PD controller working in parallel with a T2FNN. In order to test the proposed control scheme in a noisy environment, ten experiments are done by using different levels of noise (from 10dB to 80dB), but only the one with SNR=80dB is shown in these figures. It can be seen that the proposed control scheme in which the conventional PD controller works in parallel with a T2FNN is able to enforce the system to follow the target trajectory with a better performance than the case when the PD controller acts alone.

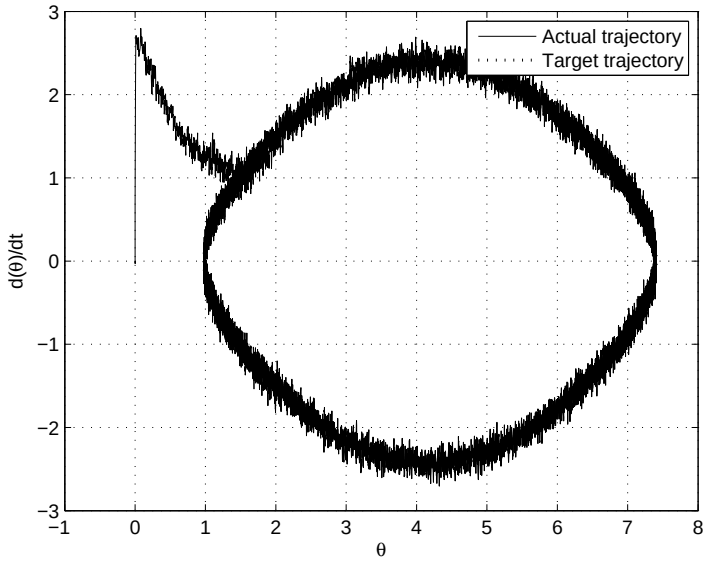


Figure 7.5. The phase portrait of the actual and the target trajectories when the PD controller acts together with T2FNN (η_1: SNR=80dB).

Figure 7.8-7.9 show the error of the system and the phase portrait of the error respectively. In Figure 7.9, the reaching of the sliding surface and the sliding mode thereafter can clearly be seen.

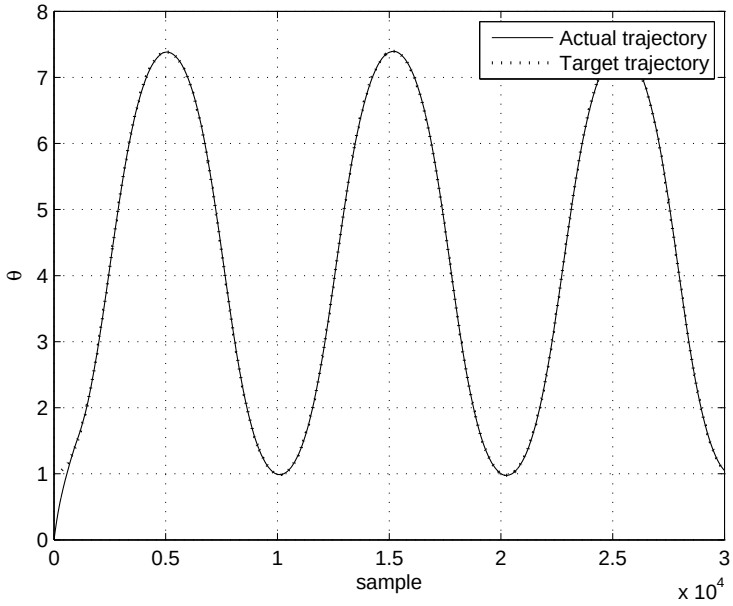


Figure 7.6. The actual and the target trajectories when the PD controller acts together with T2FNN (η_1: SNR=90dB).

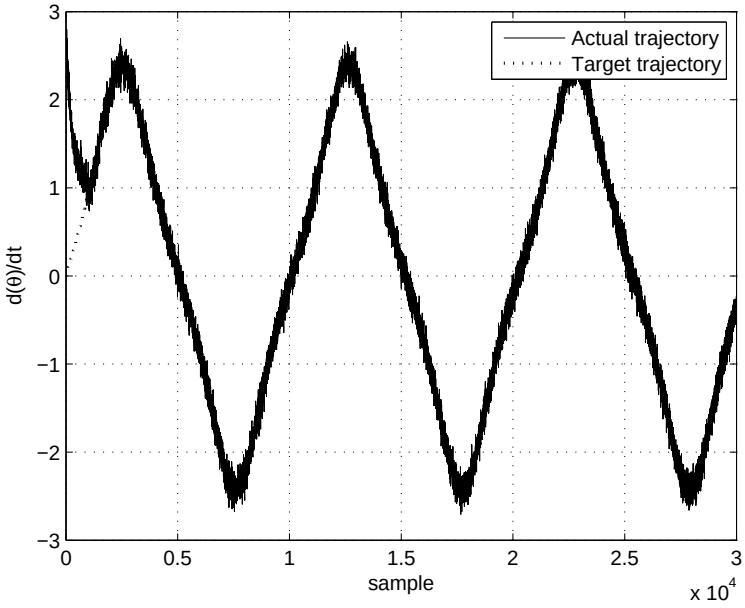


Figure 7.7. The time derivative of the actual and the target trajectories when the PD controller acts together with T2FNN (η_1: SNR=80dB).

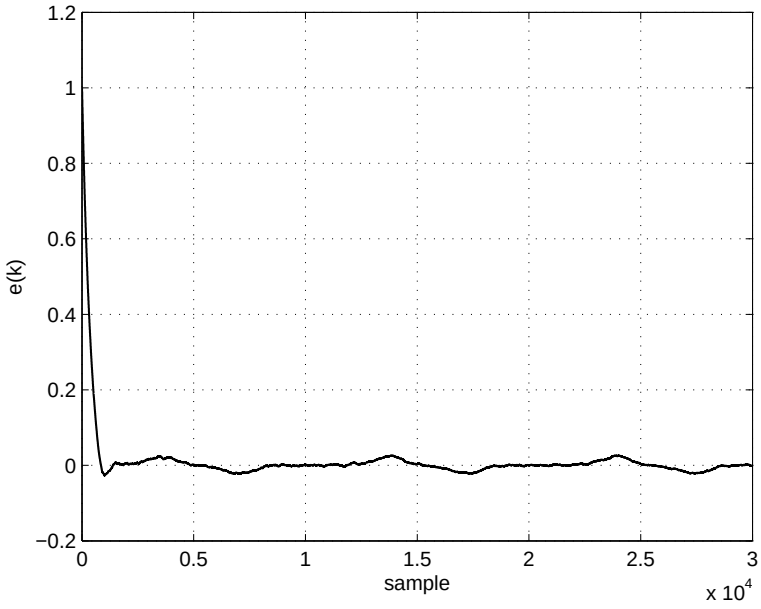


Figure 7.8. The error of the system when the PD controller acts in parallel with T2FNN (η_1: SNR=80dB).

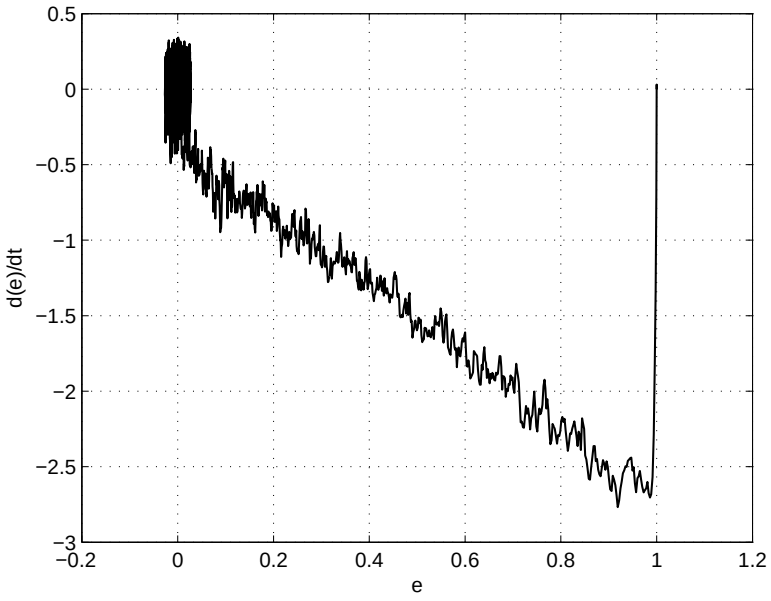


Figure 7.9. The phase portrait of the system when the PD controller acts in parallel with T2FNN (η_1: SNR=80dB).

