ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A MODIFIED ANFIS SYSTEM FOR AERIAL VEHICLES CONTROL

Ph.D. THESIS

Muhammet ÖZTÜRK

Department of Aeronautics and Astronautics Engineering

Aeronautics and Astronautics Engineering Programme

FEBRUARY 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A MODIFIED ANFIS SYSTEM FOR AERIAL VEHICLES CONTROL

Ph.D. THESIS

Muhammet ÖZTÜRK
(511142119)

Department of Aeronautics and Astronautics Engineering

Aeronautics and Astronautics Engineering Programme

Thesis Advisor: Prof. Dr. İbrahim ÖZKOL

FEBRUARY 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

HAVA ARAÇLARI KONTROLÜ İÇİN DEĞİŞTİRİLMİŞ ANFIS SİSTEMİ

DOKTORA TEZİ

Muhammet ÖZTÜRK
(511142119)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Prof. Dr. İbrahim ÖZKOL

ŞUBAT 2022

Muhammet ÖZTÜRK, a Ph.D. student of ITU Graduate School student ID 511142119 successfully defended the thesis entitled “A MODIFIED ANFIS SYSTEM FOR AERIAL VEHICLES CONTROL”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. İbrahim ÖZKOL** Istanbul Technical University	....
Jury Members :	**Prof. Dr. Fikret ÇALIŞKAN** Istanbul Technical University	....
	Prof. Dr. Vasfi Emre ÖMÜRLÜ Yıldız Technical University	....
	Asst. Prof. Dr. Sıddık Murat YEŞİLOĞLU Istanbul Technical University	....
	Asst. Prof. Dr. Barış GÖKÇE Necmettin Erbakan University	....

Date of Submission : **31 December 2021**
Date of Defense : **11 February 2022**

To my family,

FOREWORD

First and foremost, I would like to express my special appreciation and gratitude to my supervisor, Prof. Dr. İbrahim ÖZKOL, for his guidance, encouragement and kindness throughout this study. His advice and support has been priceless in this doctorate process. I would like to thank my previous supervisor, Asst. Prof. Dr. İsmail BAYEZİT, with whom we started this research together, for his support and guidance and valuable suggestions.

I would like to thank Prof. Dr. Fikret ÇALIŞKAN and Prof. Dr. Vasfi Emre ÖMÜRLÜ for serving on my thesis evaluation committee members and later being the part of my thesis defense jury. I would also like to thank Asst. Prof. Dr. İlker ÜSTOĞLU for serving on my thesis evaluation committee member. Their valuable suggestions and comments on my reports and thesis helped me improve the quality of the thesis. I also thank Asst. Prof. Dr. Sıddık Murat YEŞİLOĞLU and Asst. Prof. Dr. Barış GÖKÇE for being part of my thesis defence jury.

I am also grateful to researchers Muhsin HANÇER, Rahman BİTİRGEN and Hacı BARAN for their moral support and friendship.

And finally, I would like to thank my father, Ahmet ÖZTÜRK, my mother, Hatice ÖZTÜRK, my wife, Tuba ÖZTÜRK, my brothers and sister for their endless support and love.

February 2022

Muhammet ÖZTÜRK
(Astronautical Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

A1-C1	: Antecedent type-1 MF - Consequent type-1 MF
A2-C1	: Antecedent type-2 MF - Consequent type-1 MF
A2-C2	: Antecedent type-2 MF - Consequent type-2 MF
ANFIS	: Adaptive Neuro Fuzzy Inference System
CI	: Computational Intelligence
DOF	: Degree of Freedom
EIASC	: Enhanced Iterative Algorithm with Stop Condition
EKMA	: Enhanced Karnik Mendel Algorithm
FEL	: Feedback Error Learning
FIS	: Fuzzy Inference System
FLC	: Fuzzy Logic Control
FOU	: Footprint of Uncertainty
GD	: Gradient Descent
IASC	: Iterative Algorithm with Stop Condition
IT2FIS	: Interval Type-2 Fuzzy Inference System
IT2FLC	: Interval Type-2 Fuzzy Logic Control
IT2FLS	: Interval Type-2 Fuzzy Logic System
KMA	: Karnik-Mendel Algorithm
LMF	: Lower Membership Function
LSE	: Least Square Estimation
M-KMA	: Modified Karnik-Mendel Algorithm
MANFIS	: Mamdani Adaptive Neuro Fuzzy Inference System
MF	: Membership Function
MIMO	: Multi Input Multi Output
NFC	: Neuro-Fuzzy Control
NN	: Neural-Network
RMSE	: Root Mean Square Error
SISO	: Single Input Single Output
SMC	: Sliding Mode Control
T1FIS	: Type-1 Fuzzy Inference System
T2FIS	: Type-2 Fuzzy Inference System
TS	: Takagi-Sugeno
UMF	: Upper Membership Function

LIST OF TABLES

LIST OF FIGURES

A MODIFIED ANFIS SYSTEM FOR AERIAL VEHICLES CONTROL

SUMMARY

This thesis presents fuzzy logic systems (FLS) and their control applications in aerial vehicles. In this context, firstly, type-1 fuzzy logic systems and secondly type-2 fuzzy logic systems are examined. Adaptive Neuro-Fuzzy Inference System (ANFIS) training models are examined and new type-1 and type-2 models are developed and tested. The new approaches are used for control problems as quadrotor control.

Fuzzy logic system is a humanly structure that does not define any case precisely as 1 or 0. The Fuzzy logic systems define the case with membership functions. In literature, there are very much fuzzy logic applications as data processing, estimation, control, modeling, etc. Different Fuzzy Inference Systems (FIS) are proposed as Sugeno, Mamdani, Tsukamoto, and Şen. The Sugeno and Mamdani FIS are the most widely used fuzzy logic systems. Mamdani antecedent and consequent parameters are composed of membership functions. Because of that, Mamdani FIS needs a defuzzification step to have a crisp output. Sugeno antecedent parameters are membership functions but consequent parameters are linear or constant and so, the Sugeno FIS does not need a defuzzification step.

The Sugeno FIS needs less computational load and it is simpler than Mamdani FIS and so, it is more widely used than Mamdani FIS. Training of Mamdani parameters is more complicated and needs more calculation than Sugeno FIS. The Mamdani ANFIS approaches in the literature are examined and a new Mamdani ANFIS model (MANFIS) is proposed. Training performance of the proposed MANFIS model is tested for a nonlinear function and control performance is tested on a DC motor dynamic. Besides, Şen FIS that was used for estimation of sunshine duration in 1998, is examined. This ŞEN FIS antecedent and consequent parameters are membership functions as Mamdani FIS and needs to defuzzification step. However, because of the structure of the Şen defuzzification structure, the Şen FIS can be calculated with less computational load, and therefore Şen ANFIS training model has been created. These three approaches are trained on a nonlinear function and used for online control.

In this study, the neuro-fuzzy controller is used as online controller. Neuro-fuzzy controllers consist of simultaneous operation of two functions named fuzzy logic and ANFIS. The fuzzy logic function is the one that generates the control signal. It generates a control signal according to the controller inputs. The other function is the ANFIS function that trains the parameters of the fuzzy logic function. Neuro-fuzzy controllers are intelligent controllers, independent of the model, and constantly adapting their parameters. For this reason, these controllers' parameters values are constantly changing according to the changes in the system. There are studies on

different neuro-fuzzy control systems in the literature. Each approach is tested on a DC motor model that is a single-input and single-output system, and the neuro-fuzzy controllers' advantages and performances are examined. In this way, the approaches in the literature and the approaches added within the scope of the thesis are compared to each other. Selected neuro-fuzzy controllers are used in quadrotor control.

Quadrotors have a two-stage controller structure. In the first stage, position control is performed and the position control results are defined as angles. In the second stage, attitude control is performed over the calculated angle values. In this thesis, the neuro-fuzzy controller is shown to work perfectly well in single layer control structures, i.e., there was not any overshooting, and settling time was very short. But it is seen from quadrotor control results that the neuro-fuzzy controller can not give the desired performance in the two-layered control structure. Therefore, the feedback error learning control system, in which the fuzzy controller works together with conventional controllers, is examined.

Fundamentally, there is an inverse dynamic model parallel to a classical controller in the feedback error learning structure. The inverse dynamic model aims to increase the performance by influencing the classical controller signal. In the literature, there are a lot of papers about the structure of feedback error learning control and there are different proposed approaches. In the structure used in this work, fuzzy logic parameters are trained using ANFIS with error input.The fuzzy logic control signal is obtained as a result of training. The fuzzy logic control signal is added to the conventional controller signal. This study has been tested on models such as DC motor and quadrotor. It is seen that the feedback error learning control with the ANFIS increases the control performances.

Antecedent and consequent parameters of type-1 fuzzy logic systems consist of certain membership functions. A type-2 FLS is proposed to better define the uncertainties, because of that, type-2 fuzzy inference membership functions are proposed to include uncertainties. The type-2 FLS is operationally difficult because of uncertainties. In order to simplify type-2 FLS operations, interval type-2 FLS is proposed as a special case of generalized type-2 FLS in the literature.

Interval type-2 membership functions are designed as a two-dimensional projection of general type-2 membership functions and represent the area between two type-1 membership functions. The area between these two type-1 membership functions is called Footprint of Uncertainty (FOU). This uncertainty also occurs in the weight values obtained from the antecedent membership functions. Consequent membership functions are also type-2 and it is not possible to perform the defuzzification step directly because of uncertainty. Therefore, type reduction methods have been developed to reduce the type-2 FLS to the type-1 FLS. Type reduction methods try to find the highest and lowest values of the fuzzy logic model. Therefore, a switch point should be determined between the weights obtained from the antecedent membership functions. Type reduction methods find these switch points by iterations and this process causes too much computation, so many different methods have been proposed to minimize this computational load. In 2018, an iterative-free method called Direct Approach (DA) was proposed. This method performs the type reduction process faster than other iterative methods.

In the literature, studies such as neural networks and genetic algorithms on the training for parameters of the type-2 FLS still continue. These studies are also used in the interval type-2 fuzzy logic control systems. There are proposed interval type-2 ANFIS structures in literature, but they are not effective because of uncertainties of interval type-2 membership functions.

FLS parameters for ANFIS training should not contain uncertainties. However, the type-2 FLS should inherently contain uncertainty. For this reason, Karnik-Mendel algorithm is modified, which is one of the type-reduction methods, to apply the ANFIS on interval type-2 FLS. The modified Karnik-Mendel algorithm gives the same results as the Karnik-Mendel algorithm. The modified Karnik-Mendel algorithm also gives exact parameter values for use in ANFIS. One can notice that the ANFIS training of the interval type-2 FLS has been developed successfully and has been used for system control.

HAVA ARAÇLARI KONTROLÜ İÇİN DEĞİŞTİRİLMİŞ ANFIS SİSTEMİ

ÖZET

Bu tez; bulanık çıkarım sistemlerini ve eğitimlerini inceleyerek bulanık çıkarım sistemlerinin kontrol alanında uygulamalarını incelemektedir. Yapılan çalışmada öncelikle tip-1 bulanık mantık sistemleri incelenmiş, sonra tip-2 bulanık sistemleri incelenmiştir. Literatürde ki eğitim modelleri incelenerek yeni eğitim modelleri geliştirilmiş ve belli denklemler için test edilmiştir. Bu yeni modeller ile quadrotor kontrolü yapılmıştır.

Bulanık mantık sistemleri insanların düşünsel yapısı göz önünde bulundurularak geliştirilmiş sistemlerdir. Herhangi bir olaya 1 veya 0 gibi bir yaklaşım sergilemek yerine belli üyelik dereceleri ile yaklaşılmaktadır. Bu sistemler ortaya çıktıktan sonra literatürde veri işleme, tahmin, sistem kontrolü, modelleme vb. çok farklı alanlarda uygulanmışlardır. Bugün için ulaşılabilir literatürde Sugeno, Mamdani, Şen, Tsukamato gibi farklı çıkarım sistemleri türetilmiş ve kullanılmıştır. Mamdani ve Sugeno bu sistemlerin en yaygın kullanılanlarıdır. Mamdani bulanık çıkarım sisteminin giriş ve çıkış parametreleri üyelik fonksiyonlarından oluşmaktadır. Çıkış parametreleri üyelik fonksiyonlarından oluştuğu için durulaştırma adımına ihtiyaç duymaktadır. Sugeno bulanık çıkarım sistemi ise hesap yükü daha az olan bir sistemdir ve çıkış parametreleri sabit veya doğrusal olarak seçilebilir. Çıkış parametreleri üyelik fonksiyonu barındırmadığı için tekrar durulaştırma işlemine ihtiyaç duymamaktadır.

Sugeno bulanık çıkarım sistemi, çıkış parametrelerinin doğrusal veya sabit olmasından kaynaklı olarak daha az matematiksel yük içerdiğinden daha yaygın olarak kullanılmaktadır. Ne var ki, çıkış parametreleri üyelik fonksiyonları şeklinde olan Mamdani bulanık çıkarım sisteminin parametrelerinin eğitimi daha zor olmaktadır ve kullanım alanı daha kısıtlı kalmıştır. Bu tezde literatürde bulunan eğitim modelleri incelenmiş ve Mamdani için yeni bir eğitim modeli oluşturulmuştur. Ayrıca 1998 yılında güneş ışınımının tahmini için kullanılan Şen bulanık çıkarım sistemi de incelenmiştir. Bu bulanık çıkarım sisteminin giriş ve çıkış parametreleri Mamdani gibi bulanıktır ve durulaştırma işlemine ihtiyaç duymaktadır. Fakat, çıkarım sisteminden kaynaklı olarak daha az hesaplama yükü ile hesaplanabilmektedir. Ulaşılabilir literatürde Şen bulanık çıkarım sisteminin eğitimi üzerine çalışma bulunmadığı için tez çalışması kapsamında yeni bir eğitim modeli oluşturulmuştur. Bu üç yaklaşımında eğitimi bir örnek üzerinden yapılarak karşılaştırılmış ve çevrimiçi olarak kontrol için kullanılmışlardır.

Bu tez kapsamında nöro-bulanık kontrolcüler üzerinde çalışılmıştır. Nöro-bulanık kontrolcüler akıllı kontrolcü olup, modelden bağımsızdırlar ve sürekli olarak kendi parametrelerini eğiterek değiştirmektedirler. Bu sebeple, bu kontrolcünün

parametreleri sistem çalışırken sürekli olarak değişmektedirler. Nöro-bulanık kontrolcüler iki ayrı fonksiyonun aynı anda çalışmasından oluşmaktadır. Bu fonksiyonlar bulanık mantık fonksiyonu ve eğitim fonksiyonu olarak isimlendirilebilir. Bulanık mantık fonksiyonu kontrol sinyalini üreten fonksiyondur ve bulanık kontrol sistemine giren girdilere göre kontrol sinyali üretmektedir. Eğitim fonksiyonu ise uyarlamalı nöro-bulanık çıkarım sisteminden (ANFIS) oluşmaktadır. Bu fonksiyon bulanık mantık fonksiyonunun parametrelerinin eğitimini yapmaktadır. Literatürde farklı nöro-bulanık kontrol sistemleri üzerine çalışmalar bulunmaktadır. Bu yaklaşımlar tez kapsamında tek giriş ve tek çıkışlı bir sistem olan bir motor modeli üzerinde denenerek avantajları ve performansları incelenmiştir. Bu sayede literatürdeki yaklaşımlar ve tez kapsamında eklenen yaklaşımlar incelenmiştir. Seçilen nöro-bulanık kontrolcüler quadrotor modeli üzerinde test edilmiştir.

Quadrotörler iki aşamalı bir kontrolcü yapısına sahiptir. Birinci aşamada pozisyon kontrolü yapılmaktadır ve bu kontrol sayesinde açı değerlerine ulaşılmaktadır. İkinci aşamada hesaplanan açı değerleri üzerinden yönelme kontrolü yapılmaktadır. Yapılan testler sonucunda nöro-bulanık kontrolcünün tek katmanlı yapılarda mükemmel derecede iyi çalıştığı görülmüştür yani aşım oluşmamış, oturma zamanı ise çok kısa olmuştur. Yalnız iki kontrolün peş peşe olduğu katlı yapıda nöro-bulanık kontrolcünün istenen performansı veremediği görülmüştür. Bu sebeple bulanık kontrolcünün başka kontrolcülerle beraber çalıştığı Geri-Beslemeli hata öğrenme kontrol sistemi incelenmiştir. Bu kontrol yapısında genel olarak klasik bir kontrolcüye paralel şekilde dinamik modelin tersini içeren bir model daha bulunmaktadır. Ters dinamik model klasik kontrolcüden gelen kontrol sinyalini etkileyerek kontrolcü performansını artırmayı hedeflemektedir. Literatürde bu geri-beslemeli hata öğrenme kontrolünün yapısı üzerine çok çalışma yapılmış ve farklı yaklaşımlar önerilmiştir. Bu tezde kullanılan yapıda hata girişi ile beraber uyarlamalı nöro-bulanık çıkarım sistemi kullanılarak bulanık mantık parametreleri eğitilmiştir. Bu parametrelerin eğitimi sonucunda elde edilen kontrol sinyali klasik kontrol sinyalinin üzerine eklenmiştir. Bu çalışma DC motor ve quadrotor gibi modeller üzerinde test edilmiştir ve kontrol performanslarının çok iyi düzeyde arttığı görülmüştür.

Tip-1 bulanık mantık sistemlerinin giriş ve çıkış parametreleri açık ve kesin üyelik fonksiyonlarından oluşmaktadır. Belirsizlikleri daha iyi tanımlayabilmek için tip-2 bulanık mantık sistemi önerilmiştir. Tip-2 üyelik fonksiyonları belirsizlikler içeren fonksiyonlardır ve bu durum tip-2 bulanık mantık sistemi ile işlem yapılmasını zorlaştırmaktadır. Bu sebeple literatürde tip-2 bulanık sistemin özel bir durumu olarak aralık değerli tip-2 bulanık mantık sistemi önerilmiştir. Aralık değerli tip-2 üyelik fonksiyonları, genel tip-2 üyelik fonksiyonlarının iki boyutlu iz düşümü şeklinde tasarlanmıştır. Bir aralık değerli tip-2 üyelik fonksiyonu 2 tane tip-1 üyelik fonksiyonun arasında kalan alan ile ifade edilmektedir. Bu iki tane tip-1 üyelik fonksiyonu arasında kalan alana Belirsizliğin Ayakizi ismi verilmiştir. Bu üyelik fonksiyonlarındaki belirsizlik sebebiyle üyelik fonksiyonlarından elde edilen üyelik değerleri de belirsizlik içermektedir. Çıkış üyelik fonksiyonları da aralık değerli tip-2 üyelik fonksiyonu olduğundan belirsizlik içermektedir ve bu nedenle durulaştırma işlemi yapmak imkansızdır. Durulaştırma işlemini yapabilmek için tip-2 bulanık çıkarım sistemi tip-1 bulanık çıkarım sistemine indirgenmelidir. Bunun için tip indirgeme yöntemleri oluşturulmuştur.

Tip indirgeme yöntemleri herhangi bir tip-2 bulanık mantık modelinin alabileceği en yüksek ve en düşük değerleri bulmaya çalışmaktadır. Bunun için üyelik değerlerinden elde edilen ağırlıkların arasında bir değişim noktası belirlenmelidir. Tip indirgeme yöntemleri bu değişim noktalarını iterasyonlar ile bulmaktadır. İterasyonlar yüksek hesaplama yükü içerdiğinden bu hesaplama yükünü minimize edecek birçok farklı yöntem önerilmiştir. En son 2018 yılında Direkt Yaklaşım isimli iterasyonsuz bir yöntem önerilmiştir. Bu yöntem kural sayısı yüksek olduğunda tip indirgeme işlemini iterasyonlu yöntemlere göre daha hızlı yapmaktadır ancak düşük kural sayılarında diğer yöntemlerden daha yavaş kalmaktadır.

Literatürde tip-2 ve aralık değerli tip-2 bulanık sistemlerinin parametrelerinin eğitimi üzerine sinir ağları, genetik algoritmalar gibi çalışmalar h*âlâ* devam etmektedir. Bu tez kapsamında aralık değerli tip-2 bulanık mantık sisteminin eğitimi için ANFIS yapısı incelenmiştir. Açık literatürde ANFIS yapısının tip-2 için kullanımının verimli olmadığı belirtilmiştir. Bunun sebebi ANFIS eğitiminin belirsizlik içermeyen parametrelerle yapılması gerekliliğidir. Fakat belirsizlik tip-2 bulanık çıkarım sisteminin doğasından gelmektedir ve bu sebeple tip-2 ve aralık değerli tip-2 bulanık çıkarım sistemlerinin içindeki değerler belirsizlik içermektedir.

Belirsiz parametreler sorununun üstesinden gelmek için, bu tezde tip indirgeme yöntemlerinden birisi olan Karnik-Mendel algoritması üzerinde değişiklikler yapılarak değiştirilmiş Karnik-Mendel algoritması oluşturulmuştur. Bu değiştirilmiş indirgeme yöntemi sayesinde belirsiz parametreler yerini belirsizlik içermeyen parametrelere bırakmıştır. Bu kesin parametre değerleri kullanılarak aralık değerli tip-2 bulanık mantık sisteminin parametrelerinin ANFIS eğitimi başarılı şekilde yapılabilmiştir. Geliştirilen aralık değerli tip-2 ANFIS yapısı bir doğrusal olmayan fonksiyon üzerinde test edilmiştir ve sistem kontrolü (DC motor, quadrotor) için kullanılmıştır.

1. INTRODUCTION

Aircrafts are nonlinear systems that include nonlinearities and uncertainties. Their controllers have to cope with noises and disturbances like wind, turbulence, etc. So, there are very many control researches that are tested on the aircrafts as linear (PID, Linear Quadratic Regulator), model based (Adaptive Control, Sliding mode Controller) and intelligent controllers (Neural Network, Fuzzy Logic). The linear controllers are the commonly utilized controllers in industry [1] that their parameters can be calculated with respect to requirements like overshoot, settling time, etc. These control methods are linear and so they have operating limits.

The model based controllers as Adaptive Control and Sliding Mode Controller (SMC) are nonlinear. The nonlinear controllers can cope with changes in system dynamics. However, their robustness is limited with system design. For example, an inverse model for adaptive control is determined before implementation [2] and in an unexpected disturbance or model change, the controller's operating range can be exceeded. However, intelligent controllers like Neural Network (NN), Genetic algorithms (GA) and Fuzzy Logic (FL) based controllers don't depend on the mathematical model and they can handle unexpected changes better than conventional control methods. In this thesis, Fuzzy Logic Systems (FLS) are examined and Fuzzy Logic Controllers (FLC) are tested on a quadrotor model.

1.1 Purpose of Thesis

Aerial vehicles are highly nonlinear systems and implementation of intelligent controllers on aerial vehicles is still a research area. This thesis aims to develop intelligent controllers and test nonlinear system dynamics. So, fuzzy logic systems in the literature are examined and new approaches are proposed for modelling and control. The new approaches are compared to existing models to see their advantages and disadvantages.

The fuzzy logic systems can be given under two headings as type-1 FLS and type-2 FLS. The type-1 FLS has different FISs as Sugeno, Mamdani, Tsukamato, etc. In the scope of this thesis, adaptive neuro fuzzy inference system (ANFIS) is constructed for Mamdani and Şen FIS. They are compared to Sugeno ANFIS for training performances and used for online ANFIS controller. In this way, in addition to the studies proposed in the literature, better control performance is tried to be obtained.

Type-2 FLS includes highly uncertainty in its antecedent and consequent parameters. So, in the literature, interval type-2 FLS is proposed to simplify the type-2 FLS calculations. The interval type-2 ANFIS is mentioned in the literature as ineffective and so a new ANFIS approach is developed in the scope of this thesis. A new type-reduction approach that the interval type-2 FLS needs it to have a crisp output, is proposed to develop the new ANFIS model. The developed interval type-2 ANFIS model is tested and compared with the type-1 ANFIS for training and control. The new interval type-2 ANFIS model works with less error when compared to type-1 ANFIS and this shows that it can be used more efficiently in modelling and control.

1.2 Literature Review

Fuzzy Logic System is one of the Computational Intelligent (CI) methods that are very useful to represent nonlinear systems and mathematically non-solvable problems [3,4]. There are two commonly used type-1 FLS as Sugeno and Mamdani. The Mamdani FLS antecedent and consequent parameters are fuzzy, so it needs to defuzzification step to have an exact value [5]. Because of its fuzzy consequent Membership Functions (MFs), the Mamdani FLS is linguistic and it has humanly structure. Sugeno FLS consequent parameters are crisp or linear functions and so, Sugeno FLS does not have a defuzzification step. Sugeno FLS is more effective at engineering problems than Mamdani FLS [6].

FLS parameters can be determined by expert opinion. But it is very hard to use expert opinion for complicated systems. So, training methods are proposed in the literature that ANFIS is the most used training approach [7,8]. The FLS training involves two stages as antecedent parameters training and consequent parameters training. In ANFIS, the antecedent parameters training is performed with Gradient Descent

(GD) algorithm and consequent parameters' training is performed with Least Square Estimation (LSE) or Gradient Descent algorithms [9]. For FLS training, different techniques can be used as stated in the literature, but the mentioned method needs less computational load and gives better results than other methods [10,11].

Because of the simplicity of the Sugeno FLS, the ANFIS is widely used in engineering problems. However, since Mamdani FLS consequent parameters are fuzzy, it is more suitable for human nature. So, in the literature, the Mamdani ANFIS (MANFIS) structure is proposed and compared to ANFIS for traffic solution problem [12]. The suggested MANFIS structure in the literature is examined and a new MANFIS model with better results is proposed in this thesis [13]. The new enhanced MANFIS model that is created under some assumptions is tested for modelling and control studies.

In 1998, Şen FLS was proposed to estimate solar irradiation from sunshine duration [14]. The proposed Şen FLS was used for single input single output system. The ŞEN FIS antecedent and consequent parameters are MFs but output calculation is simple as Sugeno FLS. In this thesis, we made the Şen FLS available for multi input multi output system. We proposed a new ŞEN ANFIS structure and tested it in modelling and control studies.

The aerial vehicle control system is still a main problem because of their nonlinearities, coupling effects, disturbances etc [15]. So, intelligent controllers like fuzzy logic controller (FLC) [16] and Neural Network (NN) [17] are used as aircraft controllers. In the literature, the FLCs were used directly to produce control signals, to tune PID coefficients or they were used together as Fuzzy-PID controllers. FLC was directly used to control pH system in [18,19], a DC motor [20], an ABS brake system [21], a 2 DOF helicopter system [22] etc. Besides, Fuzzy-PID controllers were examined in the literature [23]–[27].

The classical FLCs need optimized FLS parameters those are determined by expert opinion or off-line ANFIS trainings. So, their parameters are constant. These controllers must run in some intervals and assumptions. Because of that, it can go into uncertainty if unexpected changes occur in the system. To overcome this problem, intelligent controllers have been studied within the scope of this thesis. In this thesis, Neural Network based FLCs are examined. In 2007, a Neuro-Fuzzy Controller (NFC) was implemented to an Induction Motor Drive with two control inputs and a target function [28]. The results were compared to PI controller. In 2010, a new NFC with three control inputs is implemented to a Servo system and compared to PI [29]. In 2014, a two input NFC compared to PID on a 2 DOF helicopter system [30]. The methods proposed in the literature have been revised and new target functions and NFC structures have been created. The NFC methods that we proposed in this study have been tested and compared to each other for Sugeno, Mamdani and ŞEN FLC [31].

Quadrotor is a highly nonlinear system that has decoupling effects and needs a cascaded control structure [32]. There are very different control studies about quadrotor in open literature [1]. In this work, the NFC controllers have been tested on a quadrotor model and the results show that there is no overshoot and oscillation for the attitude controller. However, the NFC controller was not effective for quadrotor position control because of cascaded structure. Therefore, Feedback Error Learning (FEL) Controllers, which are a combination of classical controllers and NFC, were examined.

The FEL controller was proposed to help the main controller as inverse system dynamic [33]. The dynamic inverse model can be calculated mathematically [34,35] or any other learning method as Neural Network [36]. In 2007, a new FEL approach is proposed with on-line neural network learning algorithm [37]. This new approach was effectively used for control of nonlinear systems as satellites [38] and quadrotors [15] with Type-2 Fuzzy Logic System. The FEL controller has been tested for single input single output (SISO) and multi input multi output (MIMO) systems in this work.

Although T1 FLCs are used effectively for nonlinear systems, T1 Fuzzy Sets (FS) is not adequate to define uncertainties because type-1 MFs are precise [39]. So, Type-2 Fuzzy Sets (FS) are proposed by Lotfi A. Zadeh to directly include uncertainty into FSs [40]. The Type-2 FSs include uncertainty because of the type-2 MFs' three dimensional structure but are rather difficult to calculate [41]. Because of that, the three-dimensional structure of Generalized Type-2 FSs is reduced to two-dimensional structure named as Interval Type-2 Fuzzy Logic System (IT2 FLS) [42]. Even when the FSs are two-dimensional, the uncertainty continued and a crisp value could not be obtained as a result of the defuzzification step. So, Karnik-Mendel Algorithm (KMA) is proposed as a type-reduction method that reduces the Interval Type-2 FSs to Type-1 FSs [43]. The KMA finds an optimal switch point by iterations for reduction. Enhanced Karnik-Mendel Algorithm (EKMA) is proposed to reduce the computational cost that EKMA defines starting points for switching [44]. Iterative Algorithm with Stop Condition (IASC) [45] and Enhanced Iterative Algorithm with Stop Condition (EIASC) [46] are proposed to reduce the computational load by reducing the iteration numbers. In 2018, Direct Approach (DA) is proposed to minimize the computational load by eliminating the iterations and this study is tested within the scope of this thesis [47].