The control signals originating from the conventional PD controller and the adaptive T2FNN can be seen in Figure 7.10. It can be observed that the T2FNN is able to take over the control operation, thus becoming the leading controller after a short time period. The output of the PD controller thereafter is to compensate for the noise and its mean value is zero.

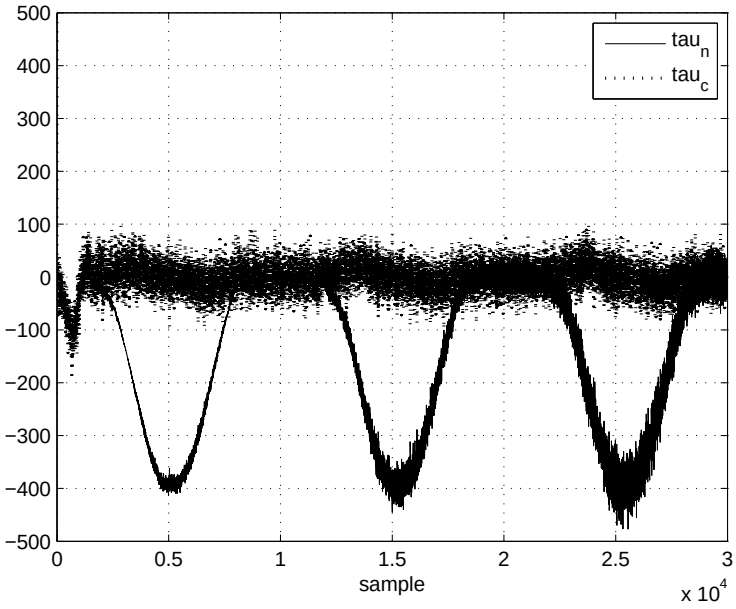


Figure 7.10. The control signals of PD and T2FNN blocks (η_1: SNR=80dB).

Finally, Table 7.1 shows the performance difference between T2FNN and T1FNN. It is seen that the benefit of the the use of T2FNN is more apparent as more noise power is injected to the system, which is an expected result. It is again stressed that the results shown are the averages of ten experiments.

7.4. Simulation Studies With Triangular Membership Functions for EHSS

EHSSs are widely used in industrial applications because of their ability to handle large inertia and torque loads and, at the same time, achieve fast starting, stopping, and reversal with smoothness and a high degree of both accuracy and performance [83, 84]. Typical applications include active suspension systems, control of industrial robots, and flight simulators. EHSSs contain several nonlinearities such as valve dead zones,

Table 7.1. The mean squared error (MSE) of the T1FNN and T2FNN.

Noise level (SNR)	**T1FNN**	**T2FNN**
10	760.33	738.93
20	161.67	142.15
30	26.27	20.40
40	2.286	2.052
50	0.284	0.250
60	0.043	0.061
70	0.0196	0.017
80	0.0135	0.015
90	0.0098	0.006

valve flow saturation, and cylinder seal friction. These nonlinearities limit the control performance of conventional controllers. This part of the dissertation investigates the control of a velocity EHSS whose mathematical model includes flow nonlinearities and internal friction. The control scheme with the proposed learning algorithm has the ability to control such a nonlinear system without having its accurate dynamical model.

The block diagram of the EHSS considered in this subsection is shown in Figure 7.11 [85]. The basic parts of the system are: 1. hydraulic power supply, 2. accumulator, 3. charge valve, 4. pressure gauge device, 5. filter, 6. two-stage electro hydraulic servo valve, 7. hydraulic motor, 8. measurement device, 9. personal computer, and 10. voltage-to-current converter [85]. The states of the system are x_1: hydro motor angular velocity [rad/sec], x_2: load pressure differential [Pa], and x_3: valve displacement [m]. The dynamical state space equations of the system are as follows:

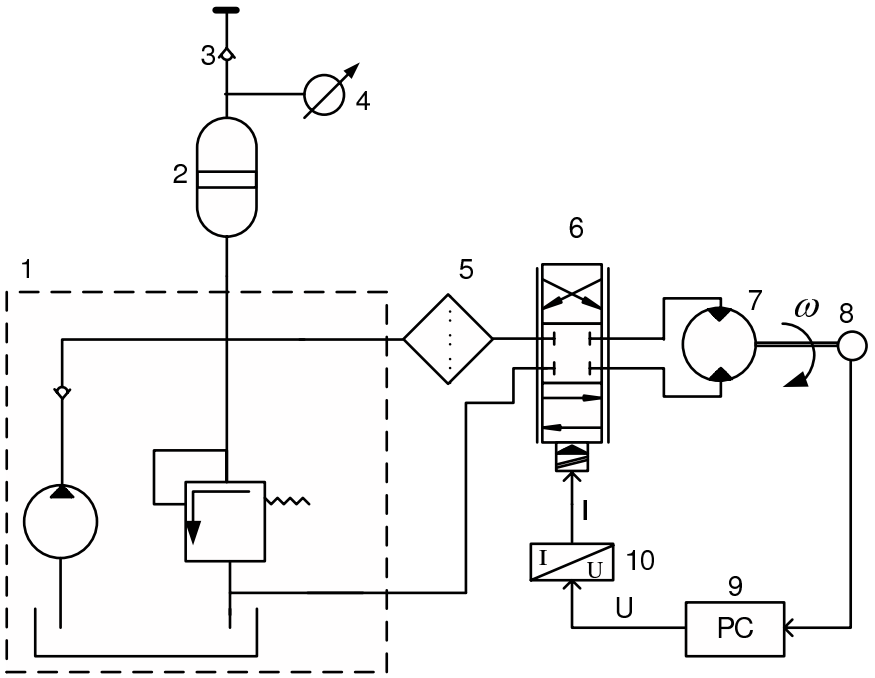


Figure 7.11. Electro Hydraulic Servo System (EHSS).

$$
\begin{aligned}
\dot{x}_1 &= \frac{1}{J_t}\big(- B_m x_1 + q_m x_2 - q_m C_f P_s sgn(x_1)\big) \qquad (7.46)\\
\dot{x}_2 &= \frac{2\beta_e}{V_0}\Bigg(- q_m x_1 - C_{im} x_2 + C_d W x_3 \sqrt{\frac{1}{\rho}\big(P_s - x_2 sgn(x_3)\big)}\Bigg)\\
\dot{x}_3 &= \frac{1}{T_r}(-x_3 + \frac{K_r}{K_q} u)\\
y &= x_1
\end{aligned}
$$

It is assumed that the motor shaft does not change its direction of rotation, $x_1 > 0$. This is a practical assumption and in order for it to be satisfied, the servo valve displacement x_3 does not have to move in both directions relative to the neutral position $x_3 = 0$. This fact allows us to restrict the entire problem to the region where $x_3 > 0$. In this case, the mathematical representation of the system simplifies to

Equation 7.47 [85]:

$$\begin{aligned}
\dot{x}_1 &= \frac{1}{J_t}\big(-B_m x_1 + q_m x_2 - q_m C_f P_s\big) \\
\dot{x}_2 &= \frac{2\beta_e}{V_0}\left(-q_m x_1 - C_{im} x_2 + C_d W x_3 \sqrt{\frac{1}{\rho}\big(P_s - x_2\big)}\right) \\
\dot{x}_3 &= \frac{1}{T_r}(-x_3 + \frac{K_r}{K_q} u) \\
y &= x_1
\end{aligned} \tag{7.47}$$

The state space representation of the system is as follows:

$$\begin{aligned}
\dot{x} &= f(x) + Bu \\
y &= Cx + Du
\end{aligned} \tag{7.48}$$

where

$$f(x) = \begin{pmatrix} \frac{1}{J_t}\big(-B_m x_1 + q_m x_2 - q_m C_f P_s\big) \\ \frac{2\beta_e}{V_0}\left(-q_m x_1 - C_{im} x_2 + C_d W x_3 \sqrt{\frac{1}{\rho}\big(P_s - x_2\big)}\right) \\ \frac{1}{-T_r} x_3 \end{pmatrix},\ B = \begin{pmatrix} 0 \\ 0 \\ \frac{K_r}{K_q} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix},\ D = 0$$

The nomenclature of the symbols and the numerical values used in this study are given in Table 7.2.