There are many IT2 FLC studies in the literature performed with the above-mentioned type-reduction methods and it is seen that the IT2 FLC is better than T1 FLC to handle uncertainties [48,49]. The Karnik-Mendel Algorithm was used for control of autonomous mobile robots and it has been stated that IT2 FLC is better than T1 FLC at dealing with uncertainties [39]. In [39], the IT2 FLS was used without any learning or model inversion. The first model inversion method for IT2 FLS is proposed to control a nonlinear Internal Model Control structure [50,51]. A precise method was proposed to create the inverse of the interval Type-2 Takagi-Sugeno FLS by using a pure analytical method. But, the presented method can only be used for SISO systems. The IT2 FLC and PID combinations are tested on a 2 DOF helicopter model [52] and on a pH control system [53]. These studies [52,53] indicate that robustness of IT2-FLCs has been observed. However, all of these IT2 FLCs were implemented in systems without intelligent learning. Training of IT2 FLS parameters is a complicated problem because of parameters uncertainty. There are very much IT2 FLS parameters training approaches in the literature such as Big Bang-Big Crunch optimization [54,55], gradient descent [56], neural network [57], sliding mode [15,58,59] that they are used in IT2 FLCs.

The IT2 ANFIS method given in the literature is examined. It is seen that the IT2 ANFIS results are worse than T1 ANFIS results [60,61]. This is because of uncertainty in the IT2 FLS antecedent and consequent parameters. The ANFIS parameters must be crisp for training. In this study, we proposed a new approach for Karnik-Mendel algorithm named Modified Karnik-Mendel algorithm (M-KMA) that M-KMA uses certain parameters to calculate FLS outputs. Thanks to this algorithm, the same results with uncertain parameters can be obtained with certain parameters [62]. This enhanced algorithm was tested for modelling and implemented in systems as controllers.

This thesis consists of 6 chapters. Chapter 1 discusses the purpose of the thesis and literature review. Chapter 2 discusses the type-1 ANFIS studies. In this chapter, two different ANFIS structures have been proposed for Mamdani FLS and Şen FLS. Chapter 3 discusses the NFC and FEL controllers based on the proposed ANFIS structures. In this section, target functions and NFC structures in the literature are examined. New target function and NFC structures are proposed. Chapter 4 discusses the interval type-2 fuzzy logic systems. Type reduction methods have been reviewed and a new IT2 ANFIS structure is proposed and tested by modifying the KMA type reduction method. Chapter 5 discusses the usage of the proposed IT2 ANFIS structure for control studies. Chapter 6 discusses the quadrotor model. The proposed fuzzy logic control structures (NFC, FEL) in the thesis are tested for quadrotor position and attitude control.

2. MAMDANI ANFIS AND ŞEN ANFIS

This chapter includes training and comparison of three fuzzy logic systems those are Sugeno FLS, Mamdani FLS and Şen FLS. Firstly, the Sugeno ANFIS is examined that is a widely used model in literature. Secondly, Mamdani ANFIS is examined that there are Mamdani ANFIS studies in the literature [12]. But we proposed a new Mamdani ANFIS model to get better results than the Mamdani ANFIS models given in the literature. The proposed Mamdani ANFIS is compared to Sugeno ANFIS. Thirdly, Şen FLS is examined. There is not any Şen ANFIS model in the open literature. We proposed a Şen ANFIS model and its advantages and disadvantages are discussed. The learning algorithms are implemented for a three input nonlinear function. So, the results are compared to each other with equation 2.1 [63].

$$y = 1 + a^{0.5} + b^{-1.5} + sin(c) \tag{2.1}$$

Range is chosen as [0.5, 10] with sample time is 0.5. For the given range [1, 20]x[1, 20]x[1, 20], there are 8000 training data pairs. So, training data matrix is (8000x4). Membership functions for every inputs are chosen as 4 trapezoid membership functions.

The performances of methods for consequent parameters are calculated for GD and LSE. In the following chapters, the training parameters are used in control algorithms. To have less complex and less computational burden, the antecedent parameters' training is not performed. Anyway, the antecedent parameters' training does not change the result much as seen in the Table 2.1. The ANFIS antecedent parameters are defined as given in Figure 2.1. "2.00" states the maximum value of the inputs and "0" states the minimum value of the inputs.

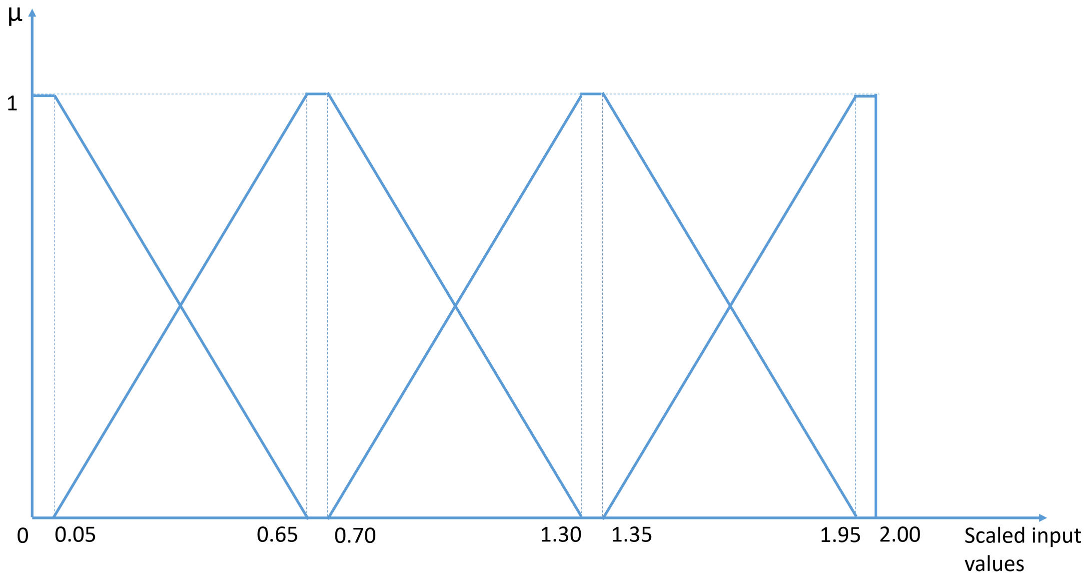

Figure 2.1 : ANFIS antecedent parameters.

2.1 Sugeno FIS

Sugeno FIS has and/or method, implication and defuzzification steps. In Matlab ANFIS toolbox; for and/or operator "and/prod", for implication operator "product" and for defuzzification operator "wtaver" is used. So, the same operators will be used. The ANFIS structure is given in Figure 2.2. The trained Sugeno FLS structure is:

Step 1: Fuzzification;

$$O_i^1 = \mu_{A_i}(x) \qquad O_i^1 = \mu_{B_i}(y) \tag{2.2}$$

Step 2: "and/or" method:

$$O_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = w_i \tag{2.3}$$

Step 3: Implication "product" operator:

$$f_i = p_i * x_1 + q_i * x_2 + r_i \qquad O_i^3 = w_i * f_i \tag{2.4}$$

Step 4: Aggregation "sum" operator:

$$O_i^4 = \sum_{i=1}^{n} O_i^3 \tag{2.5}$$

Step 5: Defuzzification "wtaver" (weighted average) operator:

$$O_i^5 = \frac{O_i^4}{\sum_{i=1}^{n} w_i} \tag{2.6}$$

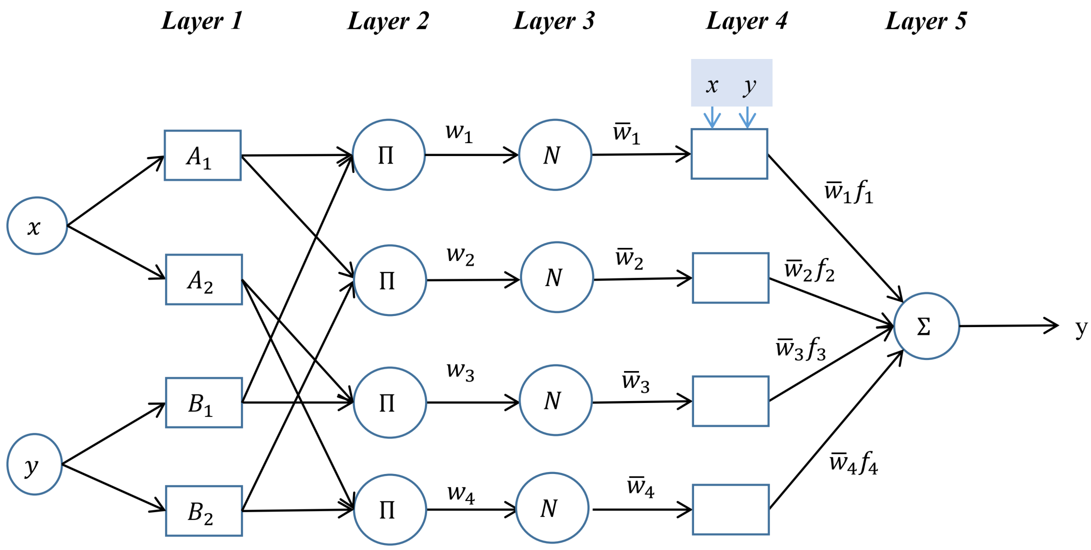

Figure 2.2 : ANFIS structure.

2.2 Sugeno ANFIS

For the given Sugeno FLS structure; the ANFIS structure in the literature is given below:

Layer 1: Membership functions are generated:

$$O_i^1 = \mu_{A_i}(x) \qquad\qquad O_i^1 = \mu_{B_i}(y) \tag{2.7}$$

Layer 2: The membership grades are multiplied with each others:

$$O_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = w_i \tag{2.8}$$

Layer 3: The "wtaver" method is represented and weights are normalized.:

$$O_i^3 = \frac{w_i}{\sum_{i=1}^{n} w_i} \tag{2.9}$$

Layer 4: The rule outputs are calculated:

$$O_i^4 = y_i = \overline{w_i} f_i = \overline{w_i}(p_i x_1 + q_i x_2 + r_i) \tag{2.10}$$

Layer 5: All rule outputs are summed up:

$$O_i^5 = \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \overline{w_i} f_i = (\overline{w}_1 x_1)p_1 + (\overline{w}_1 x_2)q_1 + \overline{w}_1 r_1 + (\overline{w}_2 x_2)p_2 + (\overline{w}_2 x_2)q_2 + \overline{w}_2 r_2 \tag{2.11}$$

In this study, consequent parameters are trained with LSE and GD methods.

2.2.1 ANFIS LSE Method

The LSE method equations are given below as stated in the literature. The equation 2.11 can be written again to use LSE as given below [9]:

$$y = (\overline{w}_1 x_1)p_1 + (\overline{w}_1 x_2)q_1 + \overline{w}_1 r_1 + (\overline{w}_2 x_1)p_2 + (\overline{w}_2 x_2)p_2 + \overline{w}_2 r_2 \tag{2.12}$$

$$y = \begin{bmatrix} \overline{w}_1 x_1 & \overline{w}_1 x_2 & \overline{w}_1 & \overline{w}_2 x_1 & \overline{w}_2 x_2 & \overline{w}_2 \end{bmatrix} \begin{bmatrix} p_1 \\ q_1 \\ r_1 \\ p_2 \\ q_2 \\ r_2 \end{bmatrix} = XW \tag{2.13}$$

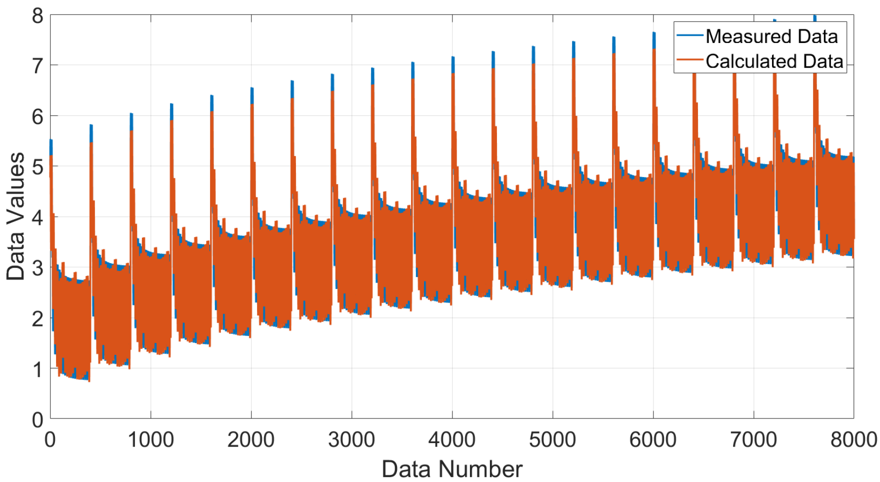

Figure 2.3 : Measured and LSE based ANFIS calculated data.

In the equations, "w" is the weights, "x" is the inputs and "W" is the consequent parameters. The inputs are known and the weights are calculated. The ANFIS trains the consequent parameters.

$$Y = XW \Rightarrow W = X^{-1}Y \tag{2.14}$$

Generally, X does not be a square matrix and so it can not be inverted. Then, pseudo-inverse is used. The trained Sugeno FLS is tested with actual data taken from the equation 2.1 and with a desired sinusoidal trajectory. The results are given in Figure 2.3 and in Figure 2.4.

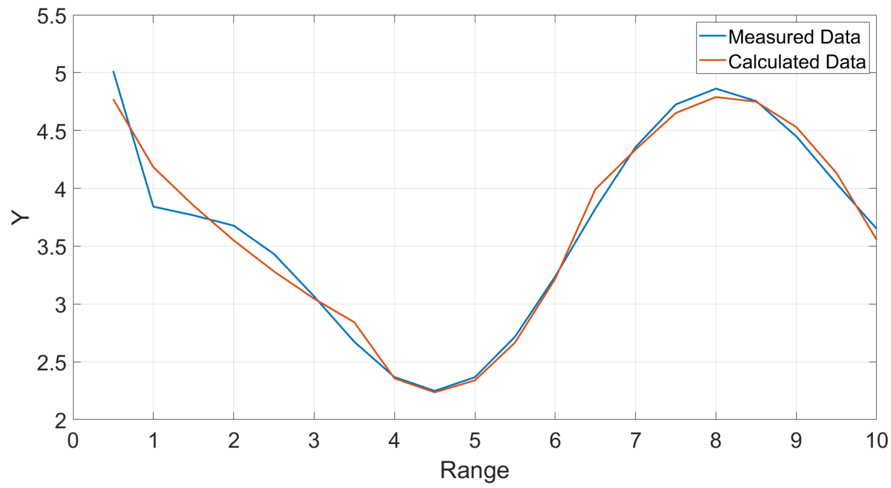

Figure 2.4 : LSE based ANFIS results on a trajectory.

2.2.2 ANFIS Gradient Descent Method

In some cases, the pseudo-inverse of the X matrix can be meaningless and then the LSE method can be ineffective. The GD that is used in online ANFIS controller, does not need to inverse of the X matrix. The GD equations are given below as stated in the literature where "y" is measured output values, "y^t" is calculated output values, "lr" is learning rate and $W = r_i$.

$$E = \frac{1}{2}(y - y^t)^2 \qquad y = XW \tag{2.15}$$

$$r(t+1) = r(t) - lr\frac{\partial E}{\partial r_i} \tag{2.16}$$

$$\frac{\partial E}{\partial r_i} = (y - y^t)X \tag{2.17}$$

$$r(t+1) = r(t) - lr(y - y^t)X \Rightarrow W(t+1) = W(t) - lr(y - y^t)X \tag{2.18}$$

The GD based ANFIS model is trained to create a Sugeno FLS. The trained Sugeno FLS is tested with actual data taken from the equation 2.1 and on a desired sinusoidal trajectory. The results are given in Figures (2.5, 2.6). Similarly, the LSE based ANFIS training with Matlab tool is made and tested with actual data taken from the equation 2.1 on a desired sinusoidal trajectory. The results are given in Figures (2.7, 2.8).

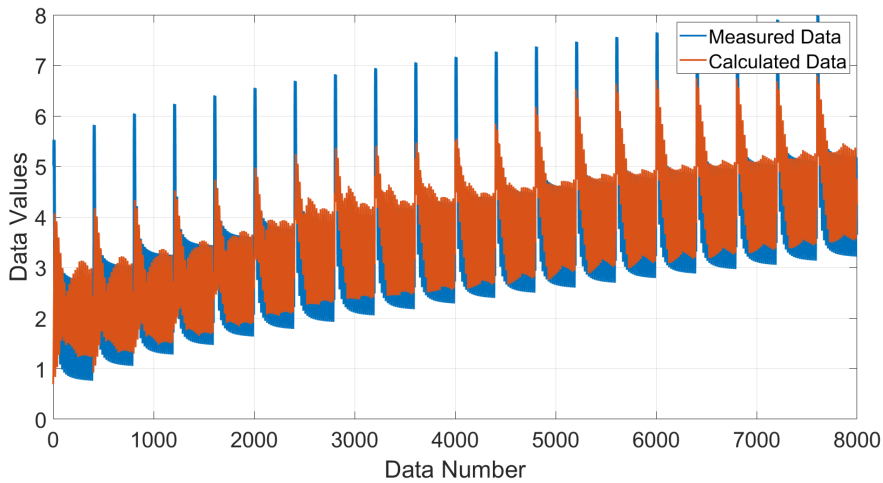

Figure 2.5 : Measured and GD based ANFIS calculated data.

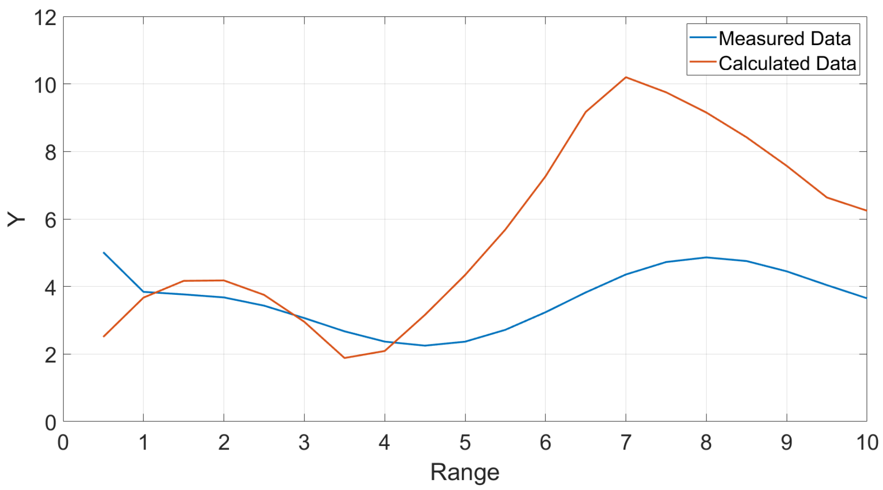

Figure 2.6 : GD based ANFIS results on a trajectory.

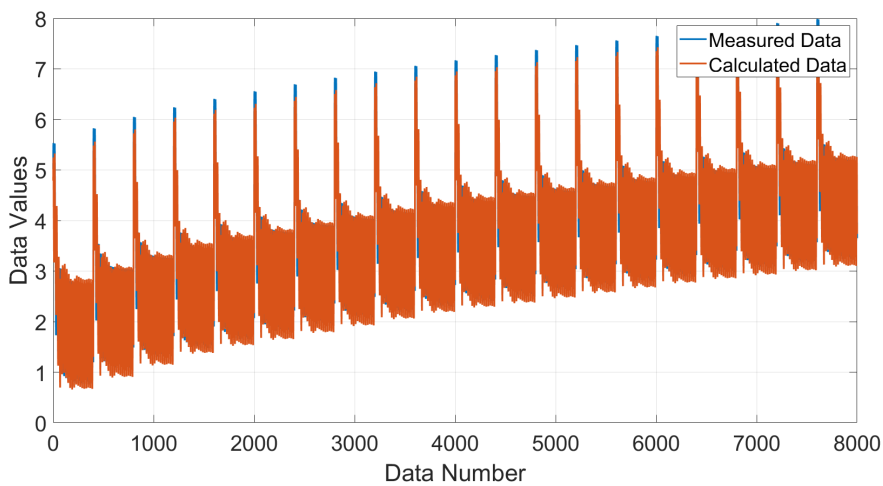

Figure 2.7 : Measured and Matlab ANFIS tool calculated (LSE) data for 40 epochs.

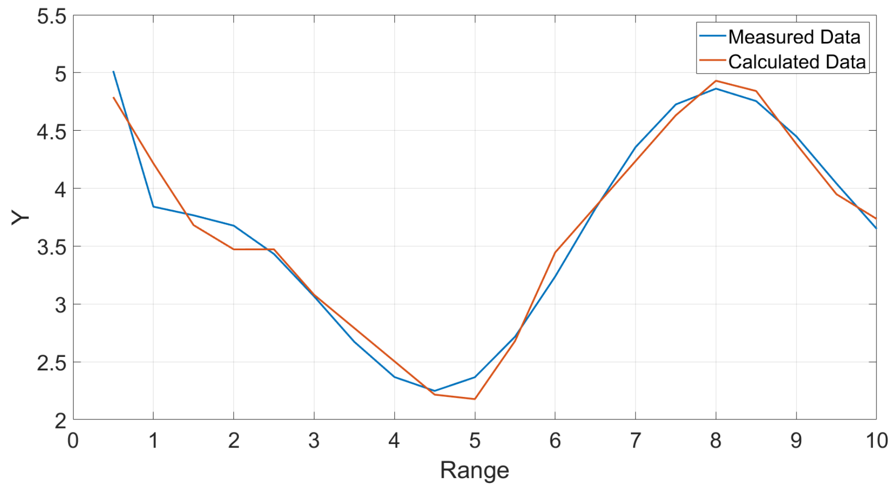

Figure 2.8 : Matlab ANFIS tool (LSE) results on a trajectory for 40 epochs.

Table 2.1 : Comparison of ANFIS training results.

Parameters	Matlab ANFIS Tool LSE	Sugeno LSE	Matlab ANFIS Tool GD	Sugeno GD
Epoch Number	40	1	40	40
Training Error	0.175904	0.155995	2.812876	0.834712

In these ANFIS studies, only consequent parameters are trained. Also, the same training is performed for antecedent and consequent training in Matlab ANFIS toolbox for hybrid learning option. In the ANFIS tool, hybrid option is chosen that the consequent parameters are trained with LSE and the antecedent parameters are trained with back-propagation algorithm. In the Matlab ANFIS tool, back propagation option is chosen that the antecedent and consequent parameters trained with back propagation (GD) algorithms. The results show that only consequent parameters training is effective as two sided training.

2.3 Mamdani FIS

Mamdani FIS antecedent and consequent parameters are fuzzy; however, Sugeno FIS consequent parameters are constant or linear. So, the Mamdani FIS has advantages over Sugeno FIS: The Mamdani FIS is institutional and very compatible with human thought structure [12].

In this section, we proposed a Mamdani ANFIS (MANFIS) approach and it is compared to Sugeno ANFIS for on-line and off-line training. Firstly, a target Mamdani

FIS model must be determined to create a convenient MANFIS model. The chosen Mamdani FIS operator: for and/or operator "and/prod", for implication operator "product", for aggregation operator "sum" and for defuzzification operator "centroid" is used. The selected Mamdani FIS structure is:

Step 1: Fuzzification:

$$O_i^1 = \mu_{A_i}(x) \qquad\qquad O_i^1 = \mu_{B_i}(y) \tag{2.19}$$

Step 2: "and-prod" operator is used:

$$O_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = w_i \tag{2.20}$$

Step 3: Implication; "product" operator is used ("$area_i$" is area of the consequent MFs) :

$$O_i^3 = w_i * area_i = a_i \tag{2.21}$$

Step 4: Aggregation; "sum" operator is used ("z_i" is center of the consequent MFs):

$$O_i^4 = \sum_{i=1}^{n} (a_i z_i) \tag{2.22}$$

Step 5: Defuzzification; "centroid" operator is used:

$$O_i^5 = \frac{\sum_{i=1}^{n} (a_i z_i)}{\sum_{i=1}^{n} a_i} \tag{2.23}$$

2.4 Mamdani ANFIS

The maximum and minimum values of the measured data are divided to rule number. So, the consequent MFs number will be equal to the rule number. The MANFIS structure is proposed as given in Figure 2.9 for the selected Mamdani FIS model.

Layer 1: Membership functions are generated:

$$O_i^1 = \mu_{A_i}(x) \qquad\qquad O_i^1 = \mu_{B_i}(y) \tag{2.24}$$

Layer 2: The "prod" operator is implemented. The membership grades are multiplied with each others:

$$O_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = w_i \tag{2.25}$$

Layer 3: Implication "prod" operator is used. a_i is the consequent MFs area:

$$O_i^3 = w_i * (area) = a_i \tag{2.26}$$

For the consequent parameters training, the area of the consequent MFs are unknown. So, the area must be determined. In this study area is defined as:

$$b_1 = \frac{max(y_{meas}) - min(y_{meas})}{rulenumber} \tag{2.27}$$

$$b_2 = 4 * b_1 \tag{2.28}$$

where b_1 is top line and b_2 is bottom line of the triangular. y_{meas} defines the training output data.

$$area = \frac{b_1 + b_2}{2} \tag{2.29}$$

This area calculation is used in this thesis for MANFIS. For simplicity, the area can be selected as 1.

Layer 4: All areas are normalized:

$$O_i^4 = \frac{a_i}{\sum_{i=1}^{n} a_i} = \overline{a}_i \tag{2.30}$$

Layer 5: Every rule output is calculated:

$$O_i^5 = y_i = z_i\overline{a}_i \tag{2.31}$$

Layer 6: All rule outputs are summed up:

$$O_i^6 = \sum_{i=1}^{n} y_i = \overline{a}_1 z_1 + \overline{a}_2 z_2 + \overline{a}_3 z_3 \ldots \tag{2.32}$$

2.4.1 Mamdani ANFIS LSE Method

Least Square Estimation (LSE) is an effective training algorithm that is used in ANFIS. The inverse of the X matrix must be calculated in LSE training method. This matrix is given in the literature as $[\overline{w}_i \overline{a}_i]$ [12]. This is a $[1, rulenumber]$ vector where the number of columns is one and the number of rows equal to the rule number. Because of that, inverse of the X vector is not precise.

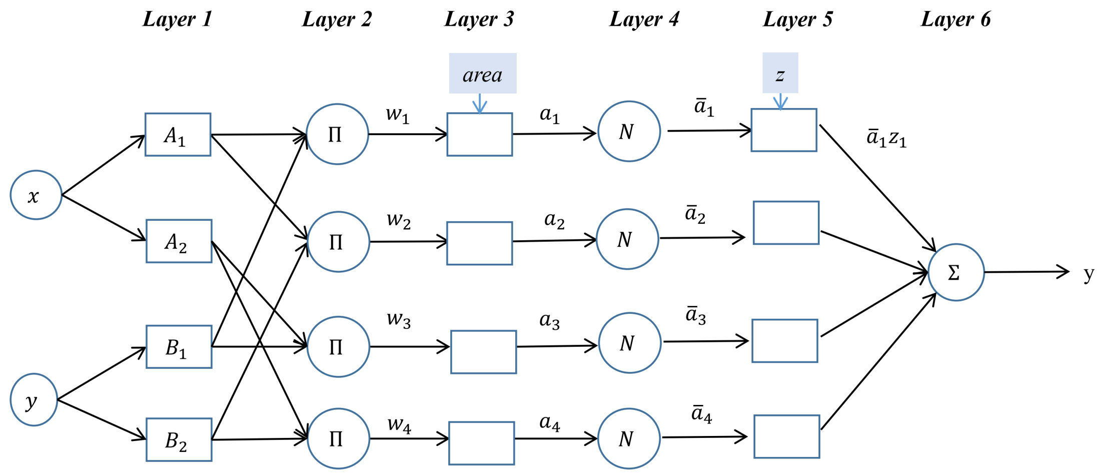

Figure 2.9 : MANFIS structure.

In this study, the Mamdani ANFIS structure is created as $[\bar{a}_i]$ instead of $[\bar{w}_i\bar{a}_i]$ as seen in Figure 2.9. The $[\bar{a}_i]$ matrix is $[rulenumber, datanumber]$. So, the X matrix can be calculated more precisely than previous version given in the literature [13]. The Mamdani ANFIS LSE equations are given below:

$$y = \begin{bmatrix} \bar{a}_1 & \bar{a}_2 & \bar{a}_3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = XW \tag{2.33}$$

"X" is triggered MF areas and "W" is the consequent parameters. The triggered membership functions and areas are known but the consequent parameters are not known. So the consequent parameters will be trained.

$$Y = XW \Rightarrow W = X^{-1}Y \tag{2.34}$$

The trained Mamdani FLS is tested with actual data taken from the equation 2.1 and with a desired sinusoidal trajectory. The results are given in Figure 2.10 and in Figure 2.11.

2.4.2 Mamdani ANFIS Gradient Descent Method

The Gradient Descent (GD) method is adapted to MANFIS structure. The equations are given below where "y" is measured output values, "y^t" is calculated output values, "lr" is learning rate and $W = r_i$.

$$E = \frac{1}{2}(y - y^t)^2 \qquad y = z_i\bar{a}_i \tag{2.35}$$

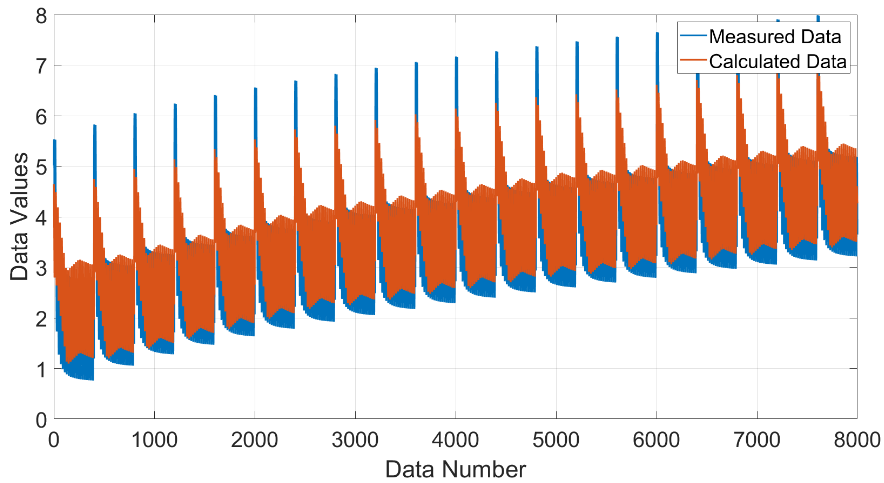

Figure 2.10 : Measured and LSE based MANFIS calculated data.

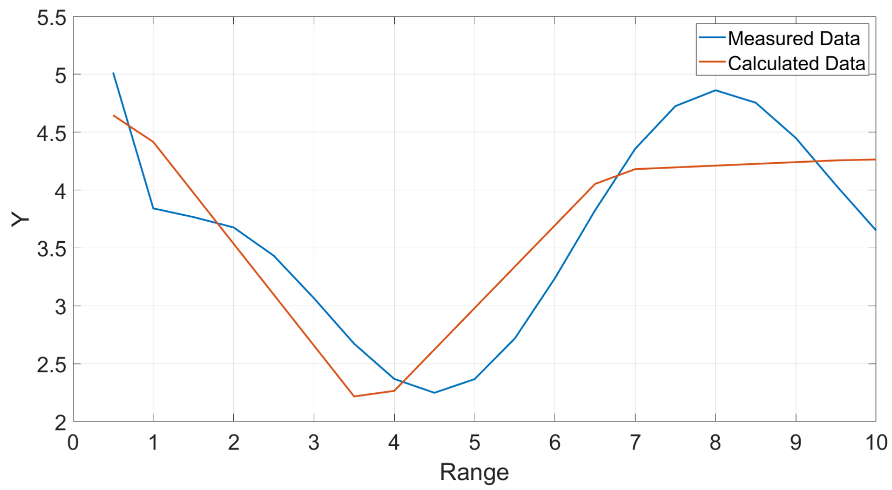

Figure 2.11 : LSE based MANFIS results on a trajectory.

$$z(t+1) = z(t) - lr\frac{\partial E}{\partial z_i} \tag{2.36}$$

$$\frac{\partial E}{\partial z_i} = (y - y^t)\bar{a}_i \tag{2.37}$$

$$z(t+1) = z(t) - lr(y - y^t)\bar{a}_i \tag{2.38}$$

The trained Mamdani FIS is tested with actual data that was produced with equation 2.1 and on a sinusoidal desired trajectory. The selected Mamdani FIS is trained for 100 epochs.

The Gradient Descent based Mamdani ANFIS training gives faster reaction than GD based Sugeno ANFIS training. It is clear that the GD based MANFIS results have less error when compared to Gradient based ANFIS. The results are given in Figures (2.12, 2.13).

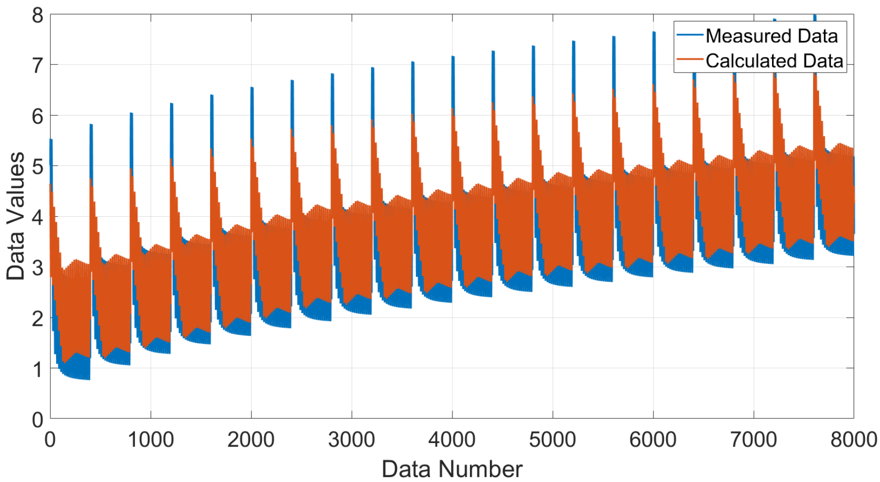

Figure 2.12 : Measured and GD based MANFIS calculated data.

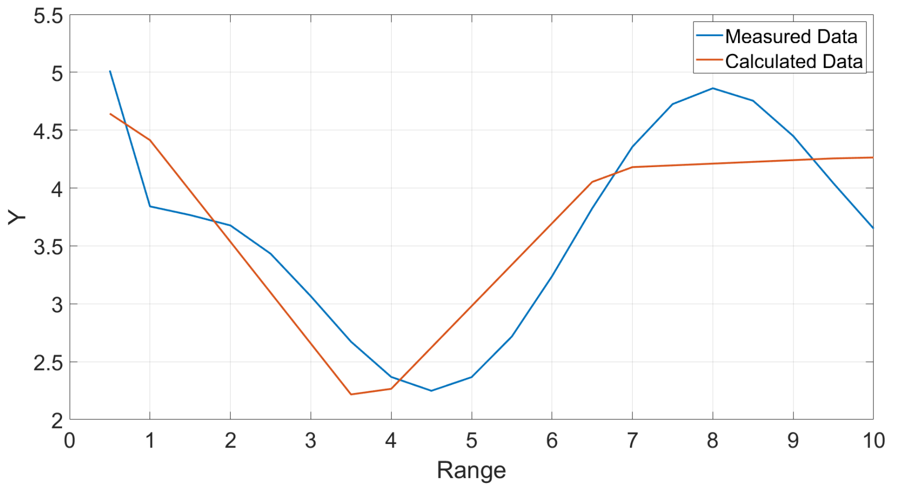

Figure 2.13 : Gradient based MANFIS results on a trajectory.

2.5 Şen FIS

Şen FIS is proposed as SISO system for estimation of solar irradiation from sunshine duration in 1988 [14]. The Şen FIS antecedent and consequent parameters are fuzzy. The Şen FIS structure is given in Figure 2.14.

The first two steps are same for Mamdani, Sugeno and Şen FIS. In the third step, α and z variables are calculated and multiplied with the rule weights [64]. This step is different from other fuzzy inference systems. The selected Şen FIS structure for training is given below:

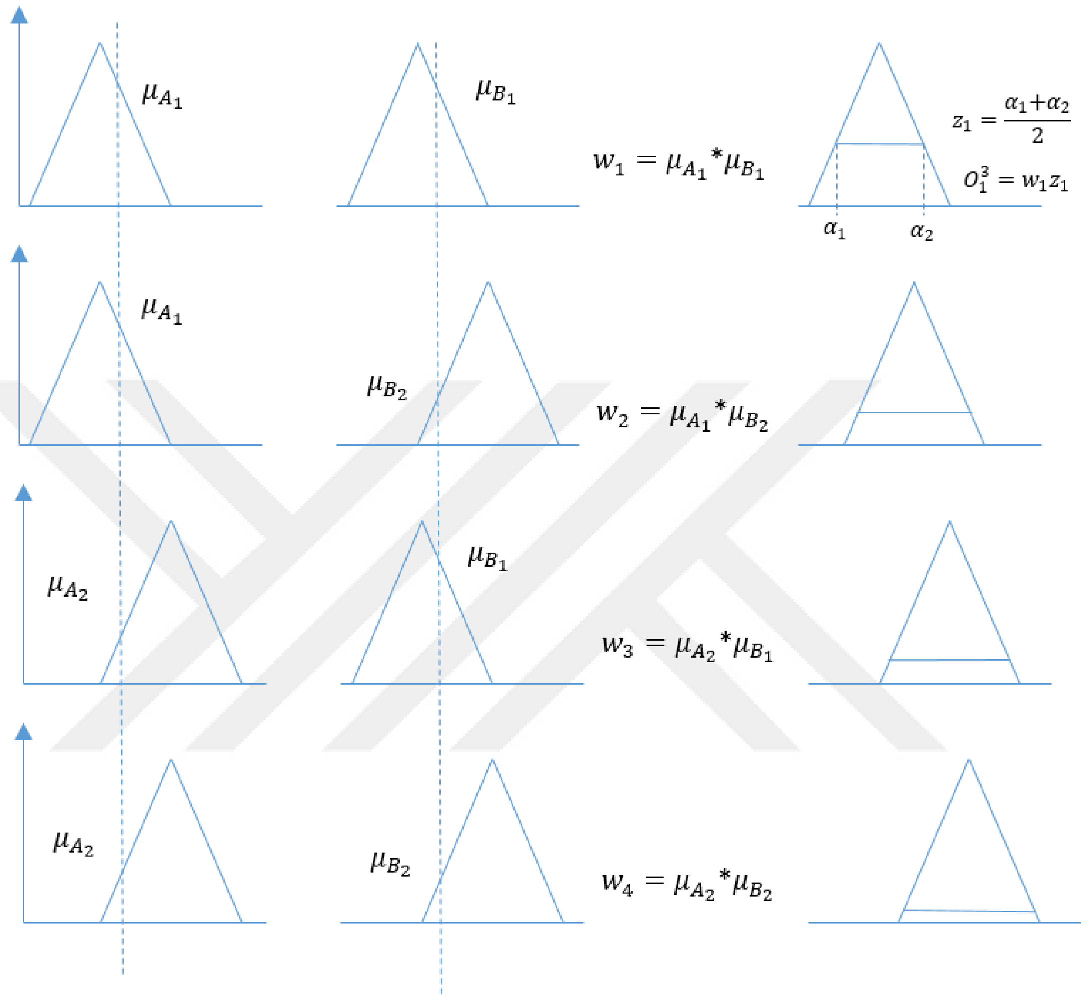

Figure 2.14 : Şen FIS structure.

Step 1: Fuzzification;

$$O_i^1 = \mu_{A_i}(x) \qquad O_i^1 = \mu_{B_i}(y) \tag{2.39}$$

Step 2: "and-prod" operator:

$$O_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = w_i \tag{2.40}$$

Step 3: Implication; "product" operator is used where z is center value of consequent membership functions for symmetric MFs:

$$O_i^3 = w_i * z_i \qquad z_i = \frac{\alpha_1 + \alpha_2}{2} \tag{2.41}$$

Step 4: Aggregation; all rule outputs are summed up:

$$O_i^4 = \sum_{i=1}^{n} O_i^3 \tag{2.42}$$

Step 5: Defuzzification; "wtaver" (weighted average) operator is used:

$$O_i^5 = \frac{O_i^4}{\sum_{i=1}^{n} w_i} \tag{2.43}$$

2.6 Şen ANFIS

As mentioned before, the Mamdani and Şen FIS antecedent and consequent parameters are fuzzy; however, the Sugeno FIS consequent parameters are linear or constant. So, Mamdani and Şen FIS have advantages on Sugeno model: they are heuristic and very compatible to human thought structure [12].

In this work, a new Şen ANFIS structure is proposed with GD and LSE methods respectively. The proposed method is given in the following sections and they are tested on a nonlinear equation 2.1. The proposed Şen ANFIS training structure is generated as given in Figure 2.15 for the selected Şen FIS model.

Layer 1: Membership functions are generated:

$$O_i^1 = \mu_{A_i}(x) \qquad\qquad O_i^1 = \mu_{B_i}(y) \tag{2.44}$$

Layer 2: The "prod" operator is implemented. The membership grades are multiplied with each others:

$$O_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = w_i \tag{2.45}$$

Layer 3: The weights are normalized:

$$O_i^3 = \frac{w_i}{\sum_{i=1}^{n} w_i} = \overline{w}_i \tag{2.46}$$

Layer 4: The rule outputs are calculated:

$$O_i^4 = y_i = \overline{w}_i z_i \tag{2.47}$$

Layer 5: All rule outputs are summed up:

$$O_i^5 = \sum_{i=1}^{n} \overline{w}_i z_i \tag{2.48}$$

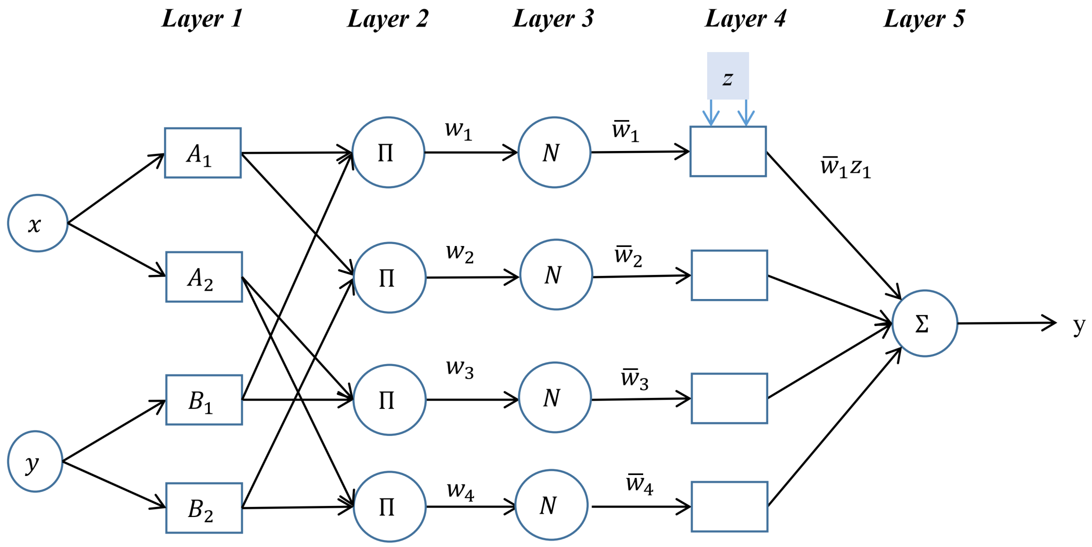

Figure 2.15 : SANFIS structure.

2.6.1 Şen ANFIS LSE Method

To implement LSE to proposed Şen ANFIS, the X and W matrices must be defined as given below:

$$y = \begin{bmatrix} \overline{w}_1 & \overline{w}_2 & \overline{w}_3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = XW \tag{2.49}$$

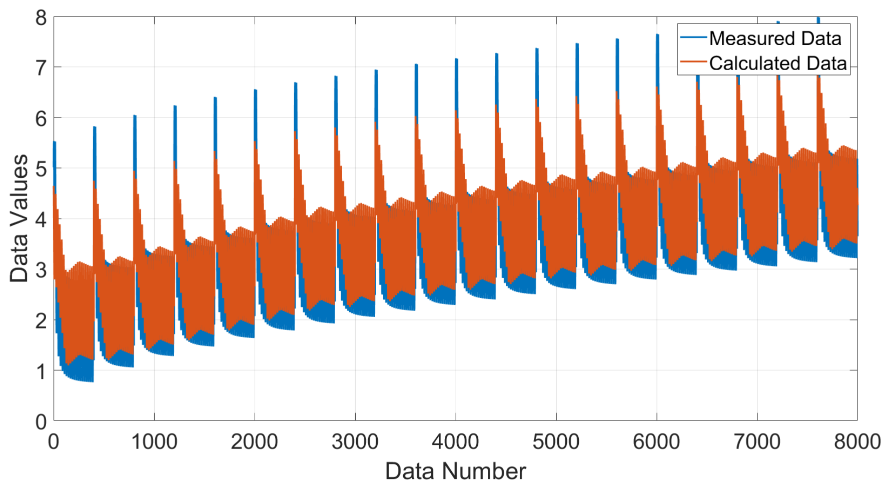

Figure 2.16 : Measured and LSE based Şen ANFIS calculated data.

"X" is normalized membership functions and "W" is the consequent parameters. The consequent parameter calculation is given below as:

$$Y = XW \Rightarrow W = X^{-1}Y \tag{2.50}$$

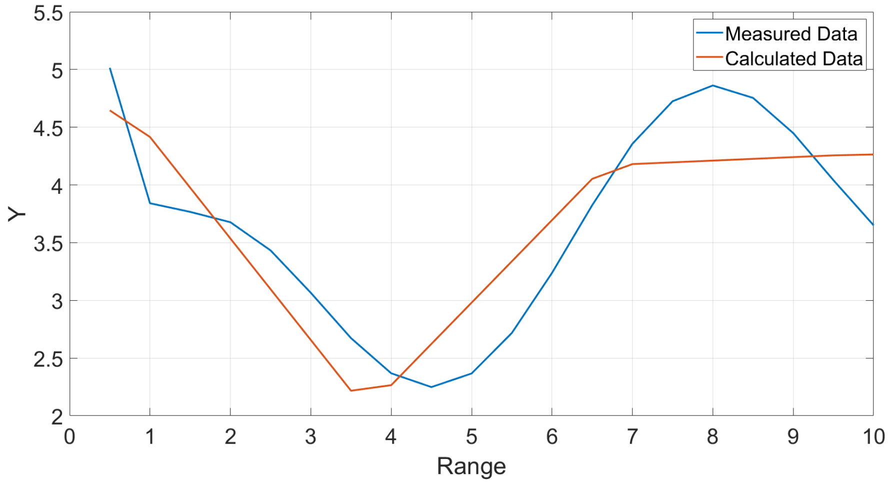

Figure 2.17 : LSE based Şen ANFIS results on a trajectory.

The trained Şen FLS is tested with actual data taken from the equation 2.1 and with a desired sinusoidal trajectory. The results are given in Figure 2.16 and in Figure 2.17.

2.6.2 Şen ANFIS Gradient Descent Method

The Gradient Descent (GD) method is adapted to Şen ANFIS structure. The equations are given below where "y" is measured output values, "y^t" is calculated output values, "lr" is learning rate. Firstly, error is defined as:

$$E = \frac{1}{2}(y - y^t)^2 \qquad y = z_i \overline{w}_i \tag{2.51}$$

$$z(t+1) = z(t) - lr\frac{\partial E}{\partial z_i} \tag{2.52}$$

$$\frac{\partial E}{\partial z_i} = (y - y^t)\overline{w}_i \tag{2.53}$$

$$z(t+1) = z(t) - lr(y - y^t)\overline{w}_i \tag{2.54}$$

The trained Şen FIS is tested with actual data taken from the equation 2.1 and with a desired sinusoidal trajectory.The results are given in Figure 2.18 and in Figure 2.19. The result figures show that the three ANFIS models work regularly. While the best

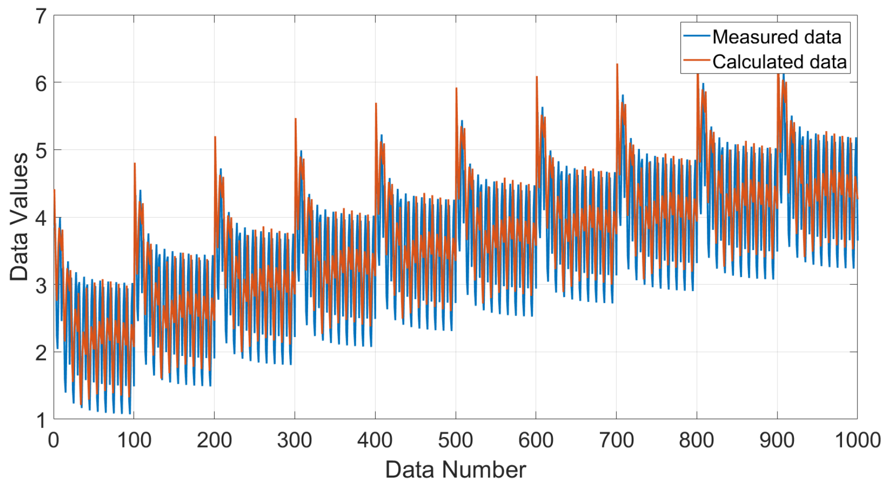

Figure 2.18 : Measured and GD based Şen ANFIS calculated data.

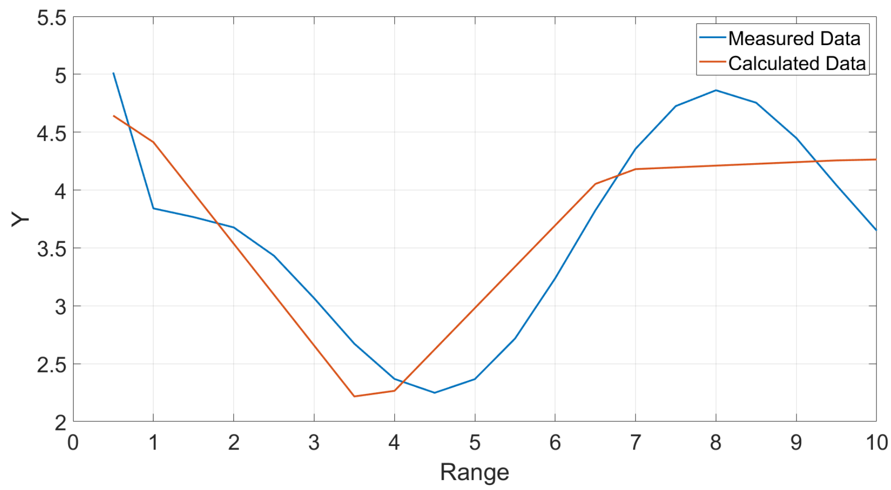

Figure 2.19 : GD based Şen ANFIS results on a trajectory.

results are taken from the LSE based Sugeno ANFIS, the worst results are taken from the GD based Sugeno ANFIS models.

2.7 Comparison of Methods

The contribution of this chapter is the proposed Mamdani ANFIS and ŞEN ANFIS models. The Şen FIS has been redesigned to be a multi input system. Afterwards, Şen ANFIS is proposed for defined multi input Şen FIS. The Sugeno ANFIS and proposed Mamdani ANFIS and Şen ANFIS structures are trained for the equation 2.1. The results are given in Table 2.2 that the iteration numbers are chosen same for the three fuzzy inference systems. All training parameters are selected as the same for an accurate comparison.

Table 2.2 : Comparison of ANFIS methods.

Parameters	Sugeno LSE	Sugeno GD	Mamdani LSE	Mamdani GD	Şen LSE	Şen GD
Epoch Number	1	40	1	40	1	40
Training Error	0.155995	0.834712	0.604405	0.645085	0.604405	0.604405
Computation Time (s)	0.6674	3.6053	0.0761	10.4652	0.0916	1.0860

For the LSE training method, the Sugeno ANFIS training error is the smallest one. However, Sugeno ANFIS needs more computation time than Mamdani and Şen LSE based ANFIS training. So, for off-line training studies, the Sugeno ANFIS is the best.

For the GD training method, the Sugeno ANFIS training error is higher than other FLSs. Mamdani and Şen based ANFIS training errors is same for GD and LSE algorithms. However, GD based Mamdani ANFIS needs higher computation time than other FLSs. The computation time is very important for control studies.

Şen ANFIS is better than Mamdani ANFIS at computation time and Şen ANFIS is better than Sugeno ANFIS at training error and computation time. So, implementation of the proposed Şen ANFIS for control studies is more effective compared to Mamdani ANFIS and Sugeno ANFIS.

3. NEURO FUZZY CONTROL SYSTEMS

Fuzzy Logic Controller (FLC) can work lonely or can tune any conventional controller parameters [19,27]. In these studies, the designer determines FLC parameters and so these systems work perfectly for defined models. However, any unexpected change in the model or environment may adversely affect the FLC results. So, tuning of FLC parameters is a widely used approach in the literature. For example, lonely Type-1 Neuro-fuzzy Controller (NFC) was tested in [30], and FLC with Sliding Mode Control (SMC) theory based training was tested in [58,59]. All of these studies are simulation studies. For 2 DOF helicopter actual systems, the FLC model was constructed by using Francis-Isidori-Byrnes (FIB) differential equation [22].

In this chapter, we focus on NFC structures proposed in the literature are examined and tried to be developed. For every loop, the training model changes the FLC parameters values and tries to reduce errors. In this chapter, the training algorithm is constructed only for consequent parameters as given in [28] by using Gradient Descent Method. Target function which is used in NFC parameters training is examined [28]. In addition to the target function given in the literature, different target functions have also been tested.

The NFC structures in the literature are viewed and we proposed new NFC structures. These existing and proposed structures are compared and the most appropriate one is decided. The most suitable NFC structure is used for Sugeno, Mamdani and Şen FIS. The chosen NFC structure is used for type-2 FLC in the following chapters.

3.1 DC Servo Motor Dynamics

The NFC is tested in a simulation for a DC motor. In this example, the input of the system is voltage and the output is rotational velocity. The physical parameters are given in Table 3.1.

Table 3.1 : DC motor physical parameters.

	Parameters	Values
J	moment of inertia of the rotor	0.01 kgm^2
b	motor viscous friction constant	0.1 Nms
Ke	electromotive force constant	0.01 $V/rad/sec$
Kt	motor torque constant	0.01 Nm/Amp
R	electric resistance	1 Ohm
L	electric inductance	0.5 H

The tested DC motor equation is:

$$J\frac{\partial^2\theta}{\partial t^2} + b\frac{\partial\theta}{\partial t} = Ki \tag{3.1}$$

$$L\frac{\partial i}{\partial t} + Ri = V - K\frac{\partial\theta}{\partial t} \tag{3.2}$$

For the given parameter values a feedback system was prepared as seen in Figure 3.1.

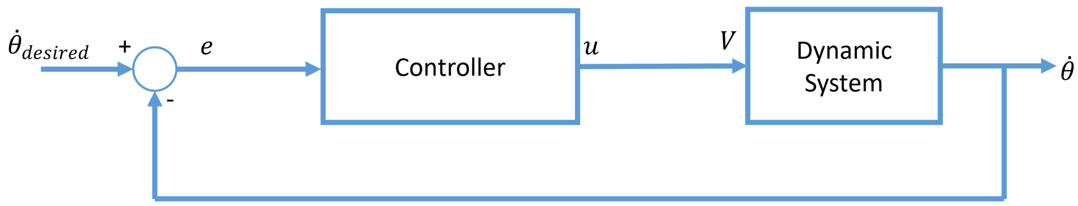

Figure 3.1 : Kp=1 Dc motor proportional control structure.

3.2 Neuro Fuzzy Controller Methods (Sugeno)

NFC is a combination of FLC and Neural Network (NN) with defining a target function. There are different implementations of NFC structures. In this section, these structures are tested and compared to each other. In this chapter, different NFC structures have been prepared and compared to each others. All the Neuro-Fuzzy Controllers have two stages. The first stage is tuning and the second is FLC.

In the literature, Fuzzy Logic antecedent parameters training is done by back-propagation (BP) method. This method imposes highly mathematical computational burden and can sometimes take the antecedent MF parameters out of acceptable limits. For this reason, as a result of the tests for the antecedent parameters, the standard antecedent MF parameters are determined as seen in Figure 3.2. These values are used as the antecedent MF parameters of all NFC structures and studies in this paper. In this way, the system is avoided from high values caused by the derivative terms in the BP algorithm used for antecedent parameters training.

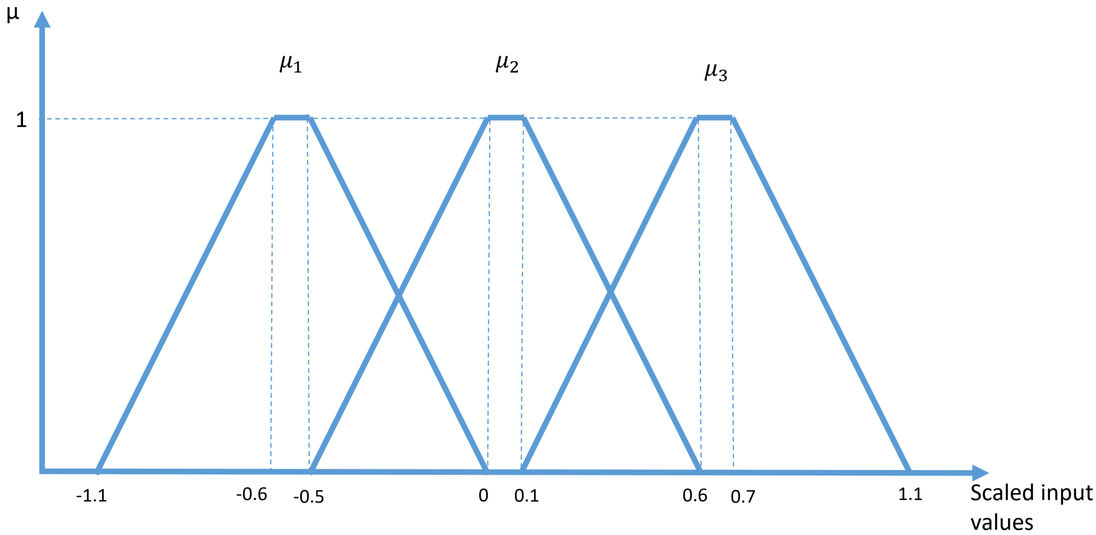

Figure 3.2 : NFC input membership function parameters.

3.2.1 NFC with two inputs (normalized error and output derivative)

NFC is designed for a DC motor with two inputs in [28]. As mentioned before, the target for DC motor control is to reach desired rotational velocity. To saturate fuzzy logic antecedent parameters, the controller inputs are chosen as normalized error and rotational acceleration. So normalized error fuzzy logic MFs can be chosen -1 to 1. Similarly, the rotational acceleration can be chosen in a small gap. The NFC structure is given in Figure 3.3.

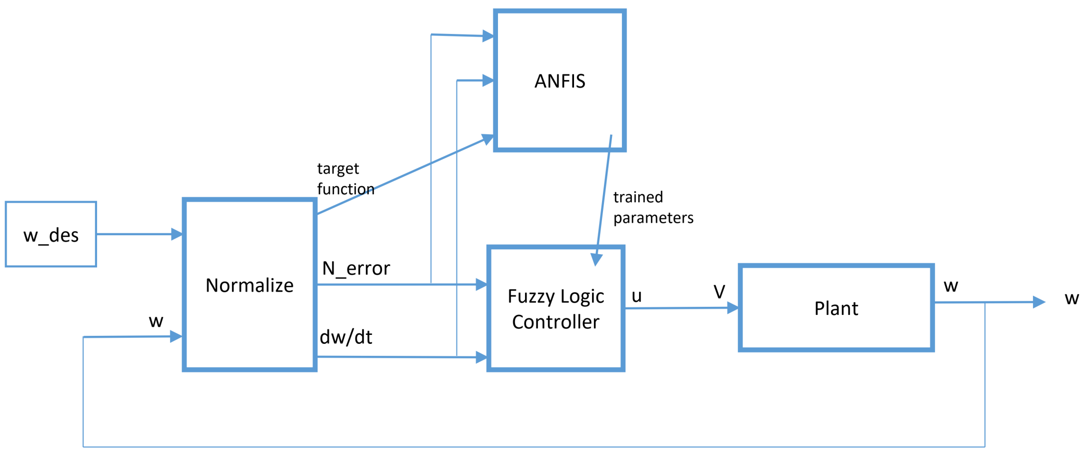

Figure 3.3 : Neuro-Fuzzy controller structure.

$$N_{error} = \frac{w_{des} - w}{w_{des}} \tag{3.3}$$

$$a(rad/s^2) = \frac{w(n) - w(n-1)}{dt} \tag{3.4}$$

where dt is sampling time, w(n) is present angular velocity and w(n-1) is past angular velocity. In this example, fuzzy logic block contains linear Sugeno FIS. The Sugeno consequent parameters are trained with ANFIS block.