A number of simulations have been done to show the effectiveness of the proposed control algorithm. The sampling period for all the simulations is selected as $0.2ms$. The implemented control structure consists of a conventional controller (in this case a PD controller) and a T2FLS feedback controller. The former is provided to guarantee global asymptotic stability in compact space as well as to act as an inverse reference model

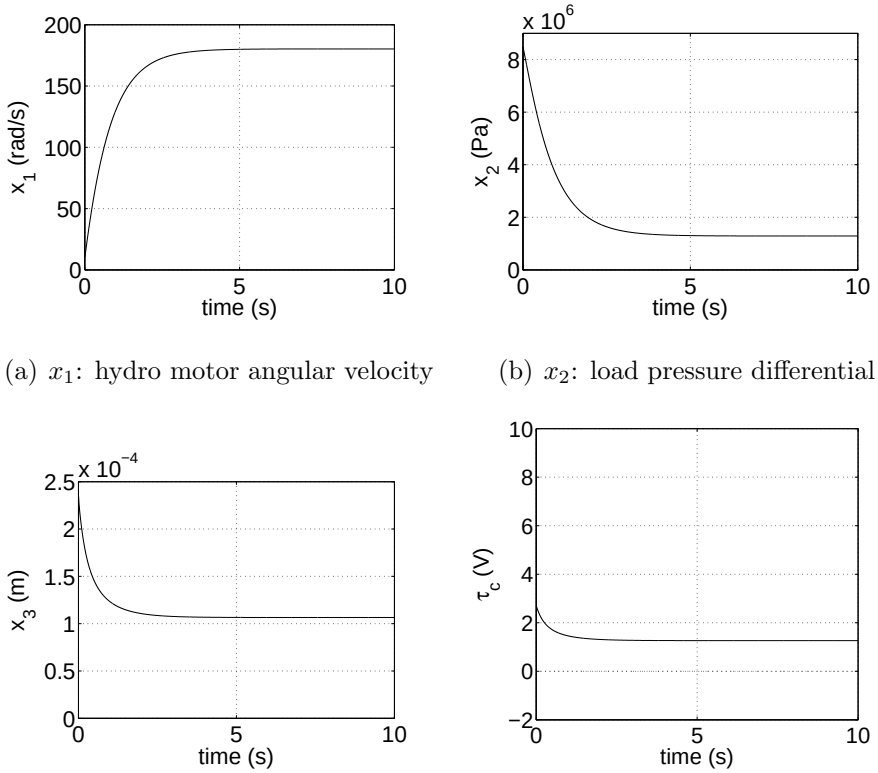
Table 7.2. Nomenclature.

Symbol	**Description**	**Numerical value**
K_r	Valve gain	$1.4x10^{-4}m^3/s.V$
ρ	Oil density	$850kg/m^3$
K_q	Valve flow gain	$1.66m^2/s$
P_s	Supply pressure	10^7Pa
W	Surface gradient	$8\Pi x10^{-3}m$
T_r	Valve time constant	$0.01s$
C_d	Discharge coefficient	0.61
B_m	Viscous damping coefficient	$1.1x10^{-3}Nms$
β_e	Effective hulk modulus of the system	$1.391x10^9Pa$
q_m	Volumetric displacement of the motor	$7.96x10^{-7}m^3/rad$
C_f	Dimensionless internal friction coefficient	0.104
V_0	Average contained volume of each motor chamber	$1.2x10^{-4}m^3$
C_{im}	Internal or cross-port leakage coefficient of the motor	$1.69x10^{-11}m^3/Pa.s$
J_t	Total inertia of the motor and load referred to the motor shaft	$0.03kgm^2$

of the response of the controlled system. Its output is used as an error signal by an incremental learning algorithm to update the parameters of the fuzzy neuro controller. In this way the latter is able to gradually replace the conventional controller.

The tracking performance of a PD controller working alone is shown in Figure 7.12. As can be seen from Figure 7.12a, the PD controller cannot eliminate the steady state error for a step input which is equal to 200rad/s for x_1.

The tracking performance of the T2FNN working in parallel with a PD controller is shown in Figure 7.13. As can be seen from 7.13a, when the T2FNN is used in parallel with a PD controller, the system does not have any steady state error.

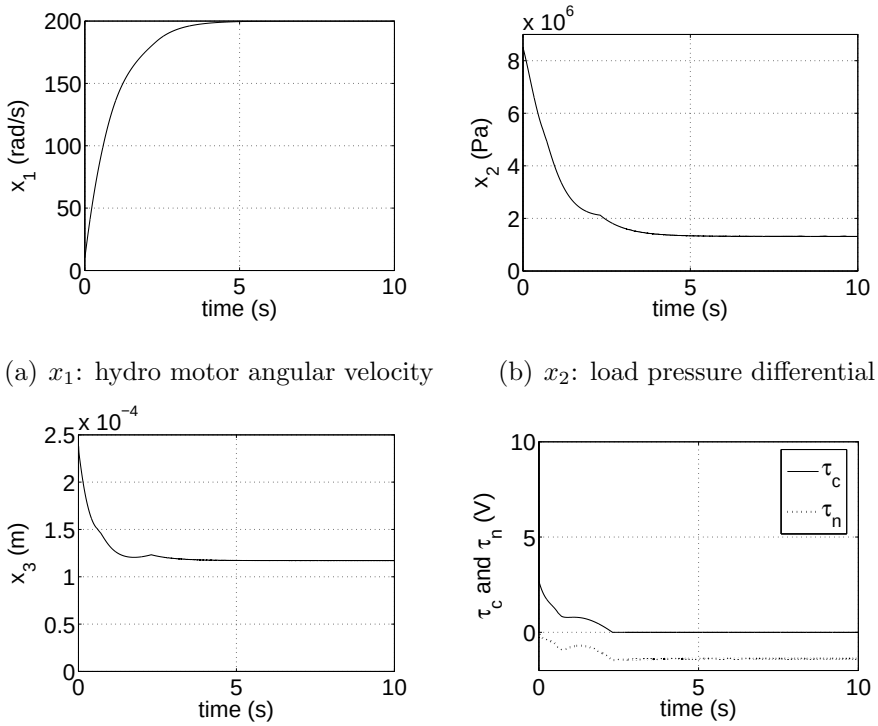


(a) x_1: hydro motor angular velocity (b) x_2: load pressure differential

(c) x_3: valve displacement (d) The output of the PD controller

Figure 7.12. Simulation results for the case of PD controller working alone when reference signal is equal to 200rad/s for x_1.

Figure 7.12(d) shows the signal τ_c when the PD controller is used alone. Figure 7.13(d) shows the control signals τ_c and τ_n when the PD controller is working in parallel with the proposed T2FNN. As can be seen from Figure 7.13(d), when PD controller is used in parallel with the proposed T2FNN, the signal τ_c goes to zero after a finite time. And then, T2FNN takes the responsibility of controlling the process. Figure 7.14 indicates the control accuracy difference between the T1FLCs and the T2FLCs with the proposed learning algorithm. It can be seen that the performance accuracy of T2FLS is higher than its type-1 counterpart in the presence of the higher levels noise. A similar conclusion has previously be reached in [38]. Consequently, it can be stated that T2FLSs should be preferred when there exists a large amount of uncertainties in the system. In other words, if the uncertainties in the system are relatively small, T1FLS might be a better choice because of having less number of parameters to be tuned.



(a) x_1: hydro motor angular velocity

(b) x_2: load pressure differential

(c) x_3: valve displacement

(d) The output of the PD controller alone and PD+T2FLS

Figure 7.13. Simulation results for the case of PD controller working in parallel with T2FLS when reference signal is equal to 200rad/s for x_1.