3.2.1.1 Tuning Methods

As it is seen from equation 2.11 the ANFIS output is $y = O_i^5$. But a target function must be defined to implement the FLC training algorithm. In the DC servo motor example, the target function is chosen with respect to acceleration.

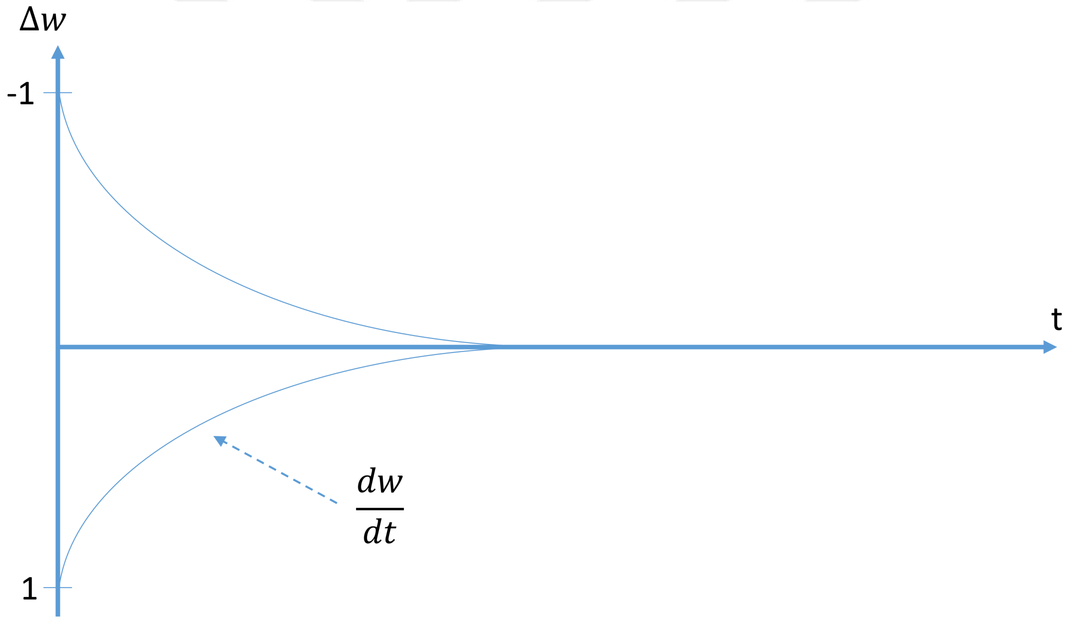

Figure 3.4 : Target function.

In this control example, main target is minimizing the error term. To decrease the error term, the desired velocity and the real velocity must be the same. So, the normalized error will be zero and the change in velocity will be zero. Related drawing is given in Figure 3.4 [28]. In literature [28], the given target equation is offered as:

$$y(cal) = (1 - exp(\frac{-w^2}{2*0.01^2}))*1000*sign(w) \tag{3.5}$$

The ANFIS will try to reach target value for acceleration. When we define desired speed slope as (y) and actual speed slope as (dw/dt) the ANFIS training difference is given by

$$E = 0.5 * (y(cal) - \frac{dw}{dt})^2 \tag{3.6}$$

As remembered from equation 2.16 the update law is:

$$W(n) = W(n-1) - lr\frac{\partial E}{\partial W} \tag{3.7}$$

From the equation 2.15:

$$y(cal) = XW \tag{3.8}$$

$$E = 0.5 * (XW - y(target))^2 \Rightarrow \frac{\partial E}{\partial W} = (XW - y(target)) * X \tag{3.9}$$

$$W(n) = W(n-1) - lr(y(cal) - y(target)) * X \tag{3.10}$$

However, the target function that is given in equation 3.5 is logically false. Because, while the error is zero or very small, physically, actual acceleration must be very small. So, the target function value must be very small. At the same time, desired value (w) can be very high. As a result, the target value "y" can be able to very small for big w values. However, in equation 3.5, the target value will be big. Because of that, we have generated some different target equations:

$$y = 1 - exp(w_{des} - w) \tag{3.11}$$

$$y = w_{des} - w \tag{3.12}$$

Equation 3.11; gives the target function versus error as stated in Figure 3.5. The results for different desired angle velocities are given in Figure 3.6 and Figure 3.7. It is seen from the Figure 3.5 that the NFC gives harder reaction to positive errors than negative errors. So, the response time becomes less effective at positive errors.

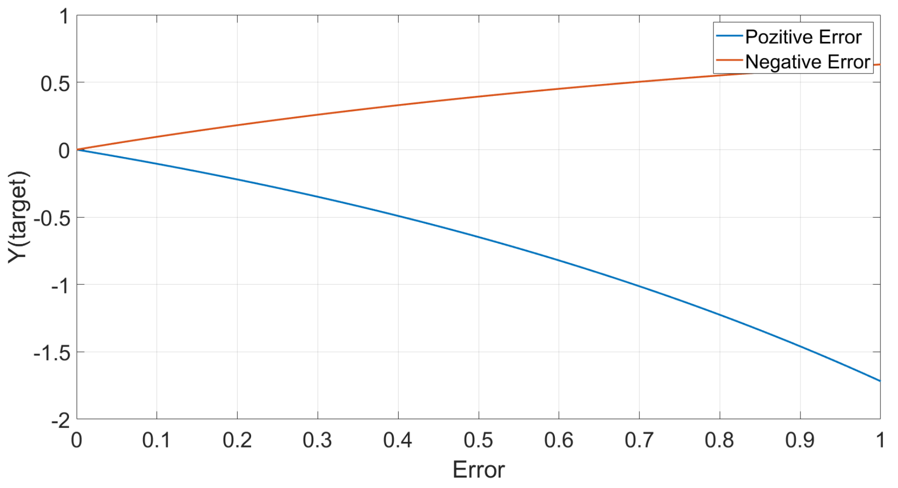

Figure 3.5 : Target function for equation 3.11.

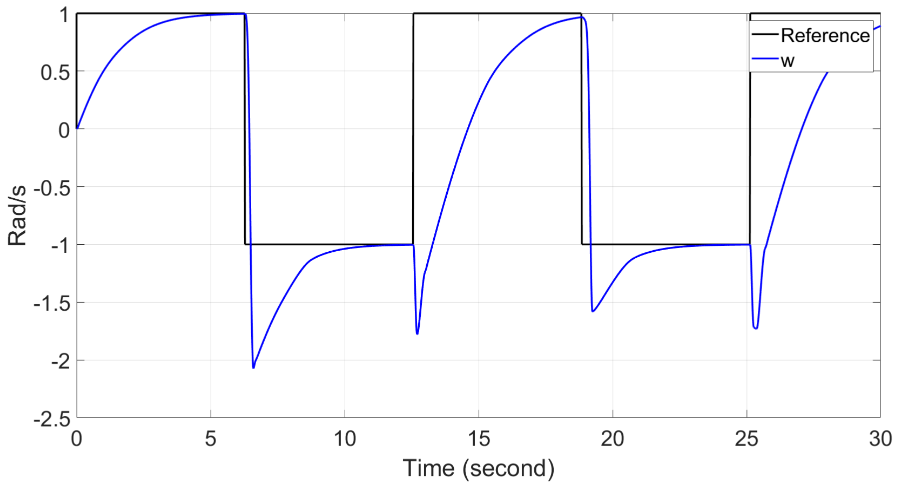

Figure 3.6 : Equation 3.11 target function result for w=1.

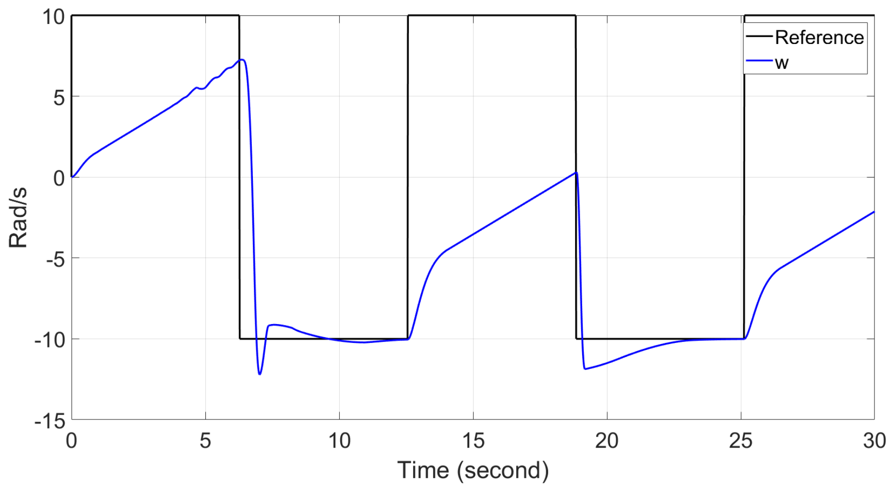

Figure 3.7 : Equation 3.11 target function result for w=10.

When the figures are examined it is seen that the answers for positive direction are very small. However, in the negative direction, the controller response becomes very fast because of the target function characteristics.

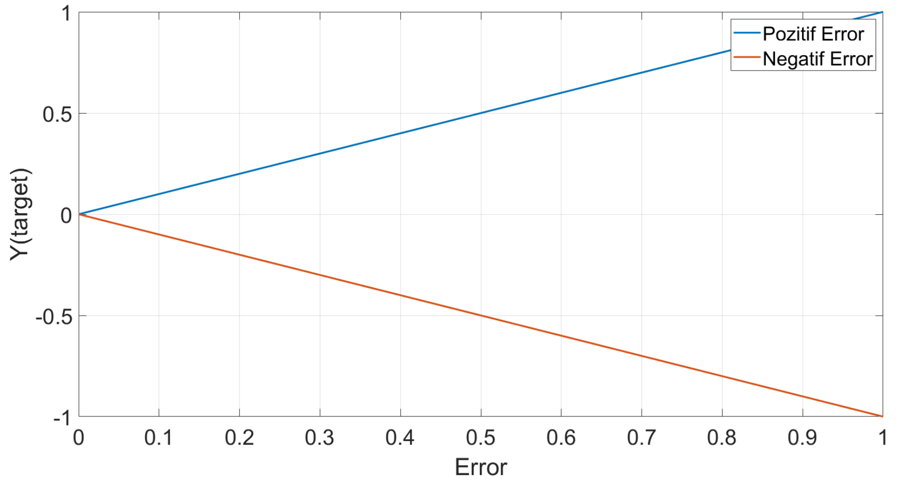

Figure 3.8 : Target function for equation 3.12.

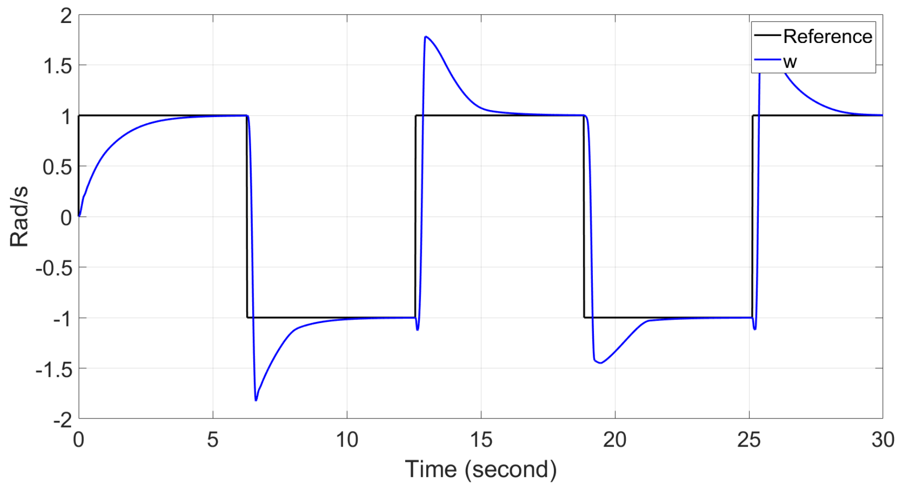

Figure 3.9 : Equation 3.12 target function result for w=1.

Equation 3.12; is used for the same studies that the results can be seen in Figure (3.8, 3.9, 3.10). As seen from the figures the symmetric target function give better results. Of course, some other target functions can be tested for better results.

It is seen from the figures that the target function with only error term is better than other target function definitions.

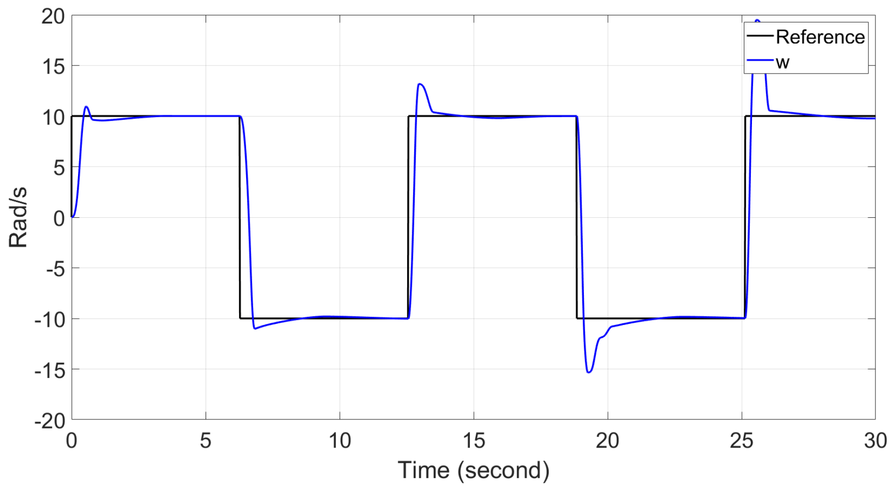

Figure 3.10 : Equation 3.12 target function result for w=10.

3.2.2 NFC with three inputs

There are different NFC implementations. In the previous heading, the two inputs NFC controllers are discussed. Target function was chosen as error. In this section, three inputs NFC controllers will be discussed and compared to each others.

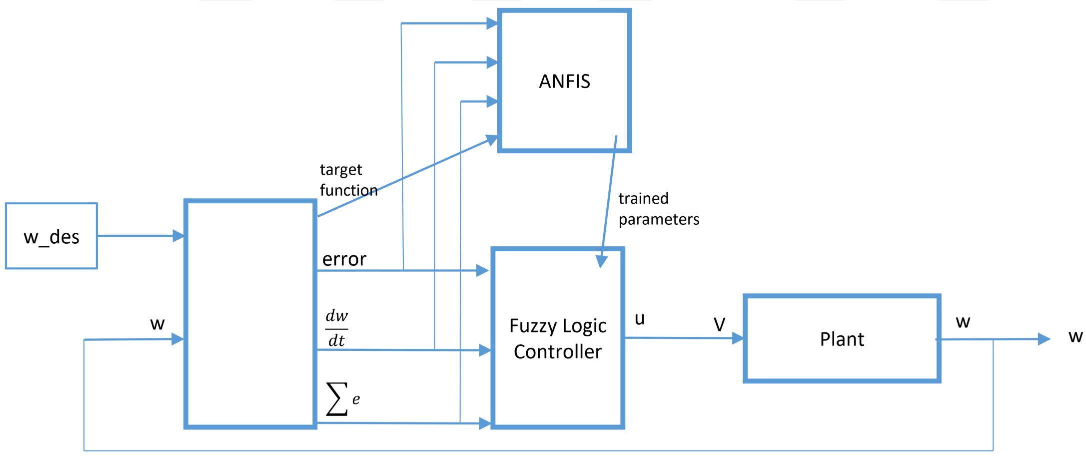

Figure 3.11 : Neuro-Fuzzy controller structure for three inputs.

In 2010, a different NFC method with three inputs (error, error derivative, sum of error) are proposed for DC servo motor [29]. However, it is trained and determined that the NFC, with the defined inputs, does not work regularly. Because of that different combinations are tested.

3.2.2.1 NFC with three inputs (error, output derivative and sum of error)

The controller inputs are selected as error, output derivative and sum of errors. The NFC structure is given in Figure 3.11. Target function is used as given in equation 3.12. The results are given in Figures (3.12, 3.13).

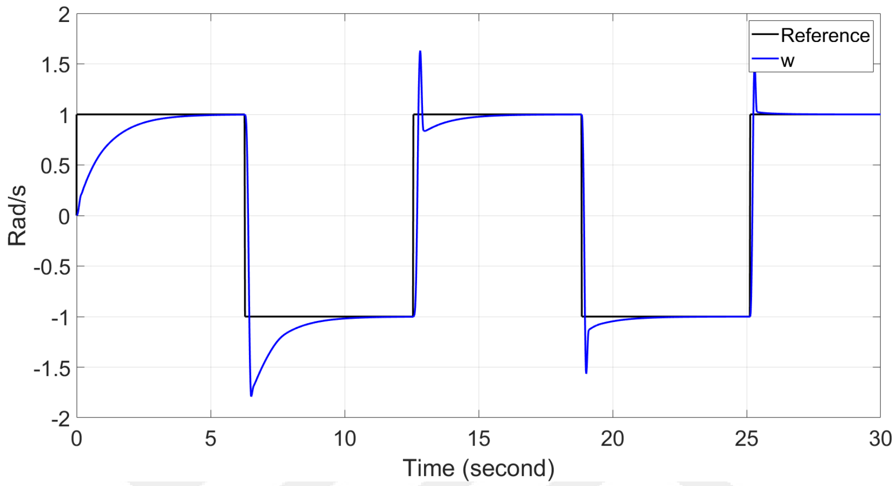

Figure 3.12 : Sugeno NFC results for three inputs (error, output derivative, sum of error) where w=1.

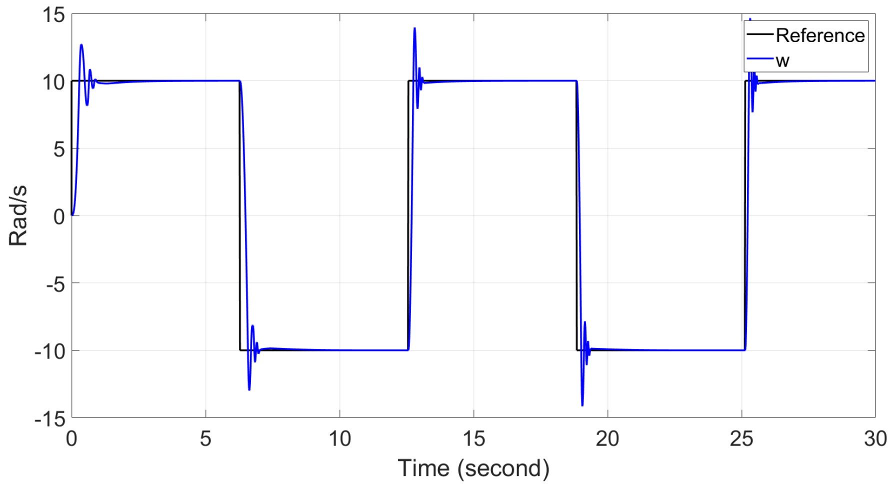

Figure 3.13 : Sugeno NFC results for three inputs (error, output derivative, sum of error) where w=10.

3.2.2.2 NFC with three inputs (error, output derivative and integral of error)

The controller inputs are selected as error, output derivative and integral of errors. The results are given in Figures (3.15, 3.16).

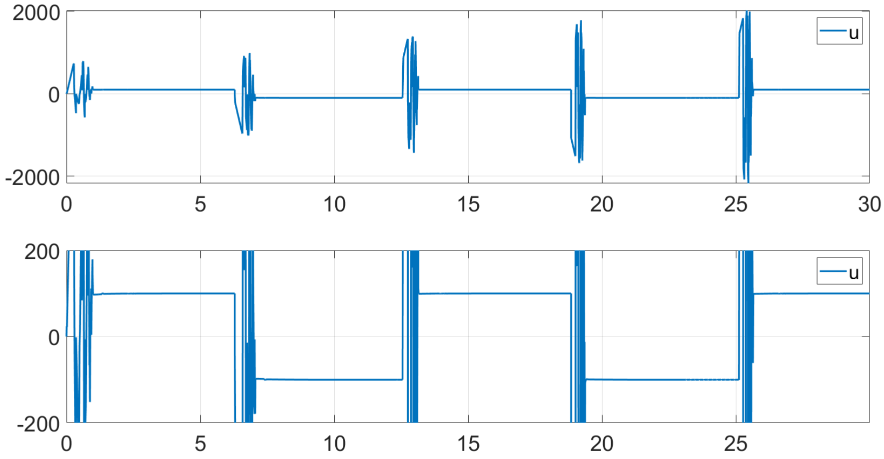

Figure 3.14 : Sugeno FIS NFC control signals for inputs (error, output derivative, sum of error) where w=10 (The figure below is zoomed in.).

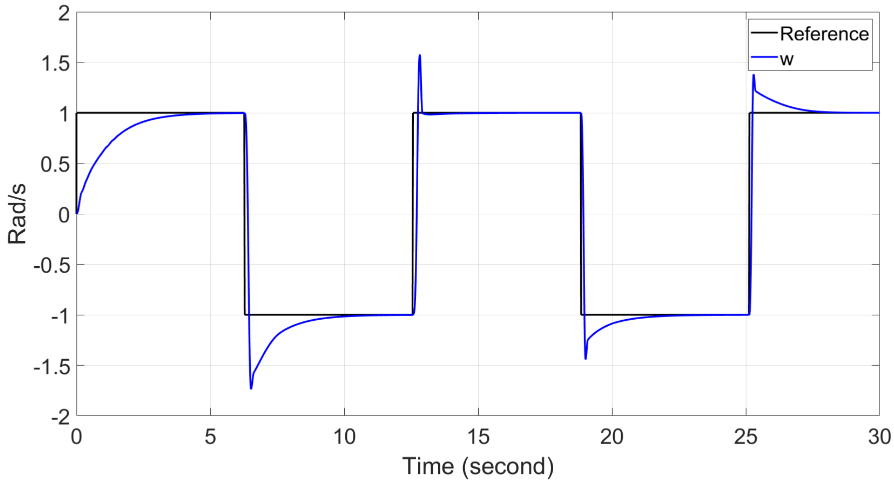

Figure 3.15 : Sugeno NFC results for three inputs (error, output derivative, integral of error) where w=1.

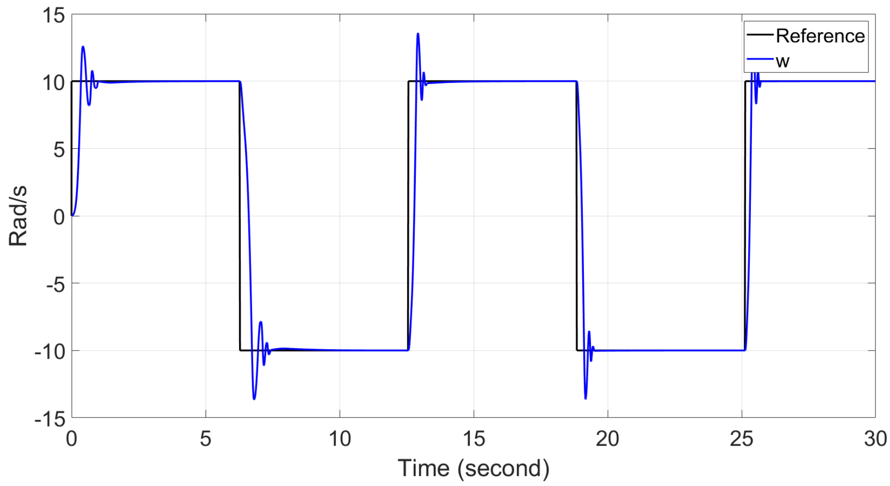

Figure 3.16 : Sugeno NFC results for three inputs (error, output derivative, integral of error) where w=10.

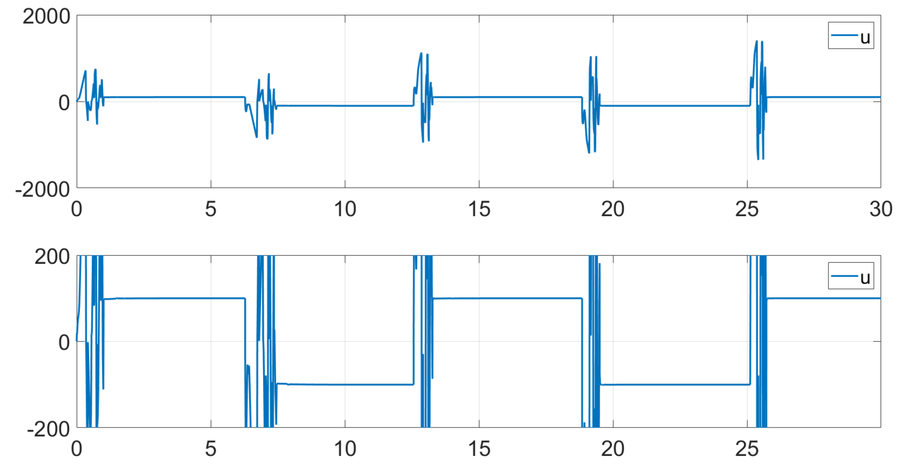

Figure 3.17 : Sugeno FIS NFC control signals for inputs (error, output derivative, integral of error) where w=10 (The figure below is zoomed in.).

3.2.2.3 NFC with three inputs (Normalized error, output derivative and sum of error)

The controller inputs are selected as normalized error, output derivative and sum of errors. The results are given in Figures (3.18, 3.19).

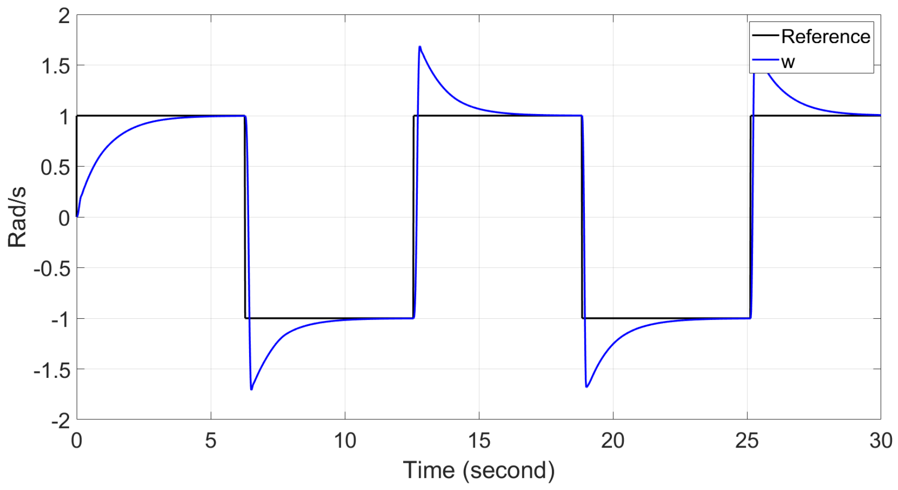

Figure 3.18 : Sugeno NFC results for three inputs (normalized error, output derivative, sum of error) where w=1.

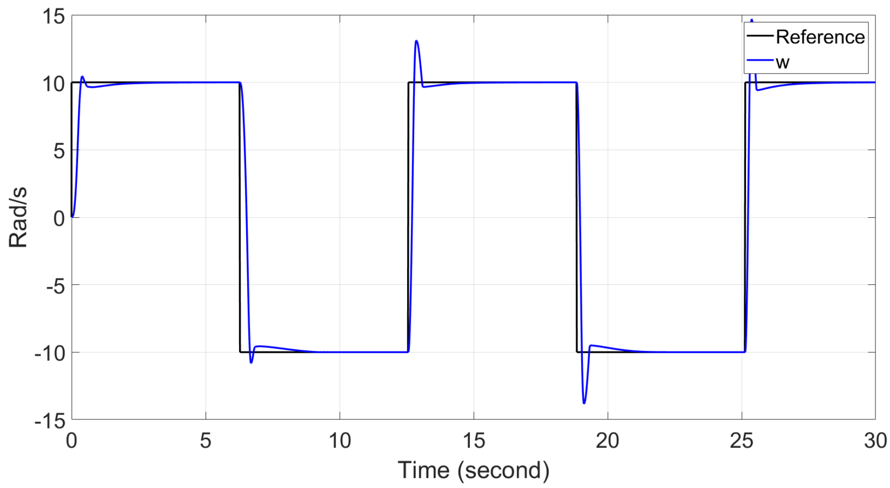

Figure 3.19 : Sugeno NFC results for three inputs (normalized error, output derivative, sum of error) where w=10.

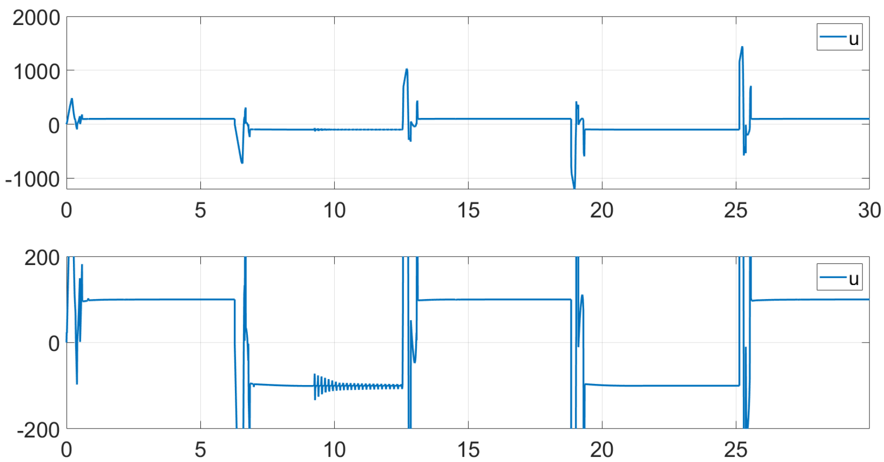

Figure 3.20 : Sugeno NFC control signals for inputs (Normalized error, output derivative, sum of error) where w=10 (The figure below is zoomed in.).