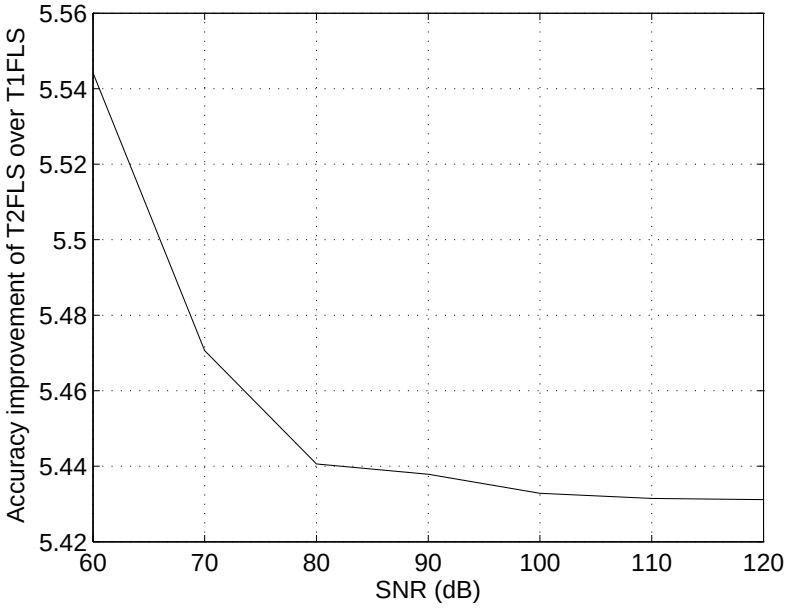


Figure 7.14. The improvement of the control accuracy of T2FLS over T1FLS.

Figure 7.15 shows the phase space behavior of the error both for the cases when the conventional PD controller is used alone and when it is used in parallel with the proposed T2FNN. This figure indicates that the error dynamics can reach the sliding line faster and remains there when the conventional PD controller is used in parallel with the proposed T2FNN.

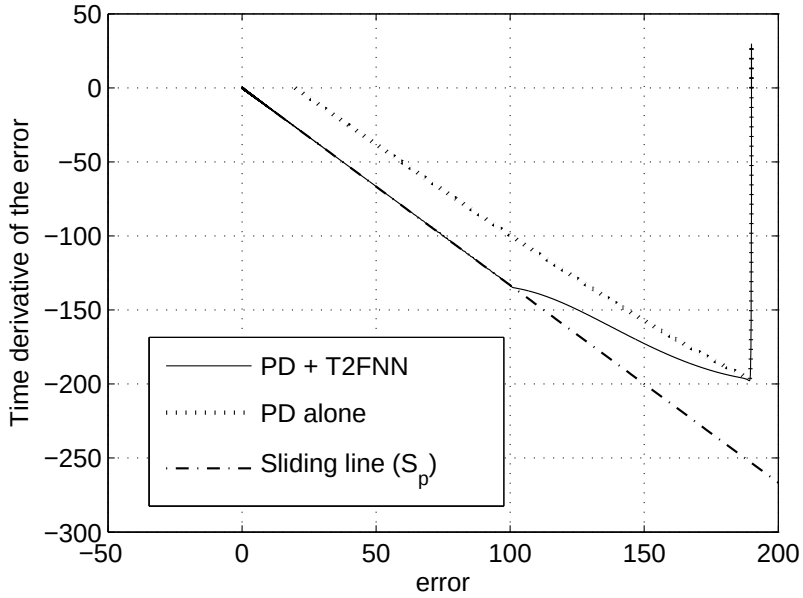


Figure 7.15. The phase portrait of the system.

7.5. Conclusion

The highly novel aspect of this section is the use of a SMC theory-based learning algorithm to train the parameters of T2FNNs. The performance of the proposed learning algorithm is tested through simulation studies on a Duffing oscillator and EHSS. It is seen that the proposed structure that uses a combination of type-2 FL theory, ANNs, and SMC theory harmoniously allows us to better handle modeling uncertainties in terms of noise, as compared to a T1FNN structure. The prominent feature of the proposed algorithm is its computational simplicity as compared with gradient based and genetic algorithms. Its real time implementation is therefore relatively simple.

8. CONCLUSIONS

To our best knowledge, there does not exist any paper in literature which makes a general comment about the noise reduction property of T2FLSs. We propose a novel type-2 membership function in this study that enables us to come up with some metrics. This membership function has certain values on both ends of the support and the kernel, and some uncertain values on other points of the support. With such a function, the parameters responsible for the width of uncertainty are de-coupled from the parameters responsible for the center and the support of the membership function. This allows us to analyze how the uncertainty in the input distorts the inference of the T2FLS. The parameter update rules are derived for the proposed type-2 fuzzy membership function using the GD method. Three different applications (identification, prediction and control) are considered and the simulation results obtained indicate that that the width of the proposed membership function increases as more noise power is injected to the system. The performance improvement of T2FLS over type-1 counterparts is much higher for the higher levels of noise. This fact shows that T2FLSs should be used when needed, i.e. in the presence of noise and uncertainties in the system. Another contribution of this dissertation is that some guidelines are derived for the selection of the parameters a_1 and a_2 of the novel membership function.

In this dissertation a T2FNN is proposed for the identification and control of a servo system with time-varying and nonlinear load conditions. The structure of T2FNN is presented and the parameter update rules of the structure are derived based on fuzzy clustering and GD algorithms. In order to improve the performance of the type-2 fuzzy structure, a fuzzy c-means clustering algorithm is used to determine the initial positions of the membership functions. The experimental results obtained indicate the potential of the proposed T2FNN structure. It is seen that type-2 fuzzy control algorithm can handle the uncertainties in the system effectively with a much better transient performance and smaller RMSE values compared to a conventional T1FNN.

The parameter update rules of a T2FLS with a novel membership function are derived using LM algorithm. In order to show the effectiveness of the use of LM for training the parameters of T2FLSs, a gradient based training method is also used. The results show that LM outperforms the gradient-based method, while its computation time is less than the computational time of GD algorithm. T2FLS trained by LM and GD methods are used for the prediction of Mackey-Glass chaotic system in the presence of measurement noise. It is shown that the performance of T2FLS is better than the performance of type-1 counterpart in the presence of higher levels of noise and uncertainties. The simulation results also show that the width of the proposed membership function increases as more noise power is injected to the system which means that the performance improvement of T2FLS over type-1 counterpart is much higher for the higher levels of noise. This fact suggests that T2FLSs should be preferred in the presence of higher levels of uniform noise and uncertainties in the system.

In this dissertation, a SMC theory-based learning algorithm is proposed to train T2FNNs for both Gaussian and triangular type-2 fuzzy membership functions (to the knowledge of the authors, this is the first time such an approach has been used). The parameter update rules are derived for a structure with two inputs, each being modeled by type-2 membership functions. It is seen through the simulations studies carried out that the T2FNN with the sliding mode learning algorithm can control the systems considered in this dissertation 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.

APPENDIX A: PROOF OF THEOREM 1

Time derivatives of Equations 7.8-7.11 are written as follows:

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

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

$$\dot{\underline{\mu_{2j}(x_2)}} = -2A_{2j}(\dot{A_{2j}})\underline{\mu_{2j}(x_2)} \tag{A.3}$$

$$\dot{\overline{\mu_{2j}(x_2)}} = -2U_{2j}(\dot{U_{2j}})\overline{\mu_{2j}(x_2)} \tag{A.4}$$

where

$A_{1i} = \left(\frac{x_1 - \underline{c_{1i}}}{\underline{\sigma_{1i}}}\right)$, $A_{2j} = \left(\frac{x_2 - \underline{c_{2j}}}{\underline{\sigma_{2j}}}\right)$, $U_{1i} = \left(\frac{x_1 - \overline{c_{1i}}}{\overline{\sigma_{1i}}}\right)$, and $U_{2j} = \left(\frac{x_2 - \overline{c_{2j}}}{\overline{\sigma_{2j}}}\right)$

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:

$$\widetilde{\underline{W_{ij}}} = \frac{\underline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}} \tag{A.5}$$

$$\widetilde{\overline{W_{ij}}} = \frac{\overline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}}} \tag{A.6}$$

The time derivative of Equation A.5 and Equation A.6 are written as follows:

$$\dot{\widetilde{\underline{W_{ij}}}} = \frac{\Big(\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Big)'\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})^2} - \frac{\Big(\underline{W_{ij}}\Big)\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Big)'}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})^2} \tag{A.7}$$

$$\dot{\widetilde{\overline{W_{ij}}}} = \frac{\Big(\overline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Big)'\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}}\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}})^2} - \frac{\Big(\overline{W_{ij}}\Big)\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Big)'}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}})^2} \tag{A.8}$$