3.2.2.4 NFC with three inputs (Normalized error, output derivative and integral of error)

The controller inputs are selected as normalized error, output derivative and integral of errors. The results are given in Figures (3.21, 3.22).

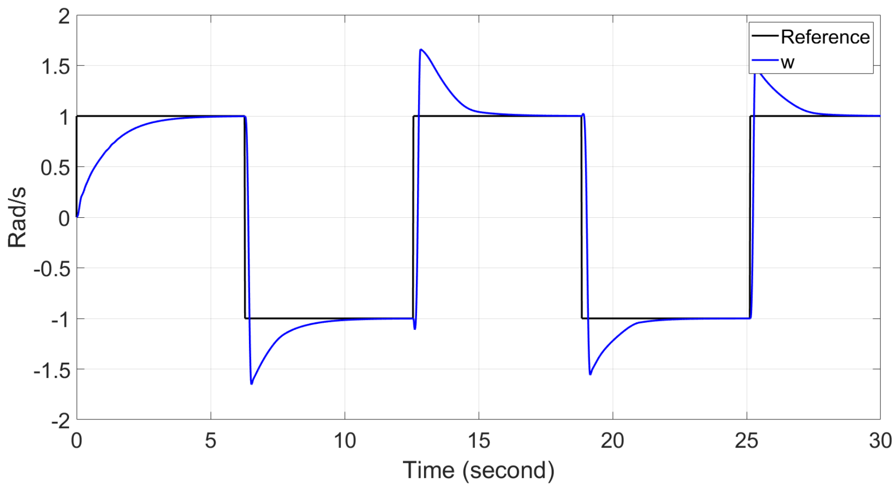

Figure 3.21 : Sugeno NFC results for three inputs (normalized error, output derivative, integral of error) where w=1.

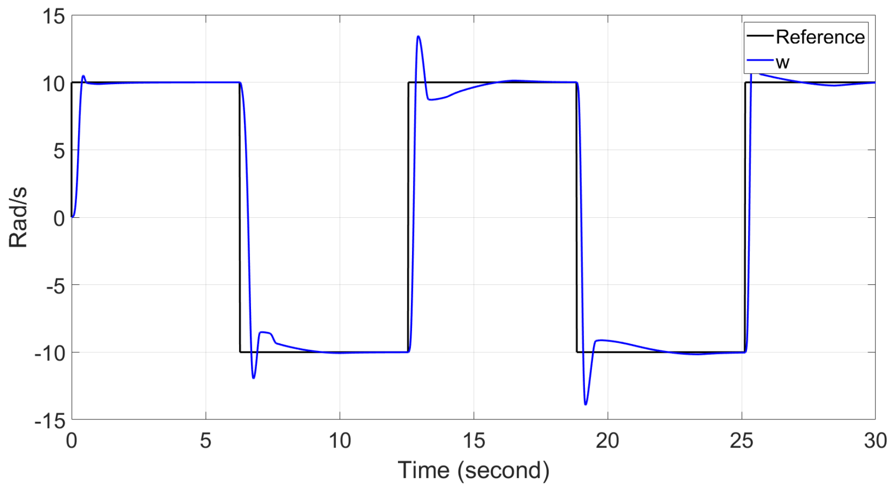

Figure 3.22 : Sugeno NFC results for three inputs (normalized error, output derivative, integral of error) where w=10.

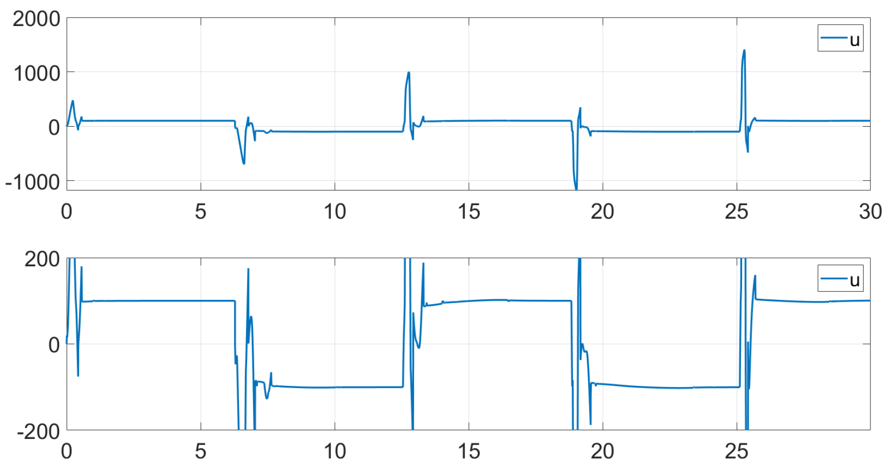

Figure 3.23 : Sugeno FIS NFC control signals for inputs (Normalized error, output derivative, integral of error) where w=10 (The figure below is zoomed in.).

3.2.3 Comparison of Sugeno NFC Methods

The methods are trained and it is seen from the results that target function with symmetric structure is better than others. So, the symmetric target function results are compared to each others. All the values are examined in Table 3.2 for $w_{desired} = 10$ and first 10 seconds.

Table 3.2 : Comparison of sugeno NFCs for w=10.

Inputs	%OS	T_s	RMSE	Computation Time (second)
Normalized error, output derivative	9.6	2.25	3.2788	19.8910
Error, output derivative, sum of error	27.1	1.39	0.2815	47.3358
Error, output derivative, integral of error	5.3	1.86	0.3252	37.4145
Normalized error, output derivative, sum of error	4.6	1.44	2.6404	42.8820
Normalized error, output derivative, integral of error	12.2	2.04	2.9121	35.1141

As seen from Table 3.2, three inputs NFC is better with overshoot, settling time and Root Mean Square Error (RMSE). Summing errors gives better results than integrating the errors at settling time and RMSE. When the figures are checked, normalizing the errors gives higher RMSE results than direct error usage.

As it is seen from Table 3.2, (error, output derivative and sum of error) the NFC inputs combination is the best at RMSE and settling time. However, it has very much overshoot. However, (normalized error, output derivative and sum of error) the NFC inputs combination has the least overshoot. So, in comparing to ŞEN and Mamdani FIS, the (error, output derivative and sum of error) and (normalized error, output derivative and sum of error) NFC inputs combinations will be used.

3.3 Neuro Fuzzy Controller Methods (Mamdani)

The three inputs NFC combinations; (error, output derivative and sum of error) and (normalized error, output derivative and sum of error) are tested for Mamdani based NFC. The same learning rate is used $(lr = 0.07)$ that was used in Sugeno NFC. All conditions are the same.

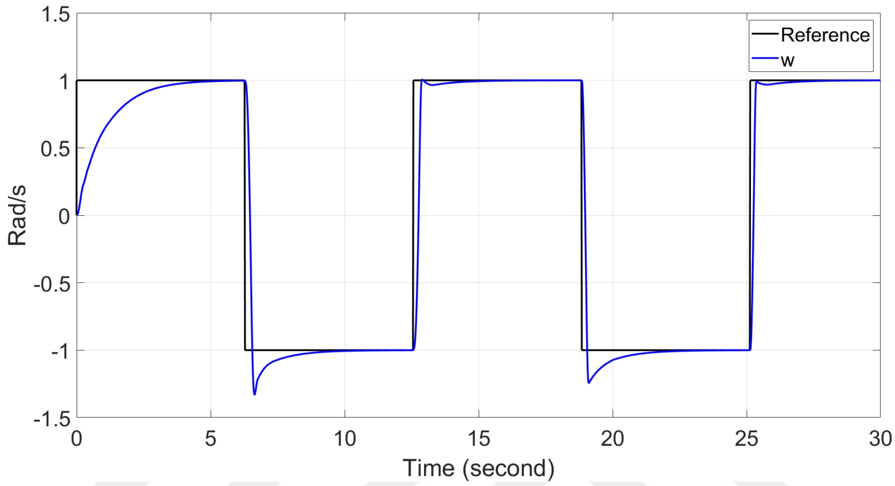

Figure 3.24 : Mamdani NFC results for inputs (error, output derivative, sum of error) where w=1.

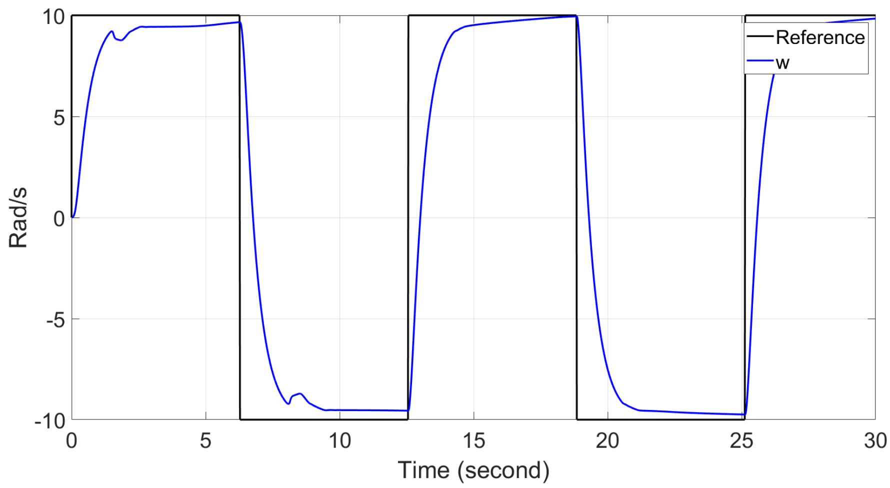

Figure 3.25 : Mamdani NFC results for inputs (error, output derivative, sum of error) where w=10.

The MANFIS structure is given in chapter 1. The given MANFIS structure works regularly for offline mode. However, in online mode, the output parameters must be

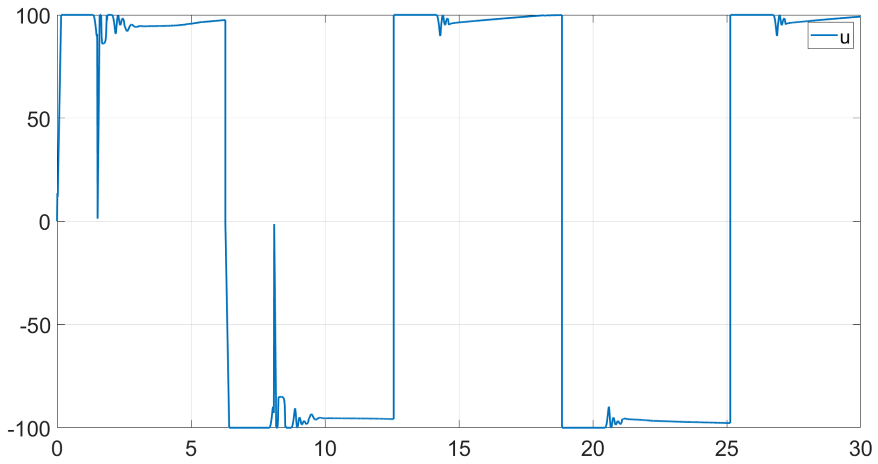

Figure 3.26 : Mamdani NFC control signals for inputs (error, output derivative, sum of error) where w=10.

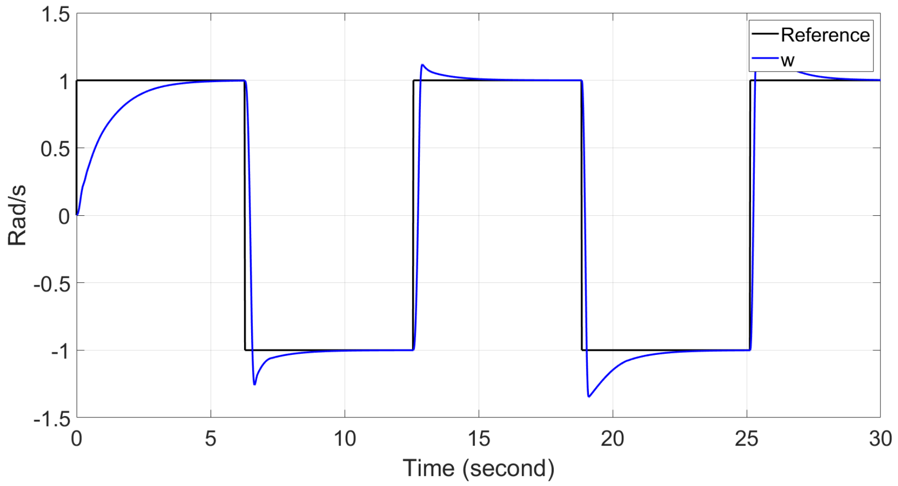

Figure 3.27 : Mamdani NFC results for inputs (normalized error, output derivative, sum of error) where w=1.

trained for an input value for every epoch. Generally, this leads to non-proportional consequent parameters that can lead the system to "Not a Number (NaN)" error.

In Mamdani codes, the consequent parameters maximum and minimum values are divided into 100 pieces as stated in Chapter 2. In irregular output MFs, sometimes, the divided parts can't trigger the any of consequent MFs. So, the controller does not work regularly. Because of that, firstly, it is defined in the controller that if the values go to NaN because of dividing to zero, the value is taken as zero. Afterward, as a more precise solution, the pieces are increased from 100 to 1000. So, the system works very properly. The "z" values that were given in Chapter 2, are restricted from -100 to 100.

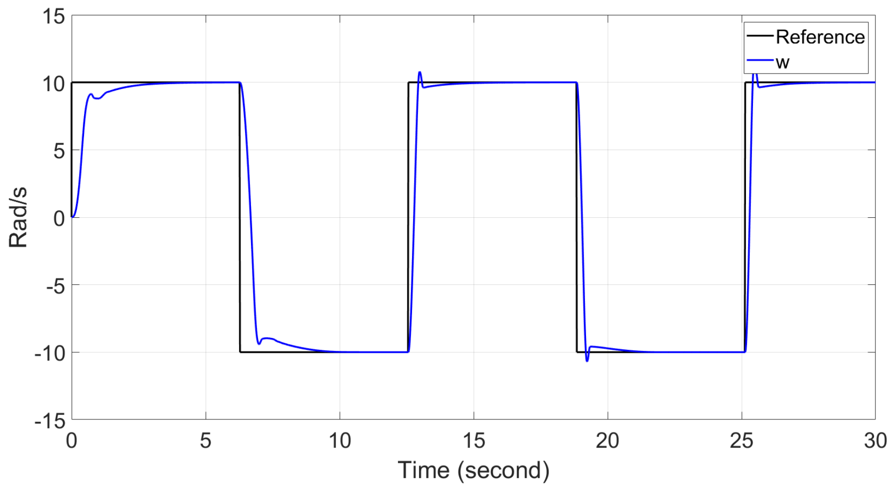

Figure 3.28 : Mamdani NFC results for inputs (normalized error, output derivative, sum of error) where w=10.

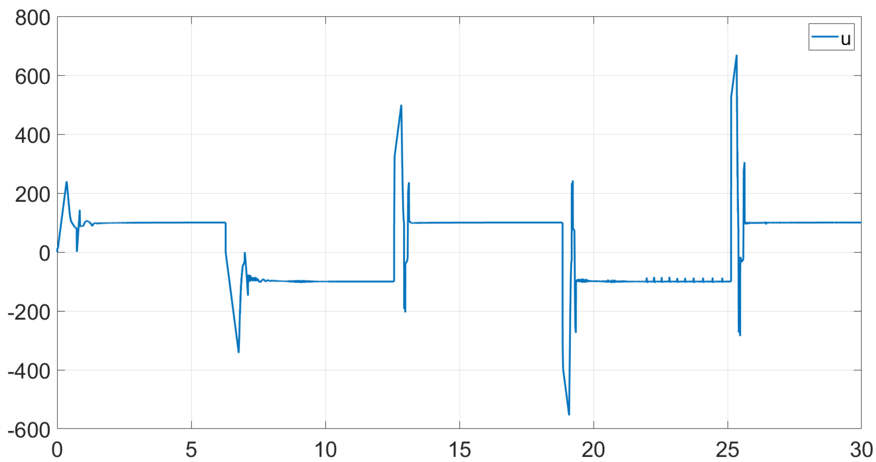

Figure 3.29 : Mamdani NFC control signals for inputs (normalized error, output derivative, sum of error) where w=10.

When the result figures are examined, it is seen that the Mamdani based NFC works regularly with very small overshoots when compared to Sugeno NFC. However, because of the complexity of the Mamdani structure, Mamdani based NFC needs more computational burden than Sugeno based NFC.

3.4 Neuro Fuzzy Controller Methods (Şen)

The (error, output derivative and sum of error) and (normalized error, output derivative and sum of error) three inputs NFC combinations are tested for Şen based NFC. The same learning rate is used $(lr = 0.07)$ that was used in Sugeno and Mamdani NFCs. All conditions are chosen as same.

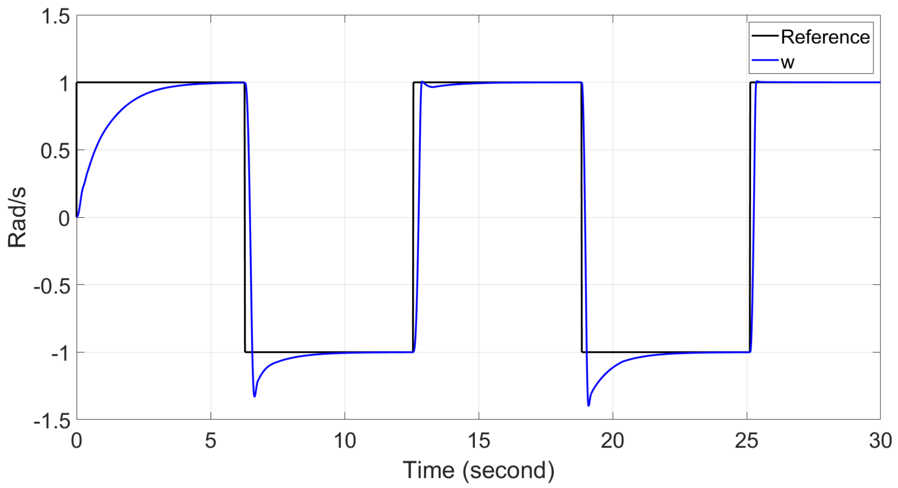

Figure 3.30 : Şen FIS NFC results for inputs (error, output derivative, sum of error) where w=1.

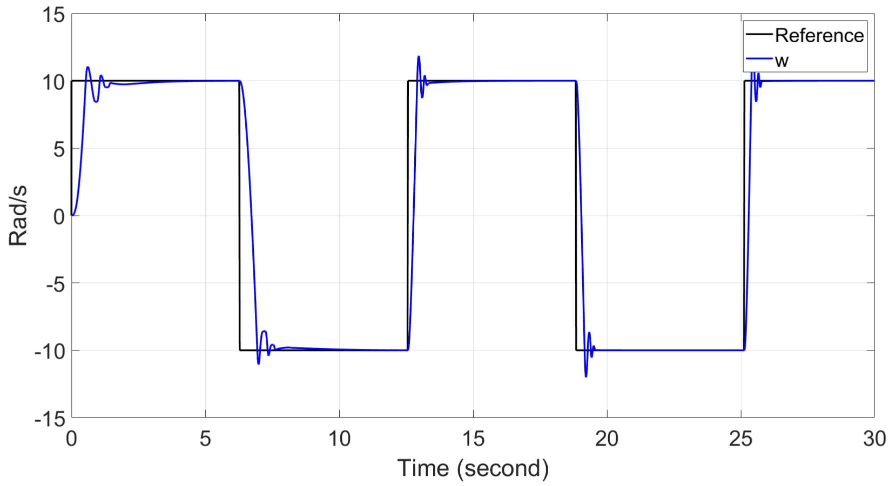

Figure 3.31 : Şen FIS NFC results for inputs (error, output derivative, sum of error) where w=10.

When the result figures are examined, it is seen that the Şen FIS based NFC works regularly with very small overshoots when compared to Sugeno NFC and has the same results as seen in Mamdani NFC. Besides, it needs less computational burden than Mamdani NFC.

As stated in previous chapter, the training algorithms of Mamdani and Şen FISs are given. The training algorithms are different but structurally they are similar. Because, the both of them is human-like systems and has similar structures. Because of that, as seen from Figures (3.30, 3.31, 3.33, 3.34) the Mamdani and Şen FLCs results are similar. The Figure (3.31) and Figure (3.25) are explicitly different. This dissimilarity

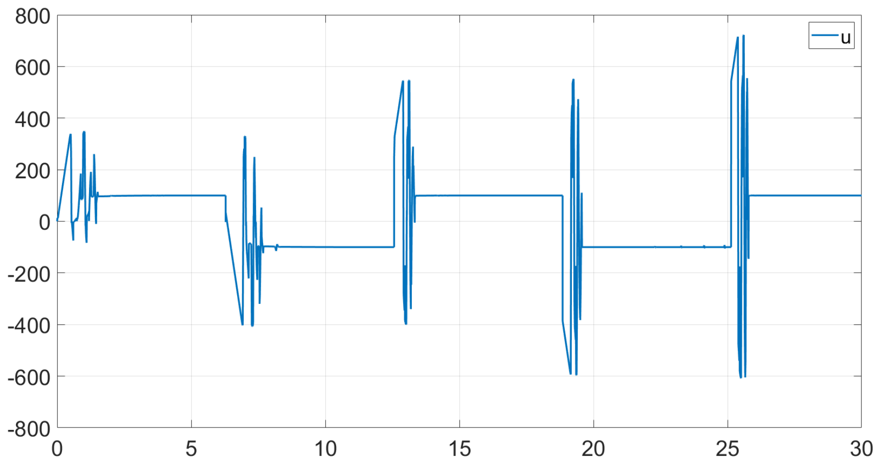

Figure 3.32 : Şen FIS NFC control signals for inputs (error, output derivative, sum of error) where w=10.

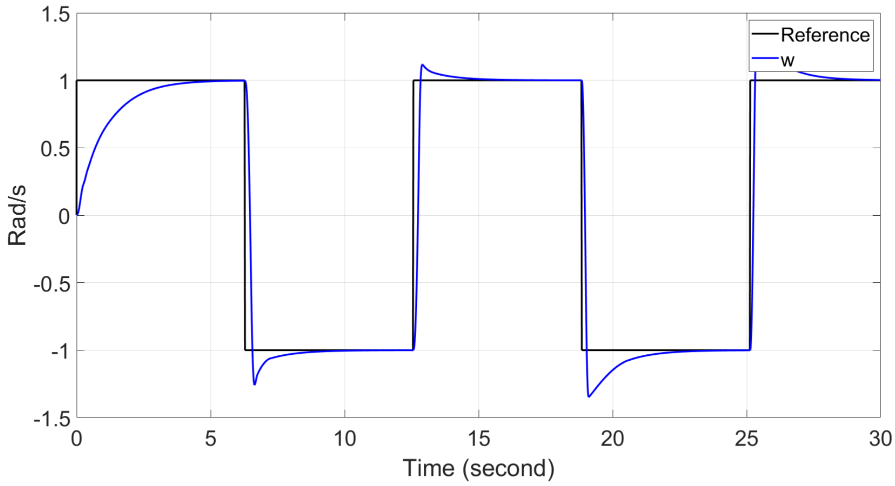

Figure 3.33 : Şen FIS NFC results for inputs (normalized error, output derivative, sum of error) where w=1.

is because of boundaries that is used for the Mamdani FIS consequent parameters as (-100, 100). If the restrictions are removed then the results could be the similar.

3.5 Comparison of NFC Methods (Sugeno, Mamdani, Şen)

The three fuzzy inference systems (Sugeno, Mamdani and Şen) are tested for three inputs NFCs as (error, output derivative, sum of errors) and (normalized error, output derivative, sum of errors). The results are given in Tables (3.3, 3.4) that in (normalized error, output derivative, sum of errors) inputs the Sugeno is the best at settling time and simulation time. However, Mamdani and Şen FIS are better than Sugeno at RMSE. Besides, Şen FIS and Sugeno FIS computation times are similar. So, the Şen NFC can

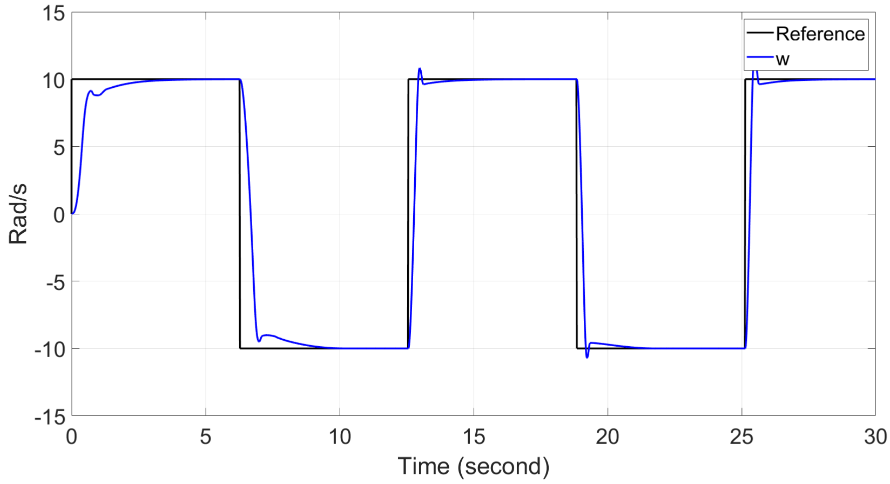

Figure 3.34 : Şen FIS NFC results for inputs (normalized error, output derivative, sum of error) where w=10.

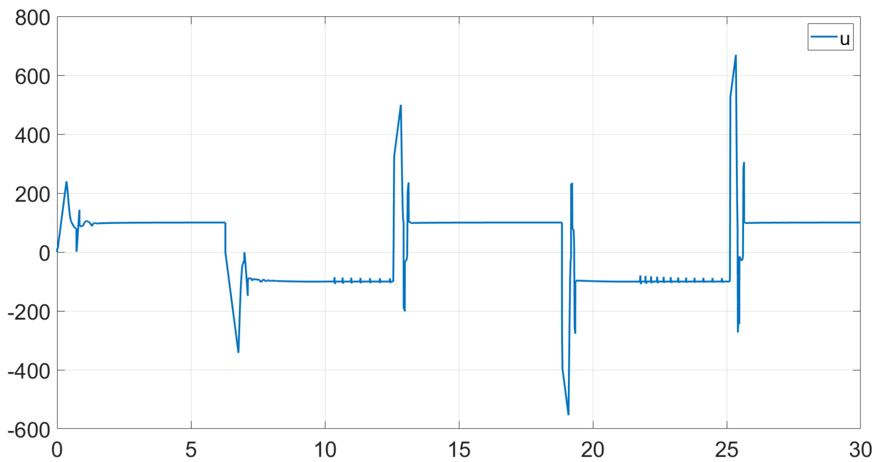

Figure 3.35 : Şen FIS NFC control signals for inputs (normalized error, output derivative, sum of error) where w=10.

be used instead of the Sugeno NFC in this inputs combination with more computation time. Besides, the Şen NFC overshoot values are better than Sugeno NFC in case studies generally.

Table 3.3 : Comparison of NFCs with three inputs (error, output derivative, sum of error) for w=10.

Inputs	%OS	T_s	RMSE	Computation Time (second)
Sugeno NFC	27.1	1.390	0.2815	47.3358
Mamdani NFC	0	7.429	0.6721	888.9318
Şen NFC	10.5	2.483	0.3427	72.0989

Table 3.4 : Comparison of NFCs with three inputs (normalized error, output derivative, sum of error) for w=10.

Inputs	%OS	T_s	RMSE	Computation Time (second)
Sugeno NFC	4.6	1.440	2.6404	42.8820
Mamdani NFC	0	2.555	0.3883	901.8182
Şen NFC	0	2.557	0.3873	74.3647

However, it is seen that in (error, output derivative, sum of errors) inputs the Sugeno is the best at settling time, RMSE and computation time. In this inputs combination the Sugeno NFC results are better than other FIS controllers. When all NFC results were examined, it is seen that the Sugeno NFC system with (error, output derivative, sum of errors) inputs is the best [31] so, this NFC control combination is chosen to compare in the best way the effects of the selected controllers (type-1 NFC, type-2 NFC, FEL) on the quadrotor model.

It is seen in these NFC studies that the control signals are very wavy. Because of that, directly using the ANFIS for a real system's controller is not efficient. So, Feedback Error Learning (FEL) controller is examined and tested for the DC motor.

3.6 Feedback Error Learning (FEL) Control with Fuzzy Logic

The FEL control is firstly proposed by Kawato [33] that it does not need a certain model. In [36], two different FEL structure is given. In the first structure, FEL inputs are desired input values and in the second structure, the FEL inputs are the system output values. There are different proposed FEL structures in literature [35] that the stability of the controller was discussed in [34]. The three different FEL structures are tested that their inputs were "desired input", "error" and "error, error derivative". It is seen that the FEL structure with error input gives the best results as examined in [37].

The DC motor PID coefficients are selected as [10 20 10]. The results of PID and FEL controllers are given in Figure 3.37. They have the similar control signals as given in Figures 3.38. In the FEL control, the conventional controller is the main controller and the NFC is the secondary controller. The NFC is used to decrease overshoot, settling time and RMSE. It is seen from Figure Figures 3.38 that the FEL controller signals are more regular when compared to the only NFC signals.

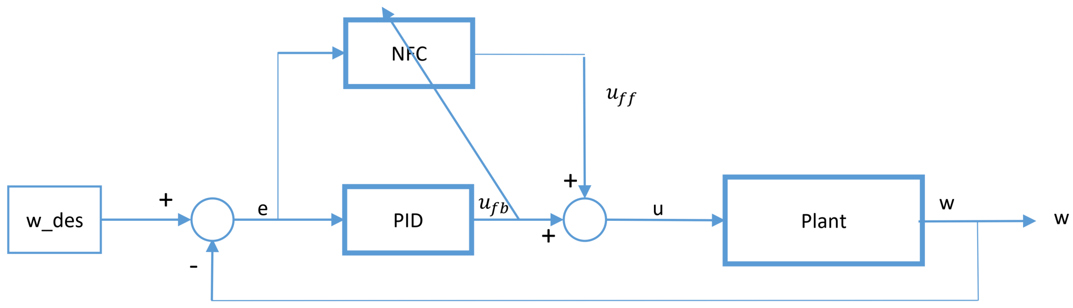

Figure 3.36 : FEL scheme.