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

$$\begin{aligned}
\dot{\widetilde{\underline{W_{ij}}}} &= \frac{\Big(\underline{\dot{\mu}_{1i}(x_1)\mu_{2j}(x_2)} + \underline{\mu_{1i}(x_1)\dot{\mu}_{2j}(x_2)}\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \\
&\quad - \frac{(\widetilde{\underline{W_{ij}}})\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\underline{\dot{\mu}_{1i}(x_1)\mu_{2j}(x_2)} + \underline{\mu_{1i}(x_1)\dot{\mu}_{2j}(x_2)}\Big)\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \\
&= \frac{\Big(-2A_{1i}(A_{1i})'\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)} - 2A_{2j}(A_{2j})'\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \\
&\quad - \frac{(\widetilde{\underline{W_{ij}}})\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(-2A_{1i}(A_{1i})'\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \\
&\quad - \frac{(\widetilde{\underline{W_{ij}}})\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(2A_{2j}(A_{2j})'\underline{\mu_{1i}(x_1)\mu_{2j}(x_2)}\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})}
\end{aligned} \tag{A.9}$$

$$\dot{\widetilde{\underline{W_{ij}}}} = -\widetilde{\underline{W_{ij}}}(\dot{\underline{K_{ij}}}) + \widetilde{\underline{W_{ij}}} \sum_{i=1}^{I} \sum_{j=1}^{J} \widetilde{\underline{W_{ij}}}(\dot{\underline{K_{ij}}}) \tag{A.10}$$

$$\dot{\widetilde{\overline{W_{ij}}}} = -\widetilde{\overline{W_{ij}}}(\dot{\overline{K_{ij}}}) + \widetilde{\overline{W_{ij}}} \sum_{i=1}^{I} \sum_{j=1}^{J} \widetilde{\overline{W_{ij}}}(\dot{\overline{K_{ij}}}) \tag{A.11}$$

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;

$$V_c = \frac{1}{2}\tau_c^2(t) \tag{A.12}$$

$$\dot{V_c} = \tau_c \dot{\tau_c} = \tau_c(\dot{\tau_n} + \dot{\tau}) \tag{A.13}$$

$$\begin{aligned} \tau_n &= \frac{q\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\underline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}} + \frac{(1-q)\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\overline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}}} \\ &= q\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\widetilde{\underline{W_{ij}}} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\widetilde{\overline{W_{ij}}} \end{aligned} \tag{A.14}$$

$$\begin{aligned}\dot{\tau_n} = q\sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f_{ij}}\widetilde{\underline{W_{ij}}} + f_{ij}\dot{\widetilde{\underline{W_{ij}}}}) \\ + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f_{ij}}\widetilde{\overline{W_{ij}}} + f_{ij}\dot{\widetilde{\overline{W_{ij}}}})\end{aligned} \tag{A.15}$$

$$\begin{aligned}\dot{\tau_n} = q\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(\Big(-\widetilde{\underline{W_{ij}}}(\underline{K_{ij}})' + \widetilde{\underline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\underline{W_{ij}}}(\underline{K_{ij}})'\Big)f_{ij} + \widetilde{\underline{W_{ij}}}\dot{f_{ij}}\Bigg) \\ + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(\Big(-\widetilde{\overline{W_{ij}}}(\overline{K_{ij}})' + \widetilde{\overline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}}(\overline{K_{ij}})'\Big)f_{ij} \\ + \widetilde{\overline{W_{ij}}}\dot{f_{ij}}\Bigg)\end{aligned} \tag{A.16}$$

$$\begin{aligned}\dot{V_c} = \tau_c\Bigg(q\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(\Big(-\widetilde{\underline{W_{ij}}}(\underline{K_{ij}})' + \widetilde{\underline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\underline{W_{ij}}}(\underline{K_{ij}})'\Big)f_{ij} \\ + \widetilde{\underline{W_{ij}}}\dot{f_{ij}}\Bigg) + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(\Big(-\widetilde{\overline{W_{ij}}}(\overline{K_{ij}})' \\ + \widetilde{\overline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}}(\overline{K_{ij}})'\Big)f_{ij} + \widetilde{\overline{W_{ij}}}\dot{f_{ij}}\Bigg) + \dot{\tau}\Bigg) \\ = \tau_c\Bigg(q\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(\Big(-\widetilde{\underline{W_{ij}}}\big(2A_{1i}(A_{1i})' + 2A_{2j}(A_{2j})'\big) \\ + \widetilde{\underline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\underline{W_{ij}}}\big(2A_{1i}(A_{1i})' + 2A_{2j}(A_{2j})'\big)\Big)f_{ij} + \widetilde{\underline{W_{ij}}}\dot{f_{ij}}\Bigg) \\ + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\Bigg(\Big(-\widetilde{\overline{W_{ij}}}\big(2U_{1i}(U_{1i})' + 2U_{2j}(U_{2j})'\big) \\ + \widetilde{\overline{W_{ij}}}\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}}\big(2U_{1i}(U_{1i})' + 2U_{2j}(U_{2j})'\big)\Big)f_{ij} + \widetilde{\overline{W_{ij}}}\dot{f_{ij}}\Bigg) + \dot{\tau}\Bigg)\end{aligned} \tag{A.17}$$

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 A.18 can be obtained as follows;

$$A_{1i}\dot{A_{1i}} = A_{2j}\dot{A_{2j}} = U_{1i}\dot{U_{1i}} = U_{2j}\dot{U_{2j}} = \alpha sgn(\tau_c) \tag{A.18}$$

$$\begin{aligned}
\dot{V_c} = \tau_c \Bigg(& \sum_{i=1}^{I}\sum_{j=1}^{J} \Bigg(2\Big(- \widetilde{\underline{W_{ij}}} f_{ij} 2\alpha sgn(\tau_c) \\
& + \widetilde{\underline{W_{ij}}} f_{ij} \sum_{i=1}^{I}\sum_{j=1}^{J} \widetilde{\underline{W_{ij}}} 2\alpha sgn(\tau_c)\Big) + \widetilde{\underline{W_{ij}}}\dot{f_{ij}} \Bigg) \\
& + (1-q) \sum_{i=1}^{I}\sum_{j=1}^{J} \Bigg(2\Big(- \widetilde{\overline{W_{ij}}} f_{ij} 2\alpha sgn(\tau_c) \\
& + \widetilde{\overline{W_{ij}}} f_{ij} \sum_{i=1}^{I}\sum_{j=1}^{J} \widetilde{\overline{W_{ij}}} 2\alpha sgn(\tau_c)\Big) \widetilde{\overline{W_{ij}}}\dot{f_{ij}} \Bigg) + \dot{\tau} \Bigg) \\
= \tau_c \Bigg(& \Bigg(- 4q\alpha sgn(\tau_c) \sum_{i=1}^{I}\sum_{j=1}^{J} \Big(\widetilde{\underline{W_{ij}}} f_{ij} - \widetilde{\underline{W_{ij}}} f_{ij} \sum_{i=1}^{I}\sum_{j=1}^{J} \widetilde{\underline{W_{ij}}} \Big) \\
& + q \sum_{i=1}^{I}\sum_{j=1}^{J} \widetilde{\underline{W_{ij}}}\dot{f_{ij}} - 4(1-q)\alpha sgn(\tau_c) \sum_{i=1}^{I}\sum_{j=1}^{J} \Big(\widetilde{\overline{W_{ij}}} f_{ij} \\
& - \widetilde{\overline{W_{ij}}} f_{ij} \sum_{i=1}^{I}\sum_{j=1}^{J} \widetilde{\overline{W_{ij}}} \Big) + (1-q) \sum_{i=1}^{I}\sum_{j=1}^{J} \widetilde{\overline{W_{ij}}}\dot{f_{ij}} \Bigg) + \dot{\tau} \Bigg)
\end{aligned} \tag{A.19}$$

Since $\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}} = 1$ and $\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\underline{W_{ij}}} = 1$, the previous equation becomes as follows:

$$\begin{aligned}
\dot{V_c} &= \tau_c\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(q\widetilde{\underline{W_{ij}}}\dot{f}_{ij} + (1-q)\widetilde{\overline{W_{ij}}}\dot{f}_{ij}\Big) + \dot{\tau}\Big) \\
&= \tau_c\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f}_{ij}\Big(q\widetilde{\underline{W_{ij}}} + (1-q)\widetilde{\overline{W_{ij}}}\Big) + \dot{\tau}\Big) \\
&= \tau_c\Big(q\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\underline{W_{ij}}}\dot{f}_{ij} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}}\dot{f}_{ij} + \dot{\tau}\Big) \\
&= \tau_c\Big(-\alpha sgn(\tau_c) + \dot{\tau}\Big) < \Big(-\alpha|\tau_c| + |\tau_c|B_{\dot{\tau}}\Big) < 0
\end{aligned} \tag{A.20}$$

APPENDIX B: PROOF OF THEOREM 2

Evaluating the time derivative of the Lyapunov function in Equation 7.31 yields:

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

Remark: The obtained result means that using τ_c as a learning error for the T2FNN together with the adaptation laws Equations 7.21-7.27 enforces the desired reaching mode, followed by a sliding regime for the system under control.

APPENDIX C: PROOF OF THEOREM 3

$$\underline{\mu_{1i}(x_1)} = \begin{cases} 1 - \left|\frac{x_1 - \underline{c_{1i}}}{\underline{d_{1i}}}\right| & |x_1 - \underline{c_{1i}}| < \underline{d_{1i}} \\ 0 & otherwise \end{cases} \tag{C.1}$$

$$\overline{\mu_{1i}(x_1)} = \begin{cases} 1 - \left|\frac{x_1 - \overline{c_{1i}}}{\overline{d_{1i}}}\right| & |x_1 - \overline{c_{1i}}| < \overline{d_{1i}} \\ 0 & otherwise \end{cases} \tag{C.2}$$

$$\underline{\mu_{2j}(x_2)} = \begin{cases} 1 - \left|\frac{x_2 - \underline{c_{2j}}}{\underline{d_{2j}}}\right| & \left|x_2 - \underline{c_{2j}}\right| < \underline{d_{2j}} \\ 0 & otherwise \end{cases} \tag{C.3}$$

$$\overline{\mu_{2j}(x_2)} = \begin{cases} 1 - \left|\frac{x_2 - \overline{c_{2j}}}{\overline{d_{2j}}}\right| & |x_2 - \overline{c_{2j}}| < \overline{d_{2j}} \\ 0 & otherwise \end{cases} \tag{C.4}$$

The time derivatives of Equation C.1 and Equation C.4 are as follows:

$$\underline{\dot{\mu}_{1i}(x_1)} = \left(- \frac{\dot{x}_1 - \underline{\dot{c}_{1i}}}{\underline{d_{1i}}} + \frac{x_1 - \underline{c_{1i}}}{\underline{d_{1i}^2}} \underline{\dot{d}_{1i}} \right) sgn\left(\frac{\overbrace{x_1 - \underline{c_{1i}}}^{\underline{E_1}}}{\underline{d_{1i}}} \right) \tag{C.5}$$

$$\overline{\dot{\mu}_{1i}(x_1)} = \left(- \frac{\dot{x}_1 - \overline{\dot{c}_{1i}}}{\overline{d_{1i}}} + \frac{x_1 - \overline{c_{1i}}}{\overline{d_{1i}^2}} \overline{\dot{d}_{1i}} \right) sgn\left(\frac{\overbrace{x_1 - \overline{c_{1i}}}^{\overline{E_1}}}{\overline{d_{1i}}} \right) \tag{C.6}$$

$$\underline{\dot{\mu}_{2j}(x_2)} = \Big(- \frac{\dot{x}_2 - \underline{\dot{c}_{2j}}}{\underline{d_{2j}}} + \frac{x_2 - \underline{c_{2j}}}{\underline{d_{2j}^2}} \underline{\dot{d}_{2j}} \Big) sgn\Big(\overbrace{\frac{x_2 - \underline{c_{2j}}}{\underline{d_{2j}}}}^{\underline{E_2}} \Big) \tag{C.7}$$

$$\overline{\dot{\mu}_{2j}(x_2)} = \Big(- \frac{\dot{x}_2 - \overline{\dot{c}_{2j}}}{\overline{d_{2j}}} + \frac{x_2 - \overline{c_{2j}}}{\overline{d_{2j}^2}} \overline{\dot{d}_{2j}} \Big) sgn\Big(\overbrace{\frac{x_2 - \overline{c_{2j}}}{\overline{d_{2j}}}}^{\overline{E_2}} \Big) \tag{C.8}$$

$$\begin{aligned} \underline{\dot{\widetilde{W}_{ij}}} = & \frac{\Big(\overbrace{\underline{\mu_{1i}}\underline{\dot{(x_1)}}\underline{\mu_{2j}(x_2)} + \underline{\mu_{1i}(x_1)}\underline{\mu_{2j}\dot{(x_2)}}}^{\mathrm{A}} \Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \\ & - \frac{(\underline{\widetilde{W_{ij}}})\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\overbrace{\underline{\mu_{1i}}\underline{\dot{(x_1)}}\underline{\mu_{2j}(x_2)} + \underline{\mu_{1i}(x_1)}\underline{\mu_{2j}\dot{(x_2)}}}^{\mathrm{A}} \Big)\Big]}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \end{aligned} \tag{C.9}$$

$$\begin{aligned} A &= \Big(\Big(\frac{x_1 - \underline{c_{1i}}}{\underline{d_{1i}^2}} \Big) \underline{\dot{d}_{1i}} - \Big(\frac{\dot{x}_1 - \underline{\dot{c}_{1i}}}{\underline{\dot{d}_{1i}}} \Big)^{\to 0} \Big) sgn(\underline{E_1}) \big(\overbrace{1 - \underline{E_2} sgn(\underline{E_2})}^{\underline{\mu_{2j}}} \big) \\ &+ \Big(\Big(\frac{x_2 - \underline{c_{2j}}}{\underline{d_{2j}^2}} \Big) \underline{\dot{d}_{2j}} - \Big(\frac{\dot{x}_2 - \underline{\dot{c}_{2j}}}{\underline{\dot{d}_{2j}}} \Big)^{\to 0} \Big) sgn(\underline{E_2}) \big(\overbrace{1 - \underline{E_1} sgn(\underline{E_1})}^{\underline{\mu_{1i}}} \big) \\ &= -2\alpha sgn(\tau_c) \underline{\mu_{1i}} \underline{\mu_{2j}} \\ &= -2\alpha sgn(\tau_c) \underline{W_{ij}} \end{aligned} \tag{C.10}$$

$$\begin{aligned}
\dot{\widetilde{\underline{W_{ij}}}} &= -2\alpha\widetilde{\underline{W_{ij}}}sgn(\tau_c) - \frac{(\widetilde{\underline{W_{ij}}})\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(-2\alpha\underline{W_{ij}}sgn(\tau_c)\Big)\Big)}{(\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}})} \\
&= -2\alpha\widetilde{\underline{W_{ij}}}sgn(\tau_c) + (\widetilde{\underline{W_{ij}}})\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(2\alpha\widetilde{\underline{W_{ij}}}sgn(\tau_c)\Big)\Big) \\
&= -2\alpha\widetilde{\underline{W_{ij}}}sgn(\tau_c) + 2\alpha sgn(\tau_c)(\widetilde{\underline{W_{ij}}})\Big(\overbrace{\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(\widetilde{\underline{W_{ij}}}\Big)}^{1}\Big) \\
&= -2\alpha\widetilde{\underline{W_{ij}}}sgn(\tau_c) + 2\alpha\widetilde{\underline{W_{ij}}}sgn(\tau_c) = 0
\end{aligned} \tag{C.11}$$