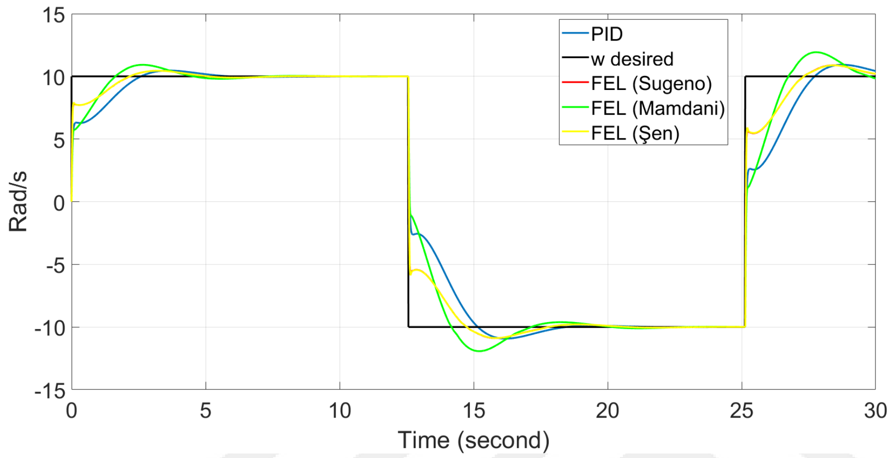

Figure 3.37 : Comparison of PID and FEL controllers for w=10.

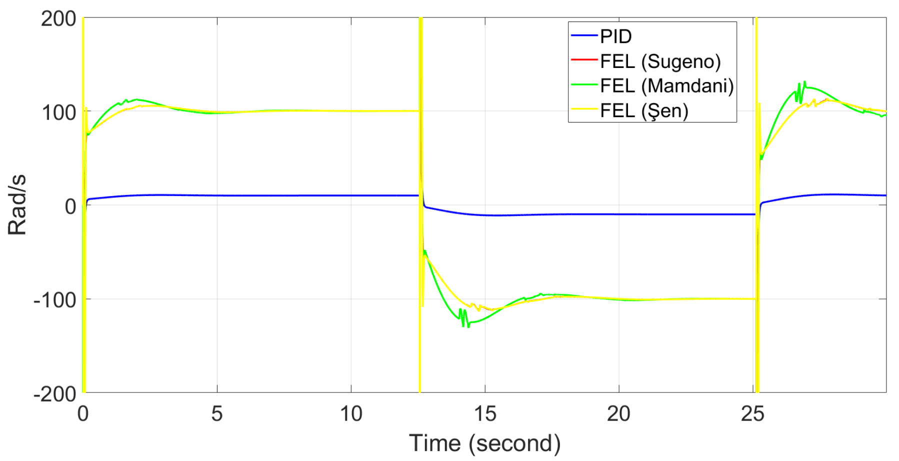

Figure 3.38 : Comparison of PID and FEL control signals for w=10.

As seen from Table 3.5 the Mamdani FLC has more computation load than other fuzzy logic systems. The Sugeno FEL and Şen FEL controllers have similar results. They decrease the overshoot, settling time and RMSE value. This study is tested for a quadrotor model in the following chapters.

Table 3.5 : Comparison of FEL controllers.

Inputs	%OS	T_s	RMSE	Computation Time (second)
PID	4.6	4.911	2.3728	0.4306
FEL (Sugeno)	4.5	4.362	1.4644	3.1239
FEL (Mamdani)	9.2	4.127	2.1057	16.5186
FEL (Şen)	4.5	4.397	1.4727	3.4909

Consequently, the contribution of this chapter is to test and compare the NFC structures and determine the best of fuzzy logic based controllers. These determined control structures are used in the following studies.

4. INTERVAL TYPE-2 ANFIS MODEL WITH MODIFIED KMA

Type-1 Fuzzy Inference System (T1 FIS) was proposed firstly by Lotfi A. Zadeh [65] that T1FIS is very useful to represent nonlinear systems and non-mathematical problems like human behavior that is not possible to solve mathematically. However, T1 FLS antecedent parameters are not fuzzy and so T1 FLS is theoretically insufficient to uncertainty. Because of that, Type-2 FLS (T2 FLS) is proposed to deal with uncertainties in a better way in 1975 [40].

T2 FIS is very operational on uncertain and nonlinear systems but defining antecedent and consequent parameters are still in research. Type-2 Fuzzy sets include uncertainty and so any type-1 Fuzzy Set can not represent Type-2 Fuzzy Sets. To overcome the difficulty of dealing with T2 membership functions, Interval Type-2 FLS (IT2 FLS) is proposed in 2000 [42] that every Fuzzy Sets consist of two Type-1 MFs as given in Figure 4.1. The external MF is named as Upper Membership Function (UMF) and the internal MF is named as Lower Membership Function (LMF). The area between the UMF and LMF is named as Footprint of Uncertainty (FOU).

Type-2 has similar rule base as Type-1 [41]:

r_i : If x_1 is X_1^n and ... and x_l is X_l^n, then y is Y_n

where r_i is rules, X_l^n is fuzzy sets of the antecedent variables and Y^n is the consequent membership set. IT2 FLS has the same structure as T1 FLS but in the next steps, there is clearness because of the membership functions. In the IT2 FLS, it is impossible to calculate the defuzzification step as it is done in T1 FLS. Type-reduction must be implemented to Interval Type-2 FLS. The type-reduction is defined as an optimization problem and solved with different methods in the literature [46]. These type-reduction methods are given in this chapter in detail.

There are different optimization methods for IT2 FLS parameters training as Big-bang big-crunch optimization [66] and Gradient Descent based algorithms [56]. A T1

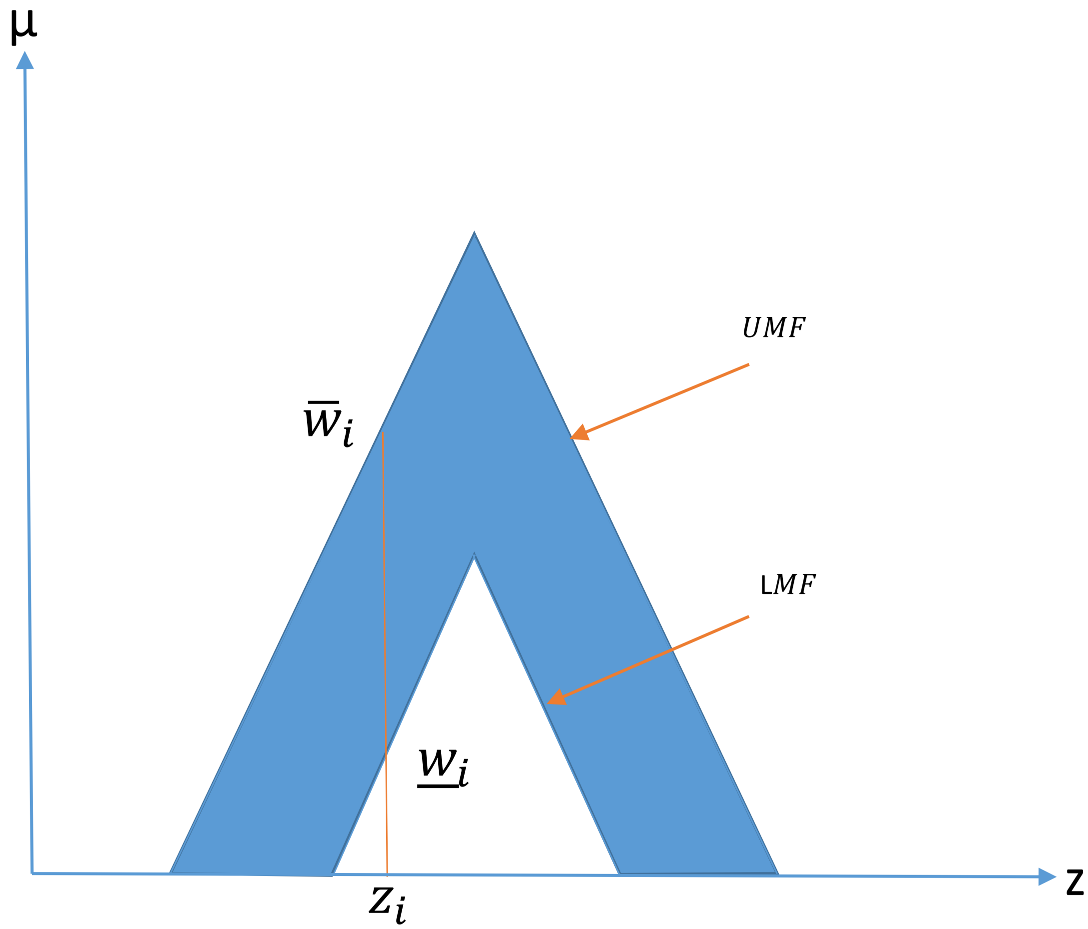

Figure 4.1 : Interval Type-2 membership function.

ANFIS like approach is adapted to IT2 ANFIS in literature [60,67]. When the results were compared, IT2 ANFIS results were not better than T1 ANFIS results.

In this chapter, we proposed a new IT2 ANFIS model. We modified the Karnik-Mendel algorithm which is one of the type reduction methods to use in the proposed IT2 ANFIS approach. The proposed IT2 ANFIS model is tested and compared to T1 ANFIS. The results are given in tables and examined.

4.1 Interval Type-2 Fuzzy Inference Systems

In IT2 FLS, there are two different weights and so they must be degraded to one weight value [68].

$$Y = \frac{\sum_{i=1}^{N} F_i W_i}{\sum_{i=1}^{N} W_i} \equiv [y_l, y_r] \tag{4.1}$$

$$f_i \in F_i \equiv [\underline{f}_i, \overline{f}_i] \qquad i = 1, 2, \ldots, N \tag{4.2}$$

$$w_i \in W_i \equiv [\underline{w}_i, \overline{w}_i] \qquad i = 1, 2, \ldots, N \tag{4.3}$$

where f is $f_i = p_i * x_1 + q_i * x_2 + r_i$. For sugeno interval type-2 fuzzy inference system (IT2 FIS) F_i can be given with p, q, r values. The equation (4.1) gives two results because of weights' uncertainty. The two weight series must be chosen as minimum and maximum:

$$y_l = min\frac{\sum_{i=1}^{N} f_i w_i}{\sum_{i=1}^{N} w_i} \tag{4.4}$$

$$y_r = max\frac{\sum_{i=1}^{N} f_i w_i}{\sum_{i=1}^{N} w_i} \tag{4.5}$$

To calculate minimum and maximum values, switch points for weights must be calculated [69]. It was stated before that:

$$y = \frac{\sum_{i=1}^{N} f_i w_i}{\sum_{i=1}^{N} w_i} \tag{4.6}$$

To determine switch point (k), y derivative is calculated and equated to zero.

$$\frac{\partial y}{\partial w_k} = \frac{\partial}{\partial w_k}[\frac{\sum_{i=1}^{N} f_i w_i}{\sum_{i=1}^{N} w_i}] = \frac{\partial}{\partial w_k}[\frac{f_k w_k + \sum_{i=1,i\neq k}^{N} f_i w_i}{w_k + \sum_{i=1,i\neq k}^{N} w_i}] \tag{4.7}$$

$$= \frac{f_k(w_k + \sum_{i=1,i\neq k}^{N} w_i) - (f_k w_k + \sum_{i=1,i\neq k}^{N} f_i w_i)1}{(w_k + \sum_{i=1,i\neq k}^{N} w_i)^2} \tag{4.8}$$

$$= \frac{f_k}{w_k + \sum_{i=1,i\neq k}^{N} w_i} - \frac{\sum_{i=1}^{N} f_i w_i}{(\sum_{i=1}^{N} w_i)^2} = \frac{f_k}{\sum_{i=1}^{N} w_i} - \frac{\sum_{i=1}^{N} f_i w_i}{\sum_{i=1}^{N} w_i}\frac{1}{\sum_{i=1}^{N} w_i} \tag{4.9}$$

$$\frac{\partial y}{\partial w_k} = \frac{f_k - y}{\sum_{i=1}^{N} w_i} \Rightarrow y = f_k \tag{4.10}$$

When the result is placed in equation (4.6):

$$y = f_k \Rightarrow \frac{\sum_{i=1}^{n} f_i w_i}{\sum_{i=1}^{n} w_i} = f_k \Rightarrow \sum_{i=1}^{n} f_i w_i = f_k \sum_{i=1}^{n} w_i \Rightarrow \sum_{i=1,i\neq k}^{n} f_i w_i \neq f_k \sum_{i=1,i\neq k}^{n} w_i \tag{4.11}$$

The inequality shows that the switch points can not be calculated analytically. So, logical approach is used for the equation (4.12).

$$\frac{\partial y}{\partial w_k} = \frac{f_k - y}{\sum_{i=1}^{n} w_i} \tag{4.12}$$

With the assumption that $\sum_{i=1}^{N} w_i > 0$:

$$\frac{\partial y}{\partial w_k} = \begin{cases} \geq 0, & f_k \geq y \\ < 0, & f_k < y \end{cases} \tag{4.13}$$

It can be shown from the equation (4.12) that for $f_k > y$, $\frac{\partial y}{\partial w_k}$ will be bigger than zero. It means that y increases for increasing w_k values. For $f_k < y$, $\frac{\partial y}{\partial w_k}$ will be smaller than zero. It means that y decreases for increasing w_k values. This leads to the following results.

(I) if $f_k > y$ and w_k increases(decreases), y increases(decreases)
(II) if $f_k < y$ and w_k increases(decreases), y decreases(increases)

This conditions defines how to choose the switching points and the weights. As stated in the conditions (I), minimum y values is calculated by minimum w values. Maximum y values is calculated by the maximum w values. In the condition (II), the calculations are made vice versa.

The condition (I) defines the minimum y value as:

$$w_k = \begin{cases} \underline{w}_k, & f_k > y \\ \overline{w}_k, & f_k < y \end{cases} \tag{4.14}$$

The condition (II) defines the maximum y value as:

$$w_k = \begin{cases} \overline{w}_k, & f_k > y \\ \underline{w}_k, & f_k < y \end{cases} \tag{4.15}$$

So, to calculate minimum and maximum y values a switch point must be defined that the switch point is stated as "l" and "r".

$$y_l = \frac{\sum_{i=1}^{l} \underline{f}_i \overline{w}_i + \sum_{i=l+1}^{N} \underline{f}_i \underline{w}_i}{\sum_{i=1}^{l} \overline{w}_i + \sum_{i=l+1}^{N} \underline{w}_i} \tag{4.16}$$

$$y_r = \frac{\sum_{i=1}^{r} \overline{f}_i \underline{w}_i + \sum_{i=r+1}^{N} \overline{f}_i \overline{w}_i}{\sum_{i=1}^{r} \underline{w}_i + \sum_{i=r+1}^{N} \overline{w}_i} \tag{4.17}$$

In literature, there are very many optimization methods to calculate the switch points. Four of them are mentioned in this study that they are Karnik-Mendel Algorithms, Enhanced Karnik-Mendel Algorithms, Iterative Algorithm with Stop Condition and Enhanced Iterative Algorithm with Stop Condition. They give the same results but only the simulation times change.

4.1.1 Karnik-Mendel Algorithm (KMA)

Karnik-Mendel is the proposed first algorithm to determine the switch points [70].

Computing y_l:

1- Sort $\underline{f}_i$ in increasing order. Match the weights with $\underline{f}_i$.

2- Initialize w_i by setting $w_i = \frac{\underline{w}_i + \overline{w}_i}{2}$ and compute

$$y = \frac{\sum_{i=1}^{N} \underline{f}_i w_i}{\sum_{i=1}^{N} w_i} \tag{4.18}$$

3- Find switch point "l" for $\underline{f}_l < y \leq \underline{f}_{l+1}$

4- Compute

$$y' = \frac{\sum_{i=1}^{l} \underline{f}_i \overline{w}_i + \sum_{i=l+1}^{N} \underline{f}_i \underline{w}_i}{\sum_{i=1}^{l} \overline{w}_i + \sum_{i=l+1}^{N} \underline{w}_i} \tag{4.19}$$

5- If $y' = y$, set $y_l = y$; if not, set $y = y'$ and go to Step 3.

Computing y_r:

1- Sort $\overline{f}_i$ in increasing order. Match the weights with $\overline{f}_i$.

2- Initialize w_i by setting $w_i = \frac{\underline{w}_i + \overline{w}_i}{2}$ and compute

$$y = \frac{\sum_{i=1}^{N} \overline{f}_i w_i}{\sum_{i=1}^{N} w_i} \tag{4.20}$$

3- Find switch point "r" for $\overline{f}_r < y \leq \overline{f}_{r+1}$

4- Compute

$$y' = \frac{\sum_{i=1}^{r} \overline{f}_i \underline{w}_i + \sum_{i=r+1}^{N} \overline{f}_i \overline{w}_i}{\sum_{i=1}^{r} \underline{w}_i + \sum_{i=r+1}^{N} \overline{w}_i} \tag{4.21}$$

5- If $y' = y, y_r = y$; if not, set y=y' and go to Step 3.

4.1.2 Enhanced Karnik-Mendel Algorithm (EKMA)

Enhanced Karnik-Mendel Algorithm (EKMA) is introduced as enhanced KMA [44]. This method propose starting values for switch points that is near to targets. So, this method gives the switch point with less iterations. By this way, simulation time is decreased. The equations (4.22,4.23) are used to define KMA start point. This generally gives midpoints and iteration starts.

$$y_l = \frac{\sum_{i=1}^{N} \underline{f}_i \frac{\overline{w}_i + \underline{w}_i}{2}}{\sum_{i=1}^{N} \frac{\overline{w}_i + \underline{w}_i}{2}} \tag{4.22}$$

$$y_r = \frac{\sum_{i=1}^{N} \overline{f}_i \frac{\overline{w}_i + \underline{w}_i}{2}}{\sum_{i=1}^{N} \frac{\overline{w}_i + \underline{w}_i}{2}} \tag{4.23}$$

In EKMA, some tests are implemented to initialize L and R values those are $L_0 = N/2.4, R_0 = N/1.7$ [44].

Computing y_l:

1- Sort $\underline{f}_i$ in increasing order. Match the weights with $\underline{f}_i$.

2- Set l=N/2.4 (the nearest integer to N/2.4) and compute

$$a = \sum_{i=1}^{l} \underline{f}_i \overline{w}_i + \sum_{i=l+1}^{N} \underline{f}_i \underline{w}_i \tag{4.24}$$

$$b = \sum_{i=1}^{l} \overline{w}_i + \sum_{i=l+1}^{N} \underline{w}_i \tag{4.25}$$

$$y = a/b \tag{4.26}$$

3- Find switch point $l' \in [1, N-1]$ that $\underline{f}_{l'} < y \leq \underline{f}_{l'+1}$

4- If $l' = l$, stop; otherwise, continue.

5-Compute $s = sign(l' - l)$, and

$$a' = a + s \sum_{i=min(l+l')+1}^{max(l,l')} \underline{f}_i (\overline{w}_i - \underline{w}_i) \tag{4.27}$$

$$b' = b + s \sum_{i=min(l+l')+1}^{max(l,l')} (\overline{w}_i - \underline{w}_i) \tag{4.28}$$

$$y' = a'/b' \tag{4.29}$$

6- If $y' = y, a = a', b = b'$ and $l = l'$. Go to Step 3.

Computing y_r:

1- Sort $\overline{f}_i$ in increasing order. Match the weights with $\overline{f}_i$.

2- Set r=N/1.7 (the nearest integer to N/1.7) and compute

$$a = \sum_{i=1}^{r} \overline{f}_i \underline{w}_i + \sum_{i=r+1}^{N} \overline{f}_i \overline{w}_i \tag{4.30}$$

$$b=\sum_{i=1}^{r}\underline{w}_i+\sum_{i=r+1}^{N}\overline{w}_i \tag{4.31}$$

$$y=a/b \tag{4.32}$$

3- Find switch point $r' \in [1, N-1]$ that $\overline{f}_{r'} < y \leq \overline{f}_{r'+1}$

4- If $r' = r$, stop; otherwise, continue.

5-Compute $s = sign(r' - r)$, and

$$a'=a-s\sum_{i=min(r+r')+1}^{max(r,r')}\overline{f}_i(\overline{w}_i-\underline{w}_i) \tag{4.33}$$

$$b'=b-s\sum_{i=min(r+r')+1}^{max(r,r')}(\overline{w}_i-\underline{w}_i) \tag{4.34}$$

$$y'=a'/b' \tag{4.35}$$

6- If $y' = y, a = a', b = b'$ and $r = r'$. Go to Step 3.

4.1.3 Iterative Algorithm with Stop Condition (IASC)

IASC depends on the increase and decrease of y_l and y_r [45]. Starting points to iterations are zero.

Computing y_l:

1- Sort $\underline{f}_i$ in increasing order. Match the weights with $\underline{f}_i$.

2- Initialize y_l and l

$$a=\sum_{i=1}^{N}\underline{f}_i\underline{w}_i \qquad b=\sum_{i=1}^{N}\underline{w}_i \tag{4.36}$$

$$y_l=\underline{f}_N \qquad l=0 \tag{4.37}$$

3-Compute $l = l + 1$, and

$$a=a+\underline{f}_l(\overline{w}_l-\underline{w}_l) \qquad b=b+\overline{w}_l-\underline{w}_l \tag{4.38}$$

$$c=a/b \tag{4.39}$$

4- Check that if $c > y_l$. If yes, stop and set $l = l - 1$. If not $y_l = c$ and go to Step 3.

Computing y_r:

1- Sort $\overline{c}_i$ in increasing order. Match the weights with $\underline{c}_i$.

2- Initialize y_r and r

$$a = \sum_{i=1}^{N} \overline{c}_i \overline{w}_i \qquad b = \sum_{i=1}^{N} \overline{w}_i \tag{4.40}$$

$$y_r = \overline{c}_1 \qquad r = 0 \tag{4.41}$$

3-Compute $r = r + 1$, and

$$a = a - \overline{c}_r(\overline{w}_l - \underline{w}_l) \qquad b = b - \overline{w}_r + \underline{w}_r \tag{4.42}$$

$$c = a/b \tag{4.43}$$

4- Check that if $c > y_r$. If yes, stop and set $r = r - 1$. If not $y_r = c$ and go to Step 3.

4.1.4 Enhanced Iterative Algorithm with Stop Condition (EIASC)

EIASC is enhanced to decrease iteration time that the starting switch point is given as r=N [46].

Computing y_l:

1- Sort $\underline{f}_i$ in increasing order. Match the weights with $\underline{f}_i$.

2- Initialize y_l and l

$$a = \sum_{i=1}^{N} \underline{f}_i \underline{w}_i \qquad b = \sum_{i=1}^{N} \underline{w}_i \tag{4.44}$$

$$y_l = \underline{f}_N \qquad l = 0 \tag{4.45}$$

3-Compute $l = l + 1$, and

$$a = a + \underline{f}_l(\overline{w}_l - \underline{w}_l) \qquad b = b + \overline{w}_l - \underline{w}_l \tag{4.46}$$

$$y_l = a/b \tag{4.47}$$

4- Check that if $\underline{f}_{l+1} \geq y_l$. If yes stop. If not go to Step 3.

Computing y_r:

1- Sort $\overline{f}_i$ in increasing order. Match the weights with $\underline{f}_i$.

2- Initialize y_r and r

$$a = \sum_{i=1}^{N} \overline{f}_i \underline{w}_i \qquad b = \sum_{i=1}^{N} \underline{w}_i \tag{4.48}$$

$$y_r = \overline{f}_1 \qquad r = N \tag{4.49}$$

3-Compute

$$a = a + \overline{f}_r(\overline{w}_l - \underline{w}_l) \qquad b = b + \overline{w}_r - \underline{w}_r \tag{4.50}$$

$$y_l = a/b \qquad r = r - 1 \tag{4.51}$$

4- Check that if $y_r \geq \overline{f}_r$. If yes stop. If not go to Step 3.

4.2 Interval Type-2 Optimization in Literature

In literature, there are different methods used in Type-2 optimization. As one of the optimization studies, the LSE and back-propagation are implemented and results are compared to Type-1 optimization in [60,67]. The proposed ANFIS architecture in [60,67] for the IT2 FLS is given below:

4.2.1 Interval Type-2 ANFIS LSE Method (KMA)

Type-2 ANFIS LSE structure is same as given in Type-1 ANFIS.

Type-1 Sugeno ANFIS LSE Structure

Layer 1:

$$O_i^1 = \mu_{A_i}(x) \qquad\qquad O_i^1 = \mu_{B_i}(y) \tag{4.52}$$

Layer 2:

$$O_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = w_i \tag{4.53}$$

Layer 3:

$$O_i^3 = \overline{w_i} = \frac{w_i}{\sum_{i=1}^n w_i} \tag{4.54}$$

Layer 4:

$$O_i^4 = y_i = \overline{w_i} f_i == \overline{w_i}(p_i x_1 + q_i x_2 + r_i) \tag{4.55}$$

Layer 5:

$$O_i^5 = \sum_{i=1}^n y_i = \sum_{i=1}^n \overline{w_i} f_i = (\overline{w}_1 x_1)p_1 + (\overline{w}_1 x_2)q_1 + \overline{w}_1 r_1 + (\overline{w}_2 x_2)p_2 + (\overline{w}_2 x_2)q_2 + \overline{w}_2 r_2 \tag{4.56}$$

$$y = (w_1 x_1)p_1 + (w_1 x_2)q_1 + w_1 r_1 + (w_2 x_1)p_2 + (w_2 x_2)p_2 + w_2 r_2 \tag{4.57}$$

$$y = \begin{bmatrix} w_1x_1 & w_1x_2 & w_1 & w_2x_1 & w_2x_2 & w_2 \end{bmatrix} \begin{bmatrix} p_1 \\ q_1 \\ r_1 \\ p_2 \\ q_2 \\ r_2 \end{bmatrix} = XW \tag{4.58}$$

$$Y = XW \Rightarrow W = X^{-1}Y \tag{4.59}$$

Interval Type-2 LSE Structure [60,67] (KMA)

Layer 1:

$$\underline{O}_i^1 = \mu_{A_i}(x) \qquad \underline{O}_i^1 = \mu_{B_i}(y) \qquad\qquad \overline{O}_i^1 = \mu_{A_i}(x) \qquad \overline{O}_i^1 = \mu_{B_i}(y) \tag{4.60}$$

Layer 2:

$$\underline{O}_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = \underline{w}_i \qquad\qquad \overline{O}_i^2 = \mu_{A_i}(x) * \mu_{B_i}(y) = \overline{w}_i \tag{4.61}$$

Layer 3:

$$\underline{O}_i^3 = \overline{\underline{w}_i} = \frac{\underline{w}_i}{\sum_{i=1}^n \underline{w}_i} \qquad\qquad \overline{O}_i^3 = \overline{\overline{w}}_i = \frac{\overline{w}_i}{\sum_{i=1}^n \overline{w}_i} \tag{4.62}$$

Layer 4:

$$\underline{O}_i^4 = \underline{y}_i = \overline{\underline{w}_i}\underline{f}_i \qquad\qquad \overline{O}_i^4 = \overline{y}_i = \overline{\overline{w}}_i\overline{f}_i \tag{4.63}$$

Layer 5:

$$\underline{O}_i^5 = \sum_{i=1}^n \underline{y}_i \qquad\qquad \overline{O}_i^5 = \sum_{i=1}^n \overline{y}_i \tag{4.64}$$

As seen from the equations, there are two output and two different weight series in Interval Type-2 ANFIS. So these equations are reduced to one value by taking the average. So, an LSE-like structure with Type-1 ANFIS is used to calculate the "W" value. But it does not work properly. Because the weights and outputs are accepted as average. However, the outputs depend on switch points and the weights can not be averaged. So, the results are worse compared to Type-1 [60,61].

$$O_i^5 = y = \frac{(\underline{Q}^5 + \overline{Q}^5)}{2} \tag{4.65}$$

$$X = \begin{bmatrix} \frac{\underline{w_1} + \overline{w_1}}{2}x_1 & \frac{\underline{w_1} + \overline{w_1}}{2}x_2 & \frac{\underline{w_1} + \overline{w_1}}{2} & \ldots \end{bmatrix} \tag{4.66}$$

$$Y = XW \Rightarrow W = X^{-1}Y \tag{4.67}$$

4.2.2 Interval Type-2 ANFIS Gradient Method (KMA)

As discussed in previous heading, the LSE based ANFIS optimization method with Karnik-Mendel algorithm is incompatible to Interval Type-2 FLS. The IT2 ANFIS with Gradient Descent Method is given below where "y" is measured output values, "y^t" is calculated output values, "lr" is learning rate and $W = r_i$.

$$E = \frac{1}{2}(y - y^t)^2 \qquad y = XW \tag{4.68}$$

$$r(t+1) = r(t) - lr\frac{\partial E}{\partial r_i} \tag{4.69}$$

$$\frac{\partial E}{\partial r_i} = (XW - y^t)X = (y - y^t)X \tag{4.70}$$

$$r(t+1) = r(t) - lr(y - y^t)X \Rightarrow W(t+1) = W(t) - lr(y - y^t)X \tag{4.71}$$

It is seen that the GD based algorithm considers the "X" and "y". The output and weights were defined as average. However, it would not give exact results. So, the LSE-like problem will be in this example.

The question is that can $\frac{\partial E}{\partial r_i}$ be calculated without "X". The error is computed and differentiated with respect to r. This gives us different results for every r. All differentiated values are added to r as seen in equation (4.69). This is trained and it is seen that the result was not acceptable.

4.3 Interval Type-2 Modified Karnik-Mendel Based ANFIS Optimization

A sample IT2 FLS MF is depicted in Figure 4.1 that input MF values are defined for two cases. So, it is impossible to define "X" to calculate IT2 FLS parameters. In this study, Karnik-Mendel Algorithm is modified and the "X" value is calculated from the Modified Karnik-Mendel Algorithm (M-KMA). The M-KMA is given below:

Computing y_l:

1- Sort $\underline{f}_i$ in increasing order. Match the weights with $\underline{f}_i$.

2- Initialize w_i by setting $w_i = \frac{\underline{w}_i + \overline{w}_i}{2}$ and compute

$$y = \frac{\sum_{i=1}^{N} \underline{f}_i w_i}{\sum_{i=1}^{N} w_i} \tag{4.72}$$

3- Find switch point "l" for $\underline{f}_l < y \leq \underline{f}_{l+1}$

4- Compute $y' = X_l W$ where

$$X_l = \frac{\begin{bmatrix} \overline{w}_1 x_1 & \overline{w}_1 x_2 & \overline{w}_1 & \dots & \overline{w}_l x_1 & \overline{w}_l x_2 & \overline{w}_l \dots \\ \underline{w}_{l+1} x_1 & \underline{w}_{l+1} x_2 & \underline{w}_{l+1} & \dots & \underline{w}_N x_1 & \underline{w}_N x_2 & \underline{w}_N \end{bmatrix}}{\sum_{i=1}^{l} \overline{w}_i + \sum_{i=l+1}^{N} \underline{w}_i} \tag{4.73}$$

5- If $y' = y$, set $y_l = y$; if not, set $y = y'$ and go to Step 3.

Computing y_r:

1- Sort $\overline{f}_i$ in increasing order. Match the weights with $\overline{f}_i$.

2- Initialize w_i by setting $w_i = \frac{\underline{w}_i + \overline{w}_i}{2}$ and compute

$$y = \frac{\sum_{i=1}^{N} \overline{f}_i w_i}{\sum_{i=1}^{N} w_i} \tag{4.74}$$

3- Find switch point "l" for $\overline{f}_l < y \leq \overline{f}_{l+1}$

4- Compute $y' = X_l W$ where

$$X_r = \frac{\begin{bmatrix} \underline{w}_1 x_1 & \underline{w}_1 x_2 & \underline{w}_1 & \dots & \underline{w}_l x_1 & \underline{w}_l x_2 & \underline{w}_l \dots \\ \overline{w}_{l+1} x_1 & \overline{w}_{l+1} x_2 & \overline{w}_{l+1} & \dots & \overline{w}_N x_1 & \overline{w}_N x_2 & \overline{w}_N \end{bmatrix}}{\sum_{i=1}^{l} \overline{w}_i + \sum_{i=l+1}^{N} \underline{w}_i} \tag{4.75}$$

5- If $y' = y$, set $y_l = y$; if not, set $y = y'$ and go to Step 3.

Result y:

$$y = \frac{y_r + y_l}{2} = \frac{X_l W + X_r W}{2} = \frac{X_l + X_r}{2} W = XW \tag{4.76}$$

The defined "X" matrix is a valid matrix to reach the same result with the Karnik-Mendel algorithm.

4.3.1 Interval Type-2 ANFIS (M-KMA)

The proposed IT2 ANFIS model is shown in Fig. 4.2 for two inputs and one output. The inputs are named x, the output is named y. The IT2 ANFIS structure has seven layers described below.

Layer 1: This layer inputs are crisp values and outputs are MF results. This MF results are defined as upper and lower values.

$$\begin{aligned} \underline{O}_i^1 &= \underline{\mu}_{A_i}(x) \qquad & \underline{O}_i^1 &= \underline{\mu}_{B_i}(y) \\ \overline{O}_i^1 &= \overline{\mu}_{A_i}(x) \qquad & \overline{O}_i^1 &= \overline{\mu}_{B_i}(y) \end{aligned} \tag{4.77}$$

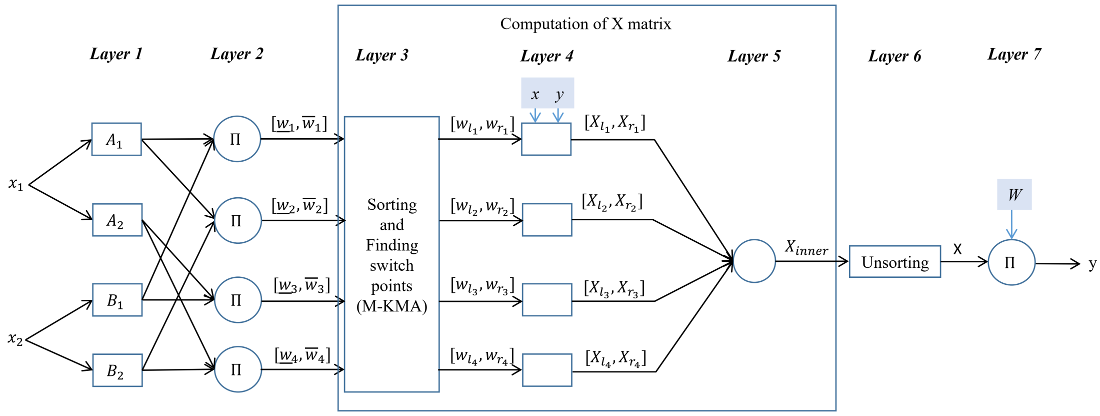

Figure 4.2 : Interval Type-2 ANFIS structure.

Layer 2: The weights are multiplied as the pairs which are the upper and lower values.

$$\begin{aligned} \underline{O}_i^2 &= \underline{\mu}_{A_i}(x) * \underline{\mu}_{B_i}(y) = \underline{w}_i \\ \overline{O}_i^2 &= \overline{\mu}_{A_i}(x) * \overline{\mu}_{B_i}(y) = \overline{w}_i \end{aligned} \tag{4.78}$$

Layer 3: In this layer, switch points are calculated and the weights are sorted with respect to f values where $f_i = p_i x_1 + q_i x_2 + r_i$.

$$\begin{aligned} \underline{O}_i^3 &= sort(\underline{w}_i \ \ \overline{w}_i \ \ \underline{f}_i) \\ \overline{O}_i^3 &= sort(\underline{w}_i \ \ \overline{w}_i \ \ \overline{f}_i) \end{aligned} \tag{4.79}$$

Layer 4: In this layer, X matrices are calculated.

$$\underline{O}_i^4 = X_l = \frac{\begin{bmatrix} \overline{w}_1 x_1 & \overline{w}_1 x_2 & \overline{w}_1 & \dots & \overline{w}_l x_1 & \overline{w}_l x_2 & \overline{w}_l \dots \\ \underline{w}_{l+1} x_1 & \underline{w}_{l+1} x_2 & \underline{w}_{l+1} & \dots & \underline{w}_n x_1 & \underline{w}_n x_2 & \underline{w}_n \end{bmatrix}}{\sum_{n=1}^{l} \overline{w}_n + \sum_{n=l+1}^{N} \underline{w}_n} \tag{4.80}$$

$$\overline{O}_i^4 = X_r = \frac{\begin{bmatrix} \underline{w}_1 x_1 & \underline{w}_1 x_2 & \underline{w}_1 & \dots & \underline{w}_r x_1 & \underline{w}_r x_2 & \underline{w}_r \dots \\ \overline{w}_{r+1} x_1 & \overline{w}_{r+1} x_2 & \overline{w}_{r+1} & \dots & \overline{w}_n x_1 & \overline{w}_n x_2 & \overline{w}_n \end{bmatrix}}{\sum_{n=1}^{r} \underline{w}_n + \sum_{n=r+1}^{N} \overline{w}_n} \tag{4.81}$$

Layer 5: The X matrices are converted to a single X matrix.

$$O_i^5 = X_{inner} = \frac{\underline{O}_i^4 + \overline{O}_i^4}{2} = \frac{X_l + X_r}{2} \tag{4.82}$$

Layer 6: The calculated X matrix in Layer 5 emerged as sorted matrix. However, the X matrix must be unsorted.

$$O_i^6 = X = unsort(X_{inner}) \tag{4.83}$$

Layer 7: The unsorted matrix X obtained in Layer 6 must be implemented in the following equation.

$$O_i^7 = y = XW \tag{4.84}$$

Hence, y values can be obtained by using X values. LSE and GD can now be applied to IT2 ANFIS. LSE is implemented as seen in equation (4.85).

$$W = X^{-1}y \tag{4.85}$$

Gradient Descent method equation is given below for "y" is output values, "y^t" is target values, and "lr" is learning rate.

$$E = \frac{1}{2}(y - y^t)^2 \qquad y = XW \tag{4.86}$$

$$W(t+1) = W(t) - lr\frac{\partial E}{\partial W_i} \tag{4.87}$$

$$\frac{\partial E}{\partial W_i} = (y - y^t)X \tag{4.88}$$

$$W(t+1) = W(t) - lr(y - y^t)X \tag{4.89}$$

It is seen from the equation (4.89) output parameters "W" are tuned with respect to error "$y - y^t$" and X matrix.

4.4 Results

The contribution of this section is the proposition of the M-KMA and IT2 ANFIS structure. In this way, the training of the IT2 FLS parameters has become as easy as the training of T1 FLS parameters. So, The T1 ANFIS, IT2 KMA based ANFIS and IT2 M-KMA based ANFIS are tested for the equation (4.90) [7,62]. The equation has two inputs and one output. The antecedent membership function parameters are chosen as triangular and fixed as given in Figure 4.3. Only consequent parameters are trained. The antecedent parameters are chosen as IT2 MFs and consequent parameters are chosen as linear. Every parameter and initial value are defined the same to make a fair comparison.

$$z = \frac{sin(x)}{x}\frac{sin(y)}{y} \tag{4.90}$$

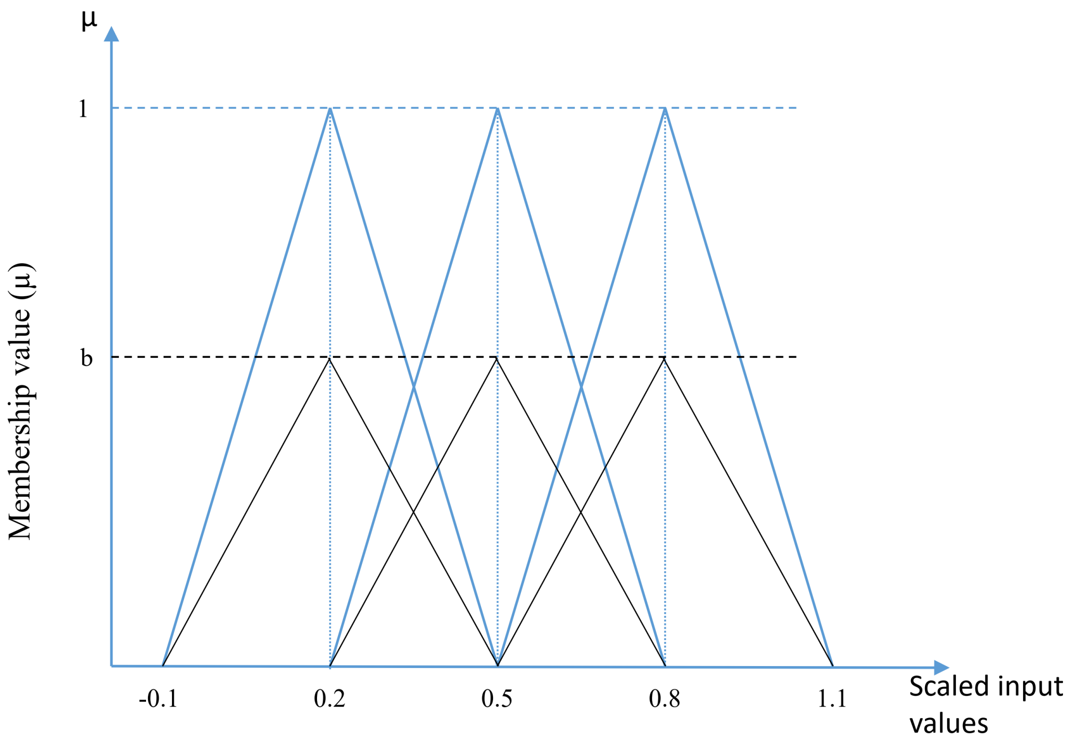

Figure 4.3 : Membership functions for inputs.

The grid points are defined as x∈[1,10] and y∈[1,10], so, 100 training pairs are used. Because of the 3 MFs for every inputs, there are 9 rules. The three methods are tested LSE and Gradient Descent respectively.

As seen from the Tables (4.1,4.2), the methods are tested for linear and constant consequent membership functions. The Root Mean Square Error (RMSE) is used to compare the results.

The results clearly show that classical IT2 KMA based ANFIS errors are higher than T1 FLS ANFIS. For this reason, KMA based ANFIS is not suitable for use in any area. Therefore, IT2 M-KMA and T1 ANFIS results were analyzed comparatively.

Table 4.1 : ANFIS (LSE) results for constant consequent parameters.

T1	b	IT2 (KMA)	IT2 (M-KMA)
0.084906	1	3.248081	0.084906
	0.9	3.248081	0.084868
	0.8	3.248081	0.084740
	0.7	3.248081	0.084486
	0.6	3.248081	0.084064
	0.5	3.248081	0.083415
	0.4	3.248081	0.082476
	0.3	3.248081	0.081271
	0.2	3.248081	0.080460
	0.1	3.248081	0.081320

Table 4.2 : ANFIS (LSE) results for linear consequent parameters.

T1	b	IT2 (KMA)	IT2 (M-KMA)
0.045523	1	3.245881	0.045523
	0.9	3.245881	0.045489
	0.8	3.245881	0.045349
	0.7	3.245881	0.045041
	0.6	3.245881	0.044502
	0.5	3.245881	0.043472
	0.4	3.245881	0.039940
	0.3	3.245881	0.037517
	0.2	3.245881	0.033585
	0.1	3.245881	0.037559

As stated in Tables (4.1, 4.2), the IT2 ANFIS results change with respect to uncertainty. When the b value is 1, the training results between T1 ANFIS and the proposed IT2 ANFIS approaches are similar. The T1 ANFIS uses A1-C1 (Antecedent type-1 MF - Consequent type-1 MF) membership functions and the proposed IT2 ANFIS uses A2-C1 (Antecedent type-2 MF - Consequent type-1 MF) membership functions. When b=1, the IT2 FLS antecedent membership functions turn into T1 FLS membership functions (A1-C1), so, the results must be same for b=1 as seen in Tables (4.1, 4.2). This shows that the system is working properly.

The b value in Figure 4.3 defines the uncertainty. The high b values mean less uncertainty and low b values mean high uncertainty. In the linear model with LSE, the RMSE is 0.045523 for b=1 and the RMSE is 0.033585 for b=0.2. It shows that the proposed IT2 ANFIS gives better results in more uncertainty for least square estimation usage.

Table 4.3 : ANFIS (GD) results for constant consequent parameters with 1000 epochs.

T1	b	IT2 (KMA)	IT2 (M-KMA)
0.084906	1	3.226571	0.084906
	0.9	3.226571	0.084868
	0.8	3.226571	0.084740
	0.7	3.226571	0.084486
	0.6	3.226571	0.084064
	0.5	3.226571	0.083415
	0.4	3.226571	0.082476
	0.3	3.226571	0.081271
	0.2	3.226571	0.080460
	0.1	3.226571	0.081320

Table 4.4 : ANFIS (GD) results for linear consequent parameters with 1000 epochs.

T1	b	IT2 (KMA)	IT2 (M-KMA)
0.067583	1	3.238083	0.067180
	0.9	3.238251	0.067600
	0.8	3.238332	0.067783
	0.7	3.237450	0.067661
	0.6	3.238050	0.067429
	0.5	3.240285	0.067126
	0.4	3.240482	0.067594
	0.3	3.238486	0.066504
	0.2	3.237452	0.065325
	0.1	3.239447	0.065037

The same study is tested for GD method. The GD tuning algorithm is iterated 1000 times with a learning rate 10^{-5}. The learning is changed in every epoch with respect to change in error. If the error is increasing the learning rate is minimized with a defined percent (%10). If the error is minimizing the learning rate is increased with a constant percent (%1). So, the learning becomes faster and can avoid unexpected failures.

The results show that when the consequent parameters are taken as constant, the LSE and GD training results are similar to the proposed IT2 ANFIS approach. However, for linear consequent parameters the LSE method results are better. It is because of the epoch numbers. The constant consequent parameters needs less iteration number and it reaches to minimum RMSE value at a small epoch number. However, the T1 ANFIS and the proposed IT2 ANFIS need more iteration numbers to reach the minimum RMSE value in the linear consequent parameters. It is clear that GD methods have more computational load than LSE, but the GD method is used in control studies.

The results are given in Tables (4.3,4.4). When the results are examined, it is clear that more epochs are needed to have better results. The IT2 ANFIS results are better than T1 ANFIS for constant and linear consequent parameters. The greater the uncertainty, the better the IT2 ANFIS results.

5. COMPARISON OF NEURO FUZZY CONTROLLERS

The Interval Type-2 Fuzzy Logic System (IT2 FLS) is a powerful system for controlling nonlinear systems and overcoming uncertainties. So, there are several IT2FLS control studies in the literature. The Interval Type-2 Fuzzy Logic Controller (IT2 FLC) studies without online parameter training are implemented to various systems [49,52,53] and the results show that the IT2FLC is more effective against uncertainties and nonlinearity.

Tuning of T2 FLS parameters is still a research area and a hard task so there are different tuning methods in the open literature. In 2011, an inverse Interval Type-2 Fuzzy Logic Controller (IT2 FLC) is proposed that the IT2 FLS Takagi-Sugeno (TS) consequent parameters are updated with pure mathematics. So, it does not contain the inverse of IT2 FLS and it works only for SISO systems [50]. In [66,71], another inverse IT2FLC is proposed that the inverse control signal is produced by Big Branch-Big Crunch (BB-BC) algorithm. So, the IT2 FLS is not inverted in this studies but antecedent parameters are trained and consequent parameters are chosen as crisp [55]. In 2015, a sliding mode based tuning for IT2 FLS is proposed by Kayacan that this approach is used on 2 DOF helicopter [58], satellites [38] and quadrotor [15]. In these studies, A2-C1 models are used and consequent parts are trained. The papers' results clearly show that the IT2FLC is better to handle uncertainties and more adaptive to input signal variations [48,54].

In this chapter, the IT2 FLC based controllers are implemented to a SISO system (DC motor). So, the IT2 fuzzy controllers are tested and compared to T1 fuzzy logic controllers.

5.1 Comparison of IT2 NFC and T1 NFC

The same conditions are used for both T1 FLC and IT2 FLC. The IT2 FIS NFC results for a DC motor example are given in Figures (5.1, 5.2) and in Table 5.1.

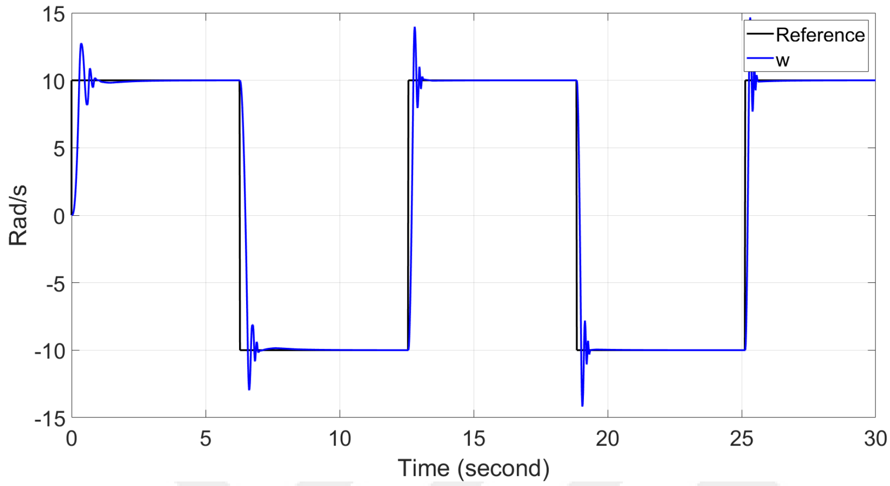

Figure 5.1 : IT2 NFC results for inputs (error, output derivative, sum of error) where w=10.

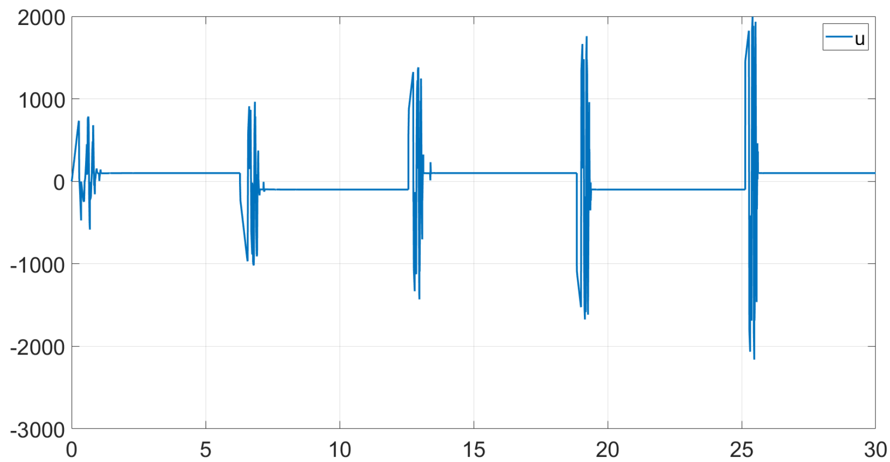

Figure 5.2 : IT2 NFC control signals for inputs (error, output derivative, sum of error) where w=10.

The used IT2 FLS is based on Sugeno FIS. The only difference is membership functions. So, IT2 NFC and Sugeno NFC results are similar. When the results are examined, it is seen that overshoot is similar for both IT2 NFC and T1 NFC. the IT2

Table 5.1 : Comparison of T1 and IT2 NFCs with three inputs (error, output derivative, sum of error) for w=10.

Inputs	%OS	T_s	RMSE	Computation Time (second)
Sugeno NFC	27.1	1.390	0.2815	47.3358
Mamdani NFC	0	7.429	0.6721	888.9318
ŞEN NFC	10.5	2.483	0.3427	72.0989
IT2 NFC (b=1)	27.5	0.830	0.2778	86.3923
IT2 NFC (b=0.5)	27.5	0.830	0.2801	74.3662

NFC is faster than other controllers and has the smallest RMSE values. So, it is clear that the IT2 NFC has better results compared to other controllers.

5.2 Comparison of IT2 FEL and T1 FEL

The IT2 FLC is implemented to FEL controller as performed for the Type-1 FLS. The IT2 lower MFs' peak point "b" is taken as "1" and "0.5". The results are given in Figures (5.3, 5.4).

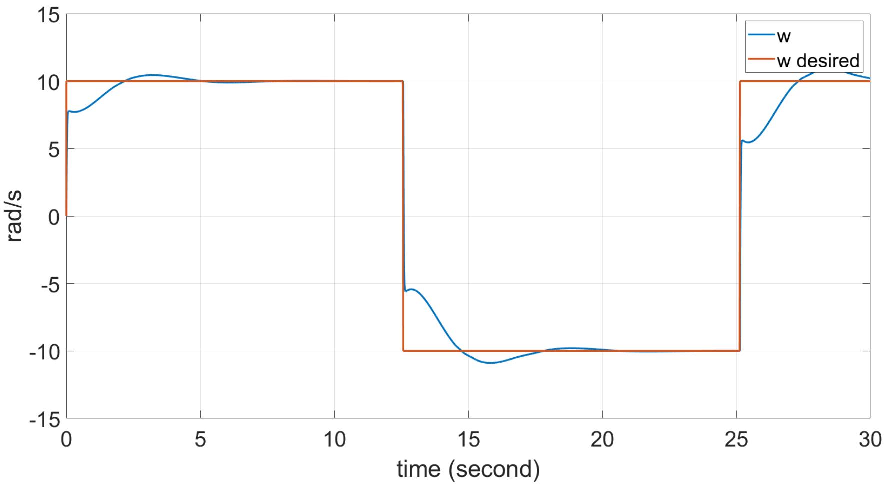

Figure 5.3 : IT2 FEL results where w=10.

Table 5.2 : Comparison of T1 and IT2 FEL controllers for w=10.

Inputs	%OS	T_s	RMSE	Computation Time (second)
PID	4.6	4.911	2.3728	0.4306
T1 FEL (Sugeno)	4.5	4.362	1.4644	3.1239
IT2 FEL (b=1)	4.5	4.372	1.4646	5.1455
IT2 FEL (b=0.5)	4.5	4.367	1.4646	4.7353

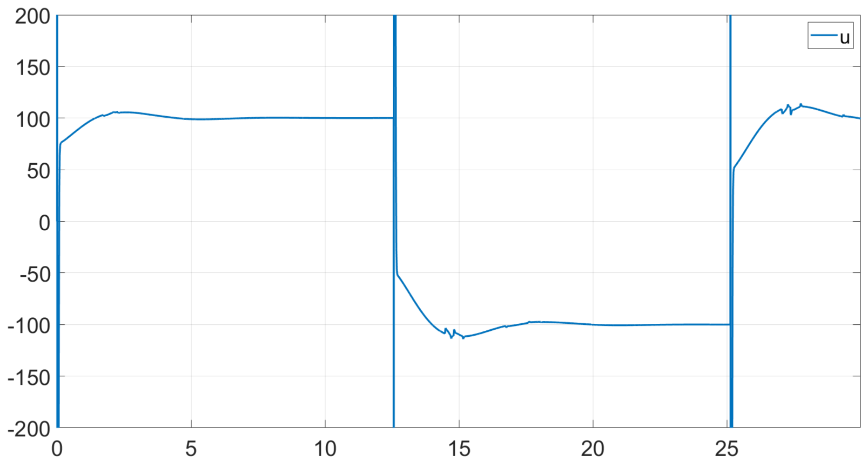

Figure 5.4 : IT2 FEL control signals where w=10.

The T1 and IT2 controllers are tested for FEL controller as seen in Table 5.2. The results show that the T1 FEL and IT2 FEL controller results are similar. This similarity is because of the restrictions. The main controller is a conventional controller and the fuzzy logic is used as secondary controller. So, the effect of the fuzzy is restricted.

What should be considered in this study is the effect of FLC on the classical controller. Thanks to this effect, the system can respond faster with fewer errors and overshoots.

6. NEURO FUZZY CONTROL IMPLEMENTATION ON QUADROTOR MODELLING

In this chapter, previously proposed NFC and FEL controllers are tested on a quadrotor model. Because of that, a quadrotor model is represented. The controller approaches for quadrotors are given and some assumptions are discussed. Finally, disturbance effects on the quadrotor are tested.

The chosen quadrotor model is a T frame model as seen in Figure 6.1. The X axis direction (rotor 1) is accepted as the front side and rotor 3 side is rear. Rotor 2 side is right and rotor 4 side is left. The rotors 1 and 3 rotate counter-clockwise and rotors 2 and 4 rotate clockwise.

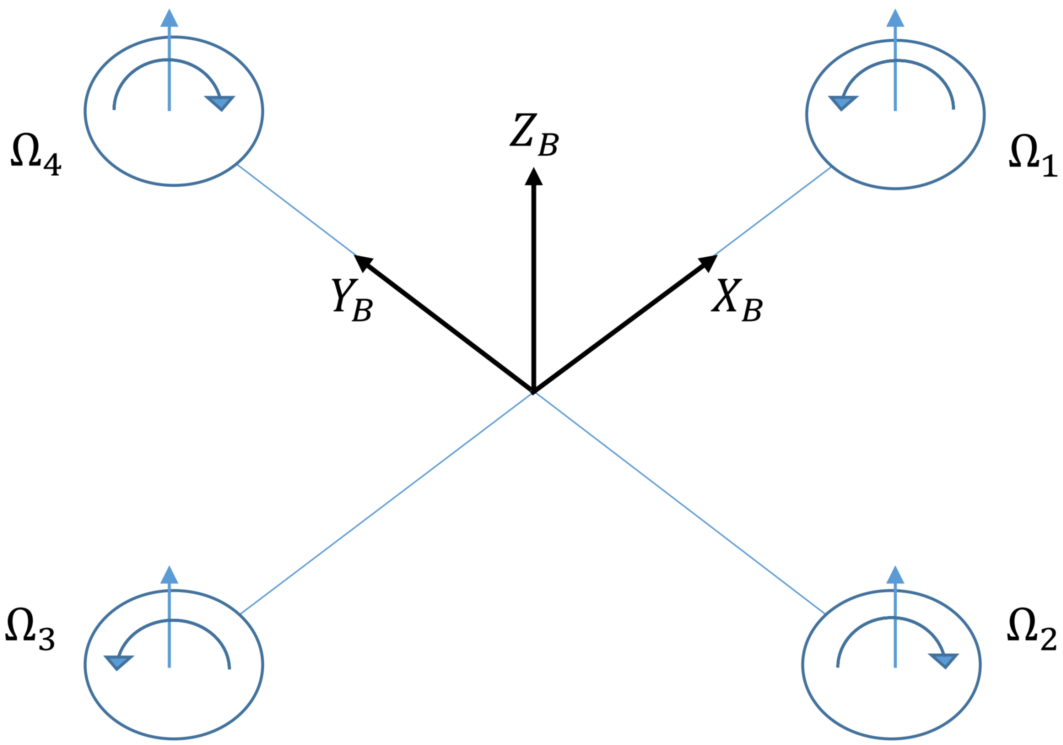

Figure 6.1 : Quadrotor X frame.

Two frames are defined in quadrotor system; the earth inertial frame (E) and body fixed frame (B). Velocities in the B frame along X_B, Y_B, Z_B axes are given as $u, v, w(m/s)$ and the angular rates along the X_B, Y_B, Z_B axes are $p, q, r(rad/s)$ respectively [32].

$$V_B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \eta_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad \nu = \begin{bmatrix} V_B \\ \eta_B \end{bmatrix} \tag{6.1}$$

The position and angle of body fixed frame with respect to earth inertial frame are represented with $x, y, z, \phi, \theta, \psi$ as given in [72].

$$\Gamma = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad \Theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \xi = \begin{bmatrix} \Gamma \\ \Theta \end{bmatrix} \tag{6.2}$$

The rotation matrix for position from B frame to E frame is given in equation 6.3 where $c = cos, s = sin, t = tan$ that the rotation matrix was given in [2] in detail.

$$R = R_E^B = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{6.3}$$

The transformation matrix for angular rates from body fixed frame to earth inertial frame is given as:

$$T = T_E^B = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \tag{6.4}$$

So, the translation between the earth inertial frame and the body fixed frame is given below:

$$\dot{\Gamma} = R_E^B V_B \qquad \dot{\Theta} = T_E^B \eta_B \tag{6.5}$$

6.1 Mathematical Model of Quadrotor

The dynamics of the quadrotor can be defined with Newton's second law. F_E vector defines the forces with respect to earth inertial frame (E) and $F_B(N)$ vector defines the forces with respect to body fixed frame (B). The $\tau_E(Nm)$ vector defines the torques with respect to E frame and τ_b vector defines the torques with respect to B frame.