Similarly, it can easily be shown that:

$$\dot{\widetilde{\overline{W_{ij}}}} = 0 \tag{C.12}$$

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

$$V_c = \frac{1}{2}\tau_c^2(t) \tag{C.13}$$

$$\dot{V_c} = \tau_c\dot{\tau_c} = \tau_c(\dot{\tau_n} + \dot{\tau}) \tag{C.14}$$

$$\begin{aligned}
\tau_n &= \frac{q\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\underline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W_{ij}}} + \frac{(1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\overline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\overline{W_{ij}}} \\
&= q\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\widetilde{\underline{W_{ij}}} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\widetilde{\overline{W_{ij}}}
\end{aligned} \tag{C.15}$$

$$\dot{\tau_n} = q\sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f_{ij}}\widetilde{\underline{W_{ij}}} + f_{ij}\dot{\widetilde{\underline{W_{ij}}}}) + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f_{ij}}\widetilde{\overline{W_{ij}}} + f_{ij}\dot{\widetilde{\overline{W_{ij}}}}) \tag{C.16}$$

$$
\begin{aligned}
\dot{V}_c &= \tau_c\Bigg[q\sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f}_{ij}\underline{\widetilde{W_{ij}}} + f_{ij}\cancelto{0}{\dot{\underline{\widetilde{W}}}_{ij}}) \\
&+ (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}(\dot{f}_{ij}\widetilde{\overline{W_{ij}}} + f_{ij}\cancelto{0}{\dot{\widetilde{\overline{W}}}_{ij}}) + \dot{\tau}\Bigg] \\
&= \tau_c\Bigg[q\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{\widetilde{W_{ij}}}\dot{f}_{ij} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\widetilde{\overline{W_{ij}}}\dot{f}_{ij} + \dot{\tau}\Bigg] \\
&= \tau_c\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J}\left(q\underline{\widetilde{W_{ij}}}\dot{f}_{ij} + (1-q)\widetilde{\overline{W_{ij}}}\dot{f}_{ij}\right) + \dot{\tau}\Bigg] \\
&= \tau_c\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f}_{ij}\left(q\underline{\widetilde{W_{ij}}} + (1-q)\widetilde{\overline{W_{ij}}}\right) + \dot{\tau}\Bigg] \qquad \text{(C.17)}
\end{aligned}
$$

If Equation 7.40 is inserted to the equation above, the following can be obtained:

$$
\dot{V}_c = \tau_c\Big[-\alpha sgn(\tau_c) + \dot{\tau}\Big] < \Big[-\alpha|\tau_c| + |\tau_c|B_{\dot{\tau}}\Big] < 0 \qquad \text{(C.18)}
$$

APPENDIX D: PROOF OF THEOREM 4

Evaluating the time derivative of the Lyapunov function in Equation 7.44 yields:

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

Remark: The obtained result means that, assuming the SMC task is achievable, using τ_c as a learning error for the T2FNN together with the adaptation laws Equations 7.32-7.40 enforces the desired reaching mode followed by a sliding regime for the system under control.

REFERENCES

1. Zadeh, L., "Toward extended fuzzy logic-A first step", *Fuzzy Sets and Systems*, Vol. 160, pp. 3175 – 3181, 2009.

2. Mendel, J. M., *Uncertain Rule-Based Fuzzy Logic System: Introduction and New Directions*, Prentice Hall, Upper Saddle River, 2001.

3. 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.

4. 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.

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

6. 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.

7. 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.

8. 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.

9. Shuanghe, Y., Y. Xinghuo, and M. Zhihong, "Fuzzy Sets and Systems", *A fuzzy neural network approximator with fast terminal sliding mode and its applications*, Vol. 148, No. 3, pp. 469–486, 11 2004.

10. 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.

11. 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.

12. Wang, L. X. and J. M. Mendel, "Back-Propagation of Fuzzy Systems as Nonlinear Dynamic System Identifiers", *Proc. of the IEEE Int'l. Conference of Fuzzy Systems*, pp. 1409–1418, San Diego, CA, USA, 1992.

13. Horikawa, S., T. Furahashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with back-propagation algorithm", *IEEE Trans. on Neural Networks*, pp. 801–806, 1992.

14. Zadeh, L., "Fuzzy Sets", *Information and Control*, Vol. 8, pp. 338–353, 1965.

15. Mamdani, E. H., "Applications of fuzzy algorithms for control of a simple dynamic plant", *Proceedings of IEE*, Vol. 121, pp. 1585–1588, 1974.

16. Karnik, N. N., J. M. Mendel, and Q. Liang, "Type-2 Fuzzy Logic Systems", *IEEE Trans. Fuzzy Syst.*, Vol. 7, pp. 643–658, 1999.

17. Mendel, J. M., R. John, and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple", *IEEE Trans. Fuzzy Syst.*, Vol. 14, pp. 808–821, 2006.

18. Juang, C.-F. and C.-H. Hsu, "Reinforcement Interval Type-2 Fuzzy Controller Design by Online Rule Generation and Q-Value-Aided Ant Colony Optimization", *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, Vol. 39, No. 6, pp. 1528 –1542, dec. 2009.

19. Juang, C. and Y. Tsao, "A Self-Evolving Interval Type-2 Fuzzy Neural Network With Online Structure and Parameter Learning", *IEEE Trans. on Fuzzy Systems*, Vol. 16, pp. 1411–1424, 2008.

20. Sepulveda, R., P. Melin, A. Rodriguez, A. Mancilla, and O. Montiel, "Analyzing the effects of the footprint of uncertainty in type-2 fuzzy logic controllers", *Engineering Letters*, Vol. 13, pp. 138–147, 2006.

21. Khanesar, M. A., M. Teshnehlab, E. Kayacan, and O. Kaynak, "A Novel Type-2 Fuzzy Membership Function: Application to the Prediction of Noisy Data", *Computational intelligence for measurement system and applications*, pp. 128 – 133, sep. 2010.

22. Biglarbegian, M., W. Melek, and J. 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, april 2011.

23. Karnik, N. and J. Mendel, "Applications of type-2 fuzzy logic systems to forecasting of time-series", *Information Sciences 120*, pp. 89 – 111, 1999.

24. Hagras, H. A., "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots", *IEEE Transactions on Fuzzy Systems*, pp. 524 –539, 2004.

25. Wang, C., C. Cheng, and T. Lee, "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)", *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 34*, pp. 1462 –1477, 2004.

26. Huang, S. and Y. Chen, "VLSI implementation of type-2 fuzzy inference processor", *in: Proceedings of IEEE ISCAS 2005*, pp. 3307 –3310, May. 2005.

27. 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.

28. Mendel, J. M. and R. I. B. John, "Type-2 Fuzzy Sets Made Simple", *IEEE transactions on fuzzy systems*, Vol. 10, pp. 117–127, 2002.

29. Karnik, N. N. and J. M. Mendel, "Operations on type-2 fuzzy sets", *Fuzzy Sets and Systems*, Vol. 122, No. 2, pp. 327 – 348, 2001.

30. Liang, Q., *Fading Channel Equalization and Video Traffic Classification Using Nonlinear Signal Processing Techniques*, University of Southern California, University of Southern California, USA, 2000.

31. Karnik, N. and J. Mendel, *An introduction to type-2 fuzzy logic systems*, 1998, `http://sipi.usc.edu/~mendel/report`, October 1998.

32. Begian, M., W. Melek, and J. Mendel, "Parametric design of stable type-2 TSK fuzzy systems", *Fuzzy Information Processing Society, 2008. NAFIPS 2008. Annual Meeting of the North American*, pp. 1 –6, may 2008.

33. Juang, C.-F. and C.-H. Hsu, "Reinforcement Interval Type-2 Fuzzy Controller Design by Online Rule Generation and Q-Value-Aided Ant Colony Optimization", *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 39, pp. 1528 – 1542, 2009.

34. Sepulveda, R., P. Melin, A. Rodriguez, A. Mancilla, and O. Montiel, "Analyzing the effects of the Footprint of Uncertainty in Type-2 Fuzzy Logic Controllers", *Engineering Letters*, Vol. 13, pp. 138–147, 2006.

35. Mendel, J. M., "Uncertainty, fuzzy logic, and signal processing", *Signal Processing*, Vol. 80, pp. 913 – 933, 2000.

36. Hsiao, M.-Y., T.-H. S. Li, J. Z. Lee, C. H. Chao, and S. H. Tsai, "Design of interval type-2 fuzzy sliding-mode controller", *Information Sciences: an International*

Journal, Vol. 178, pp. 1696 – 1716, 2008.

37. Mackey, M. and L. Glass, "Oscillation and Chaos in Physical control System", *Science*, Vol. 197, pp. 287–289, 1977.

38. Sepulveda, R., O. Castillo, P. Melin, O. Montiel, and L. Aguilar, "Evolutionary optimization of interval type-2 membership functions using the Human Evolutionary Model", *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*, pp. 1 –6, London, UK, 2007.

39. "Database for the Identification of Systems", 2010, `http://homes.esat.kuleuven.be/\textasciitildesmc/daisy/daisydata.html`.

40. Chen, F.-C. and H. Khalil, "Adaptive control of nonlinear systems using neural networks", *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, pp. 1707 –1712 vol.3, 5-7 1990.

41. Ku, C.-C. and K. Lee, "Diagonal recurrent neural networks for dynamic systems control", *Neural Networks, IEEE Transactions on*, Vol. 6, No. 1, pp. 144 –156, jan 1995.

42. 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.

43. Chen, T. and Y.-C. Lin, "A fuzzy back-propagation-network ensemble with example classification for lot output time prediction in a wafer FAB", *Applied Soft Computing*, Vol. 2, pp. 658–666, 2009.

44. Hwang, C. and F. C.-H. Rhee, "Uncertain Fuzzy Clustering: Interval Type-2 Fuzzy Approach to C-Means", *IEEE Trans. on Fuzzy Systems*, Vol. 15, pp. 107–120, 2007.

45. Abiyev, R. H., O. Kaynak, T. Alshanableh, and F. Mamedov, "A Type-2 Neuro-fuzzy System Based on Clustering and Gradient Techniques Applied to System

Identification and Channel Equalization", *Applied Soft Computing*, Vol. 11, pp. 1396–1406, 2011.

46. Kasabov, N., "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series", *IEEE Trans. on Fuzzy Systems*, Vol. 10, pp. 144–154, 2002.

47. Yager, R. and D. Filev, "Generation of Fuzzy Rules by Mountain Clustering", *Journal of Intelligent & Fuzzy Systems*, Vol. 2, pp. 209–219, 1994.

48. Wang, J., *A course in fuzzy systems and control*, Prentice Hall, Upper Saddle River, 1997.

49. Wang, L. and C. Wei, "Approximation accuracy of some neuro-fuzzy systems", *IEEE Trans. on Fuzzy Systems*, Vol. 8, pp. 470–478, 2000.

50. Lee, C.-H., J.-L. Hong, Y.-C. Lin, and W.-Y. Lai, "Type-2 Fuzzy Neural Network Systems and Learning", *Int. Journal of Computational Cognition*, Vol. 1, pp. 79–90, 2003.

51. Mendez, G., A. Cavazos, L. Leduc, and R. Soto, "Modeling of a Hot Strip Temperature Using Hybrid Learning for Interval Type-1 and Type-2 Non-Singleton Type-2 FLS", *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, pp. 529–533, Benalmadena, Spain, 2003.

52. Efe, M. O., O. Kaynak, and B. M. Wilamowski, "Stable Training of Computationally Intelligent Systems By Using Variable Structure Systems Technique", *IEEE Trans. on Industrial Electronics*, Vol. 47, pp. 487–496, 2000.

53. Topalov, A. V., G. L. Cascella, V. Giordano, F. Cupertino, and O. Kaynak, "Sliding Mode Neuro-Adaptive Control of Electric Drives", *IEEE Trans. on Industrial Electronics*, Vol. 54, pp. 671–679, 2007.

54. Claudia, P. and S. Miguel, "Speed Control of a DC Motor by Using Fuzzy Variable

Structure Controller", *Proc. of the IEEE 2008 27th Chinese Control Conference*, pp. 311–315, Kunming,Yunnan, China, 2008.

55. Wang, C., C. Cheng, and T. Lee, "Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network", *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. 34, pp. 1462–1477, 2004.

56. Hagras, H., "Comments on Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network (T2FNN)", *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. 36, pp. 1206–1209, 2006.

57. Mendel, J., "Computing Derivatives in Interval type-2 Fuzzy Logic Systems", *IEEE Transactions on Fuzzy Systems*, Vol. 12, pp. 84–98, 2004.

58. Mendez, G. M. and O. Castillo, "Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm", *Proc. of the IEEE Int. Conf. Fuzzy Systems*, pp. 230–235, 2005.

59. Abiyev, R., "Fuzzy Wavelet Neural Network for Prediction of Electricity Consumption", *AIEDAM: Artificial Intelligence for Engineering Design*, Vol. 23, pp. 109–118, 2009.

60. Abiyev, R. H. and O. Kaynak, "Fuzzy Wavelet Neural Networks for Identification and Control of Dynamic Plants - A Novel Structure and a Comparative Study", *IEEE Trans. on Industrial Electronics*, Vol. 55, pp. 3133–3140, 2008.

61. Abiyev, R. and O. Kaynak, "Identification and Control of Dynamic Plants Using Fuzzy Wavelet Neural Networks", *IEEE Multi-conference on Systems and Control*, pp. 1295–1301, San Antonio, Texas (USA), 2008.

62. AMIRACorp., "DR300 Laboratory setup speed control with variable load", Technical report, AMIRA, 2000.

63. Ruano, A., C. Cabrita, J. Oliveira, D. Tikk, and L. Koczy, "Supervised training

algorithms for B-spline neural networks and fuzzy systems", *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, pp. 2830 –2835 vol.5, jul. 2001.

64. Yong, P. and L. Guo-hua, "A Fuzzy Optimization Neural Network Model Using Second Order Information", *Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on*, Vol. 4, pp. 221 –227, aug. 2009.

65. Palit, A. and R. Babuska, "Efficient training algorithm for Takagi-Sugeno type neuro-fuzzy network", *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, Vol. 3, pp. 1367 –1371, 2001.

66. Wang, F., S. Wen, and C. Wu, "Gait pattern estimation for intelligent bionic leg", *Control and Decision Conference, 2008. CCDC 2008. Chinese*, pp. 2216 –2221, jul. 2008.

67. Amini, F. and M. Shahidzadeh, "Damage detection using a new regularization method with variable parameter", *Archive of Applied Mechanics*, Vol. 80, pp. 255–269, 2010.

68. cong Chen, T., D. jian Han, F. Au, and L. Tham, "Acceleration of Levenberg-Marquardt training of neural networks with variable decay rate", *Neural Networks, 2003. Proceedings of the International Joint Conference on*, Vol. 3, pp. 1873 – 1878 vol.3, jul. 2003.

69. Beyhan, S. and M. Alci, "Extended fuzzy function model with stable learning methods for online system identification", *International Journal of Adaptive Control and Signal Processing*, Vol. 25, No. 2, pp. 168–182, 2011.

70. Celikyilmaz, A. and I. B. Turksen, "Uncertainty Modeling of Improved Fuzzy Functions With Evolutionary Systems", *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 38, pp. 1098–1110, 2008.

71. Panigrahi, S. P., S. K. Nayak, and S. K. Padhy, "A genetic-based neuro-fuzzy controller for blind equalization of time-varying channels", *International Journal of Adaptive Control and Signal Processing*, Vol. 22, No. 7, pp. 705–716, 2008.

72. Yu, X. and O. Kaynak, "Sliding-Mode Control With Soft Computing: A Survey", *Industrial Electronics, IEEE Transactions on*, Vol. 56, No. 9, pp. 3275 –3285, 2009.

73. Kaynak, O., K. Erbatur, and M. Ertugrul, "The fusion of computationally intelligent methodologies and sliding-mode control-a survey", *Industrial Electronics, IEEE Transactions on*, Vol. 48, No. 1, pp. 4 –17, February 2001.

74. 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.

75. Yoo, B. and W. Ham, "Adaptive fuzzy sliding mode control of nonlinear system", *Fuzzy Systems, IEEE Transactions on*, Vol. 6, No. 2, pp. 315 –321, may 1998.

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

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

78. Iliev, B. and I. Kalaykov, "Minimum-time sliding mode control for second-order systems", *American Control Conference, 2004. Proceedings of the 2004*, Vol. 1, pp. 626 –631 vol.1, 30 2004-july 2 2004.

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

80. 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.

81. Shakev, N. G., A. V. Topalov, and O. Kaynak, "Sliding mode algorithm for on-line

learning in analog multilayer feedforward neural networks", *in the book: Kaynak et al. (Eds): Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003, Lecture Notes in Computer Science, Springer-Verlag*, pp. 1064 –1072, 2003.

82. Loria, A., E. Panteley, and H. Nijmeijer, "Control of the chaotic Duffing equation with uncertainty in all parameters", *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, Vol. 45, No. 12, pp. 1252 –1255, dec 1998.

83. Merritt, H. E., *Hydraulic Control Systems*, John Wiley & Sons, Inc., 1967.

84. Watton, J., *Fluid Power Systems*, Prentice Hall, 1989.

85. Jovanovic, M., "Nonlinear control of an electrohydraulic velocity servosystem", *American Control Conference, 2002. Proceedings of the 2002*, Vol. 1, pp. 588 – 593, 2002.