The force equations are given below:

$$F_B = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T \tag{6.6}$$

$$F = ma \Rightarrow F_E = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \Rightarrow F_E = m\ddot{\Gamma} = m\overset{\cdot}{\overbrace{RV_B}} \Rightarrow RF_B = m\overset{\cdot}{\overbrace{RV_B}} \tag{6.7}$$

$$RF_B = m(\dot{R}V_B + R\dot{V_B}) = m(R[\eta_B x V_B] + R\dot{V_B}) \tag{6.8}$$

$$F_B = m(\dot{V_B} + \eta_B x V_B) \tag{6.9}$$

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} m\dot{u} \\ m\dot{v} \\ m\dot{w} \end{bmatrix} + m \begin{bmatrix} p \\ q \\ r \end{bmatrix} x \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} m\dot{u} \\ m\dot{v} \\ m\dot{w} \end{bmatrix} + m \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix} = \begin{bmatrix} m(\dot{u} + qw - rv) \\ m(\dot{v} + ru - pw) \\ m(\dot{w} + pv - qu) \end{bmatrix} \tag{6.10}$$

where "x" is cross product [73].

The torque equations are given below:

$$\tau_B = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \tag{6.11}$$

$$\tau = I\ddot{\Theta} \Rightarrow \tau_E = I \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \Rightarrow \tau_E = I\overset{\cdot}{\overbrace{T\eta_B}} = T\tau_B \tag{6.12}$$

$$T\tau_B = I(\dot{T}\eta_B + T\dot{\eta_B}) = I(T[\eta_B x \eta_B] + T\dot{\eta_B}) \tag{6.13}$$

$$\tau_B = I\dot{\eta_B} + \eta_B x(I\eta_B) \tag{6.14}$$

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} I_x\dot{p} \\ I_y\dot{q} \\ I_z\dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} x \begin{bmatrix} I_x p \\ I_y q \\ I_z r \end{bmatrix} = \begin{bmatrix} I_x(\dot{p} + I_z qr - I_y qr) \\ I_y(\dot{q} + I_x pr - I_z pr) \\ I_z(\dot{r} + I_y pq - I_x pq) \end{bmatrix} \tag{6.15}$$

where "x" is cross product. So, the equations can be written in matrix form as:

$$F_T = \begin{bmatrix} F_B \\ \tau_B \end{bmatrix} = \begin{bmatrix} mI_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix} \begin{bmatrix} \dot{V_B} \\ \dot{\eta_B} \end{bmatrix} + \begin{bmatrix} \eta_B x(mV_B) \\ \eta_B x(I\eta_B) \end{bmatrix} = M_B\dot{v} + C_B v \tag{6.16}$$

where;

$$v = \begin{bmatrix} V_B \\ \eta_B \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}, \qquad M_B = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \tag{6.17}$$

$$C_B = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}q \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix} \tag{6.18}$$

In these equations, the relation between the forces, torques and the quadrotor velocities, rotation are examined. The forces and torques are produced by three components those are gravity, gyroscopic effects and thrust (motors).

$$M_B\dot{v} + C_B v = F_T = G_B + O_B\Omega + E_B\Omega^2 \tag{6.19}$$

$G_B(\frac{m^3}{kgs^2})$ is gravity vector, 0_B is gyroscopic propeller matrix, E_B is thrust and torque matrix generated by the rotors and $\Omega(rad/s)$ is propellers' speeds.

The gravitational force only affects the linear equations. The rotational parameters are not affected from gravity. So, the gravity vector can be written as;

$$G_B = \begin{bmatrix} F_B \\ 0_{3x1} \end{bmatrix} = \begin{bmatrix} R^{-1}F_E \\ 0_{3x1} \end{bmatrix} = \begin{bmatrix} R^{-1} \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ 0_{3x1} \end{bmatrix} = \begin{bmatrix} R^{T} \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ 0_{3x1} \end{bmatrix} = \begin{bmatrix} mgs_\theta \\ -mgc_\theta s_\phi \\ -mgc_\theta s_\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.20}$$

The rotation matrix is normalized orthogonal matrix [73] that R^{-1} is equal to R^T.

The gyroscopic force only affects the rotational equations. The propellers are perpendicular to X_B and Y_B axes. So, the gyroscopic force affects only the z axis.

$$O_B = \begin{bmatrix} 0_{3x1} \\ -J_r \begin{bmatrix} \eta_B x \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} -\Omega_1 \\ +\Omega_2 \\ -\Omega_3 \\ +\Omega_4 \end{bmatrix}^T \end{bmatrix} = \begin{bmatrix} 0_{3x1} \\ -J_r \begin{bmatrix} q1 - r0 \\ r0 - p1 \\ p0 - q0 \end{bmatrix} \begin{bmatrix} -\Omega_1 \\ +\Omega_2 \\ -\Omega_3 \\ +\Omega_4 \end{bmatrix}^T \end{bmatrix} \Rightarrow \tag{6.21}$$

$$O_B = \begin{bmatrix} 0_{3x1} \\ J_r \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \begin{bmatrix} +\Omega_1 \\ -\Omega_2 \\ +\Omega_3 \\ -\Omega_4 \end{bmatrix}^T \end{bmatrix} = J_r \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -q & q & -q & q \\ p & -p & p & -p \\ 0 & 0 & 0 & 0 \end{bmatrix} \Omega = J_r \begin{bmatrix} 0 \\ 0 \\ 0 \\ -q\Omega_r \\ p\Omega_r \\ 0 \end{bmatrix} \tag{6.22}$$

where O_B is the gyroscopic propeller matrix, J_r is the total rotational moment of inertia around the propeller axis and $\Omega(rad/s)$ is the propellers' speed.

Propeller's speed (Ω) is given as:

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \tag{6.23}$$

The overall rotor speeds can be given as:

$$\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \tag{6.24}$$

The force and torque those produced by actuators are the third part of the forces affecting the aircraft. The moment equation is given as:

$$E_B \Omega^2 = \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ bl(\Omega_4^2 - \Omega_2^2) \\ bl(\Omega_3^2 - \Omega_1^2) \\ -d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \tag{6.25}$$

$b(Ns^2)$ is thrust coefficient, $d(Nms^2)$ is drag coefficient and $l(m)$ is the distance between the quadrotor center and the propeller center. $U_1(N)$ is the total force produced by propellers that is the total of all propellers' thrust. $U_2(Nm)$ is the torque around the X_B axis (roll motion). So, right (Ω_2) and left (Ω_4) propellers affect the $U_2(Nm)$ value. $U_3(Nm)$ is the torque around the Y_B axis (pitch motion).So, front (Ω_1) and rear (Ω_3) propellers affect the $U_3(Nm)$ value. $U_4(Nm)$ is the torque around the Z_B axis (yaw motion). All the propellers affect the torque value around the Z_B axis.

From the equation 6.19, it can be written as;

$$M_B\dot{v} = -C_B v + G_B + O_B\Omega + E_B\Omega^2 \tag{6.26}$$

$$\dot{v} = {M_B}^{-1}(-C_B v + G_B + O_B\Omega + E_B\Omega^2) \tag{6.27}$$

$$\dot{v} = {M_B}^{-1}(-\begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}q \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} + \begin{bmatrix} mgs_\theta \\ -mgc_\theta s_\phi \\ -mgc_\theta s_\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} + O_B\Omega + E_B\Omega^2) \tag{6.28}$$

$$\dot{v} = {M_B}^{-1}(-\begin{bmatrix} mwq - mvr \\ -mwp + mur \\ mvp - muq \\ I_{zz}qr - I_{yy}qr \\ -I_{zz}pr + I_{xx}pr \\ I_{yy}qp - I_{xx}pq \end{bmatrix} + \begin{bmatrix} mgs_\theta \\ -mgc_\theta s_\phi \\ -mgc_\theta s_\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -J_r q\Omega_r \\ J_r p\Omega_r \\ 0 \end{bmatrix} + E_B\Omega^2) \tag{6.29}$$

$$\dot{v} = \begin{bmatrix} \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} (\begin{bmatrix} -mwq + mvr + mgs_\theta \\ mwp - mur - mgc_\theta s_\phi \\ -mvp + muq - mgc_\theta s_\phi \\ -I_{zz}qr + I_{yy}qr - J_r q\Omega_r \\ I_{zz}pr - I_{xx}pr + J_r p\Omega_r \\ -I_{yy}pq + I_{xx}pq \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}) \tag{6.30}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(-mwq + mvr + mgs_\theta) \\ \frac{1}{m}(mwp - mur - mgc_\theta s_\phi) \\ \frac{1}{m}(-mvp + muq - mgc_\theta s_\phi + U_1) \\ \frac{1}{I_{xx}}(-I_{zz}qr + I_{yy}qr - J_r q\Omega_r + U_2) \\ \frac{1}{I_{yy}}(I_{zz}pr - I_{xx}pr + J_r p\Omega_r + U_3) \\ \frac{1}{I_{zz}}(-I_{yy}pq + I_{xx}pq + U_4) \end{bmatrix} \tag{6.31}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (vr - wq) + gs_\theta \\ (wp - ur) - gc_\theta s_\phi \\ (uq - vp) - gc_\theta s_\phi + \frac{U_1}{m} \\ \frac{I_{yy} - I_{zz}}{I_{xx}}qr - \frac{J_r}{I_{xx}}q\Omega_r + \frac{U_2}{I_{xx}} \\ \frac{I_{zz} - I_{xx}}{I_{yy}}pr + \frac{J_r}{I_{yy}}p\Omega_r + \frac{U_3}{I_{yy}} \\ \frac{I_{xx} - I_{yy}}{I_{zz}}pq + \frac{U_4}{I_{zz}} \end{bmatrix} \tag{6.32}$$

The equation 6.32 defines the quadrotor equations with respect to body-fixed frame. The linear equations must be defined with respect to earth-inertial frame. So, a new frame is called as "H" that the linear equations are given with respect to earth inertial frame and rotational equations are given with respect to body-fixed frame [32].

$$\zeta = \begin{bmatrix} \dot{\Gamma} \\ \eta_B \end{bmatrix} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ p \\ q \\ r \end{bmatrix} \tag{6.33}$$

$$M_H \dot{\zeta} + C_H \zeta = G_H + O_H \Omega + E_H \Omega^2 \tag{6.34}$$

The M matrix will not change. So, the same matrix will be used.

$$M_H = M_B = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \tag{6.35}$$

The Coriolis matrix is changed. The linear effects will be zero with respect to earth-inertial frame.

$$C_H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}q \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix} \tag{6.36}$$

The gravitational force only affects the linear equations in z direction when designed with respect to the earth-inertial frame.

$$G_H = \begin{bmatrix} 0 \\ 0 \\ -mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.37}$$

The gyroscopic force only affects the rotational equations. The rotational frames are the same as body fixed frame. So, the gyroscopic matrix is the same.

$$O_H = J_r \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -q & q & -q & q \\ p & -p & p & -p \\ 0 & 0 & 0 & 0 \end{bmatrix} \Omega \tag{6.38}$$

The force and torque matrix was composed of the forces and torques. The torques are still on the same axes (X_B, Y_B, Z_B). So, the torque values (U_2, U_3, U_4) will not change. The thrust along the Z_B axis will be converted to Z_E axis. So, the U_1 will affect the three axes.

$$E_H \Omega^2 = \begin{bmatrix} R_B^E & O_{3x3} \\ O_{3x3} & I_{3x3} \end{bmatrix} E_B \Omega^2 \tag{6.39}$$

$$E_H \Omega^2 = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi & 0 & 0 & 0 \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi & 0 & 0 & 0 \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{6.40}$$

$$E_H \Omega^2 = \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi) U_1 \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi) U_1 \\ (c_\theta c_\phi) U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi) b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi) b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ (c_\theta c_\phi) b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ bl(\Omega_4^2 - \Omega_2^2) \\ bl(\Omega_3^2 - \Omega_1^2) \\ -d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \tag{6.41}$$

The same equation structure with the Equation 6.27 is implemented and the resulting equation is given.

$$\begin{bmatrix} \ddot{X}_E \\ \ddot{Y}_E \\ \ddot{Z}_E \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi) \frac{U_1}{m} \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi) \frac{U_1}{m} \\ -g + (c_\theta c_\phi) \frac{U_1}{m} \\ \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{J_r}{I_{xx}} q \Omega_r + \frac{U_2}{I_{xx}} \\ \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{J_r}{I_{yy}} p \Omega_r + \frac{U_3}{I_{yy}} \\ \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}} \end{bmatrix} \tag{6.42}$$

6.2 Quadrotor Open Loop

The quadrotor open loop structure is given in Figure 6.2 that inputs are the propeller speeds and the output is quadrotor position and rotation.

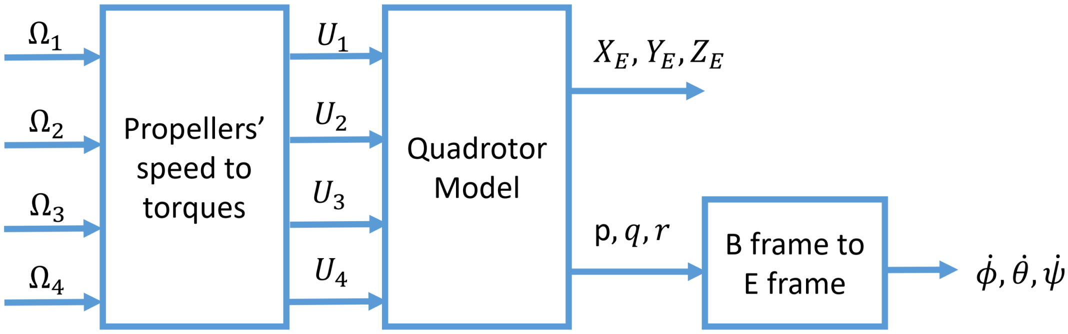

Figure 6.2 : Open loop quadrotor system.

The quadrotor parameters used as an example are given in Table 6.1 [15].

Table 6.1 : Quadrotor parameters.

Parameters	Description	Value	Unit
I_{xx}	Moment of inertia about the X-axis	0.007	$kg \cdot m^2$
I_{yy}	Moment of inertia about the Y-axis	0.007	$kg \cdot m^2$
I_{zz}	Moment of inertia about the Z-axis	0.012	$kg \cdot m^2$
J_r	Rotor moment of inertia	$6.5 \cdot 10^{-5}$	$kg \cdot m^2$
b	Thrust coefficient	$4.13 \cdot 10^{-5}$	$N \cdot s^2$
d	Drag coefficient	$8.5 \cdot 10^{-7}$	$N \cdot m \cdot s^2$
l	Distance from propellers to the center	0.17	m
m	Mass	0.68	kg
g	Gravitational constant	9.81	m/s^2

The three rotation motions are named as roll (around X_B axis), pitch (around Y_B axis) and yaw (around Z_B axis). The propeller speeds' effect on the motions are given as;

Roll motion is provided by changing the left Ω_4 and right Ω_2 propellers' speeds. For the given quadrotor parameters, when the propellers' speed is 201 rad/s, the quadrotor will be in hovering condition. When the all propellers' speed are increased to 202, the quadrotor will rise as seen in Figures (6.3, 6.4).

For roll motion the propeller's speed is defined as; $\Omega_1 = 201, \Omega_2 = 200, \Omega_3 = 201, \Omega_4 = 202$. So, the left propeller speed is increased and the right propeller speed is decreased. So, the quadrotor will do rotation about the X_B axis as shown in Figure

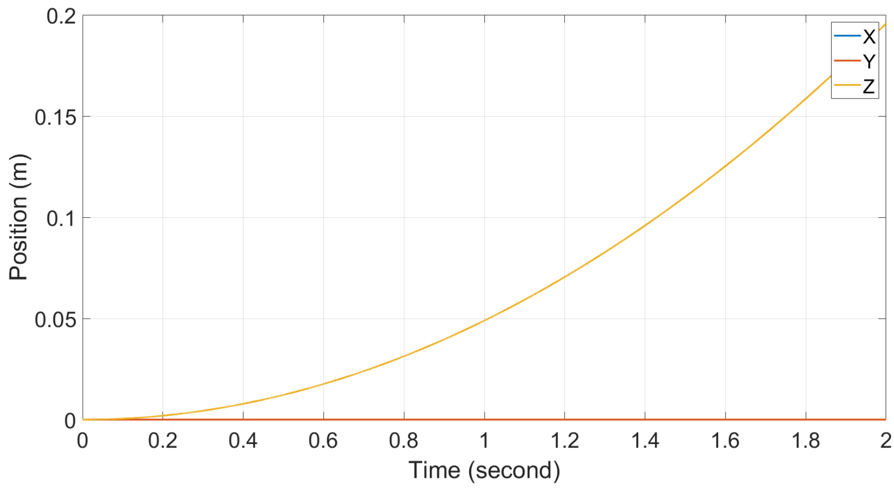

Figure 6.3 : Quadrotor position for $\Omega_1 = 202, \Omega_2 = 202, \Omega_3 = 202, \Omega_4 = 202$.

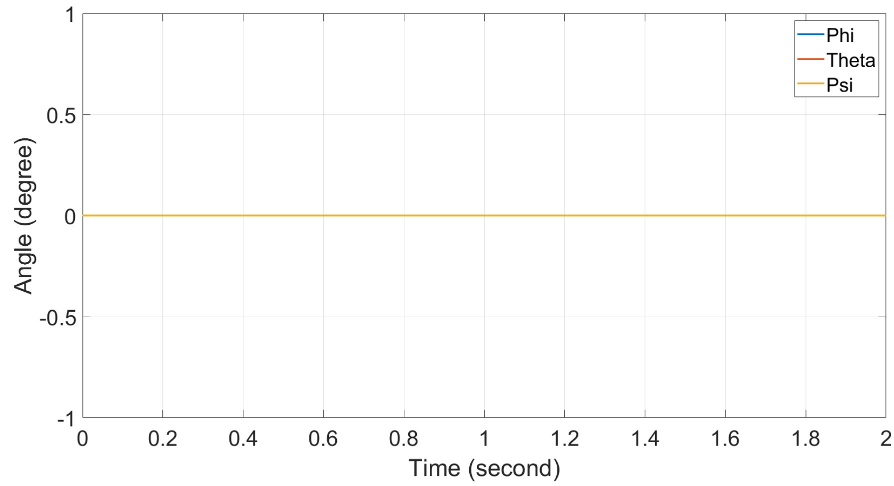

Figure 6.4 : Quadrotor rotation for $\Omega_1 = 202, \Omega_2 = 202, \Omega_3 = 202, \Omega_4 = 202$.

(6.6). In this motion, the quadrotor moves in the $-Y_B$ direction. In rolling motion, the quadrotor uses some of the forces for rolling and so the thrust in Z_B direction becomes less. For this reason, the altitude of the quadrotor will decrease as stated in Figure (6.5).

Pitch motion is provided by changing the front Ω_1 and rear Ω_2 propellers' speeds. For the pitch motion the propeller's speed is defined as; $\Omega_1 = 200, \Omega_2 = 201, \Omega_3 = 202, \Omega_4 = 201$. So, the front propeller speed is decreased and the rear propeller speed is increased. So, the quadrotor will do rotation about the Y_B axis as shown in Figure (6.8).

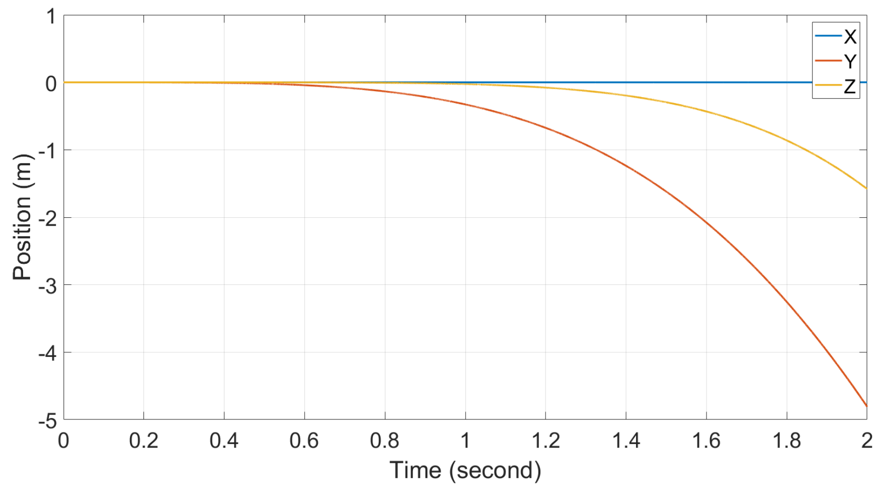

Figure 6.5 : Quadrotor position for $\Omega_1 = 201, \Omega_2 = 200, \Omega_3 = 201, \Omega_4 = 202$ (roll motion).

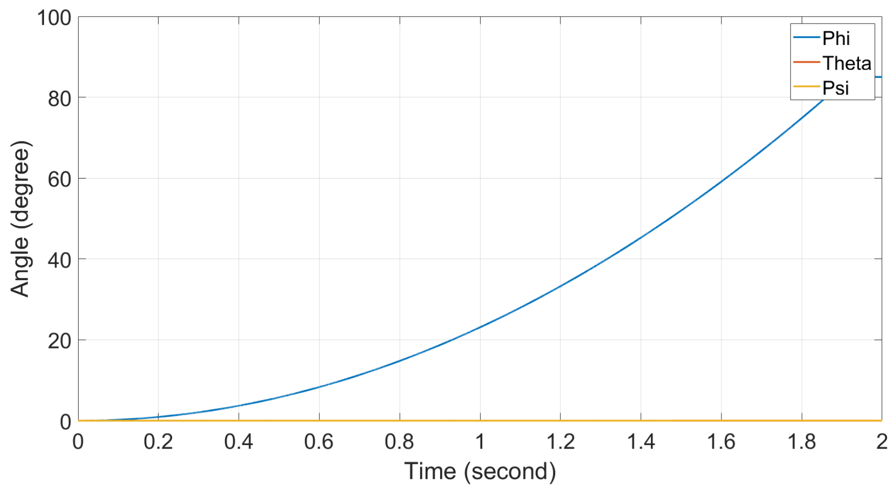

Figure 6.6 : Quadrotor rotation for $\Omega_1 = 201, \Omega_2 = 200, \Omega_3 = 201, \Omega_4 = 202$ (roll motion).

In this motion, the quadrotor moves in the X_B direction. In the pitching motion, the quadrotor uses some of the forces for pitching and so the thrust in Z_B direction becomes less. For this reason, the altitude of the quadrotor will decrease as stated in Figure (6.7).

Yaw motion is provided by changing the propellers correlatively that the Ω_1, Ω_3 couple rotates negatively to Ω_2, Ω_4 couple. When the couple propellers' speed changed the quadrotor rotates about the Z_B axis. For the yaw motion the propeller's speed is defined as; $\Omega_1 = 200, \Omega_2 = 202, \Omega_3 = 200, \Omega_4 = 202$. The quadrotor will rotate about the Z_B axis as shown in Figure (6.10). In this motion, the quadrotor does not move in any direction as seen in Figure (6.9).

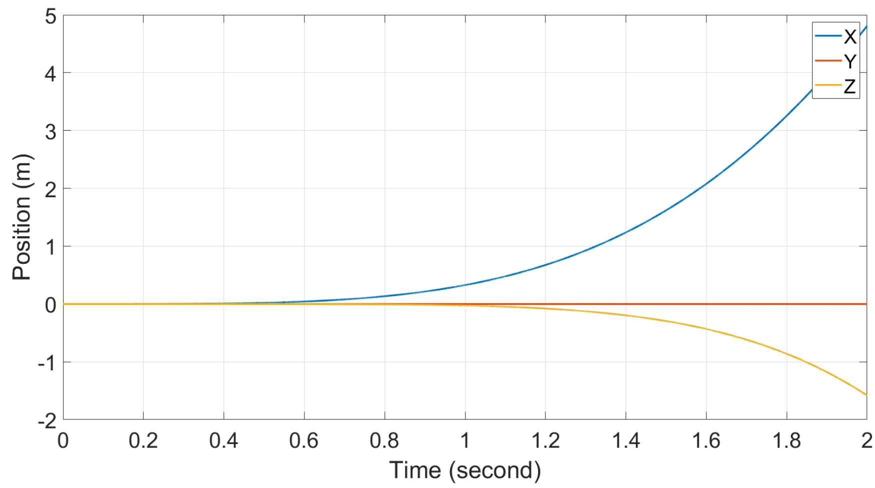

Figure 6.7 : Quadrotor position for $\Omega_1 = 200, \Omega_2 = 201, \Omega_3 = 202, \Omega_4 = 201$ (pitch motion).

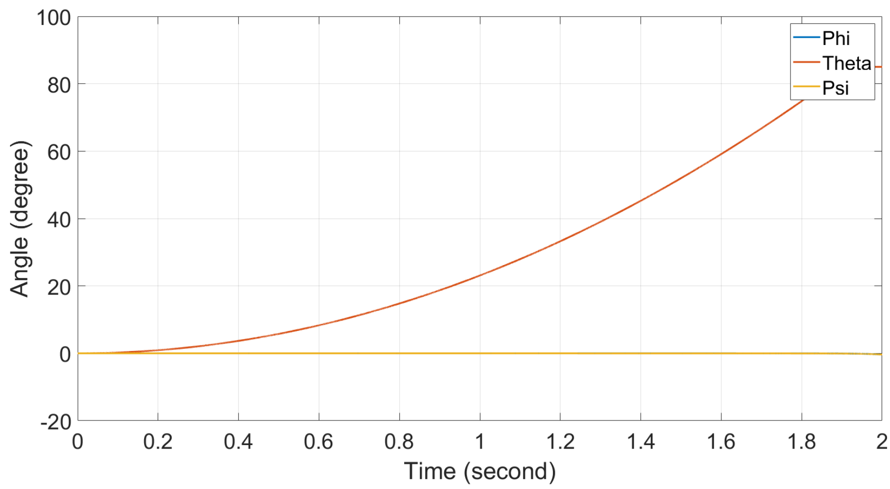

Figure 6.8 : Quadrotor rotation for $\Omega_1 = 200, \Omega_2 = 201, \Omega_3 = 202, \Omega_4 = 201$ (pitch motion).

In this example, the right and left propellers' speeds is bigger than front and rear propellers' speeds. So, the faster rotating propellers produce more gyroscopic effect. The left and right propellers turn around at clockwise direction. The effect will be anticlockwise. It means that the quadrotor will turn in "$+$" direction as seen in Figure (6.10). The quadrotor controller consists of two stages, state control and position control.

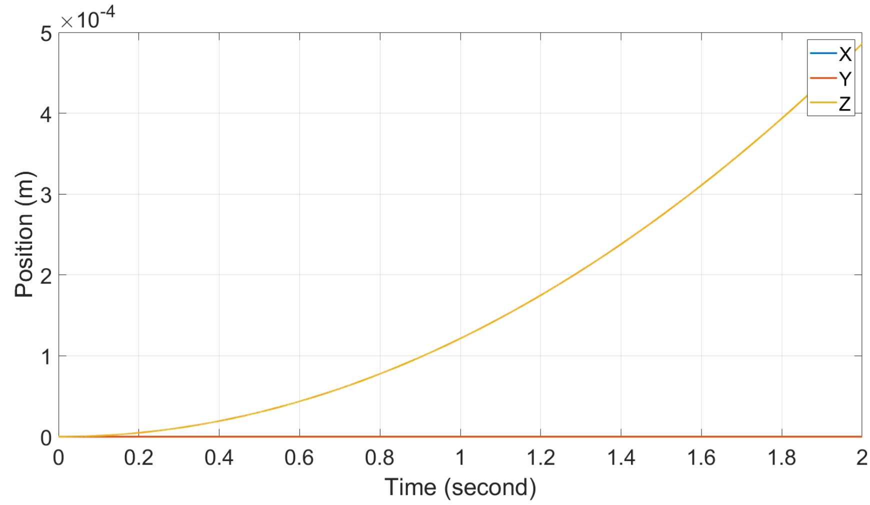

Figure 6.9 : Quadrotor position for $\Omega_1 = 200, \Omega_2 = 202, \Omega_3 = 200, \Omega_4 = 202$ (yaw motion).

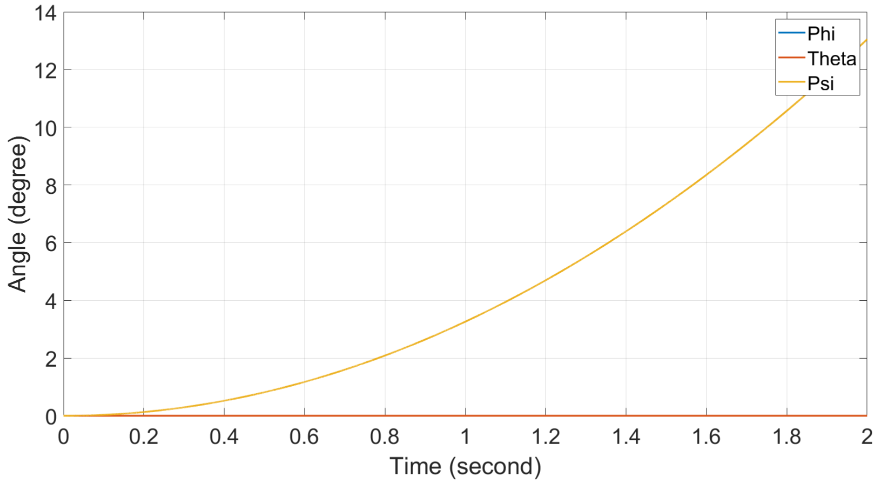

Figure 6.10 : Quadrotor rotation for $\Omega_1 = 200, \Omega_2 = 202, \Omega_3 = 200, \Omega_4 = 202$ (yaw motion).

6.3 Quadrotor Attitude Control

The quadrotor attitude control scheme is given in Figure (6.11). The error between desired Euler angles and the results are used to calculate the needed force and torques. In the previous section, the relation from the propellers' speed to forces was given. Similar work was done for the inverse relationship. The transition equations from forces to propeller speeds are got from the equation 6.25.

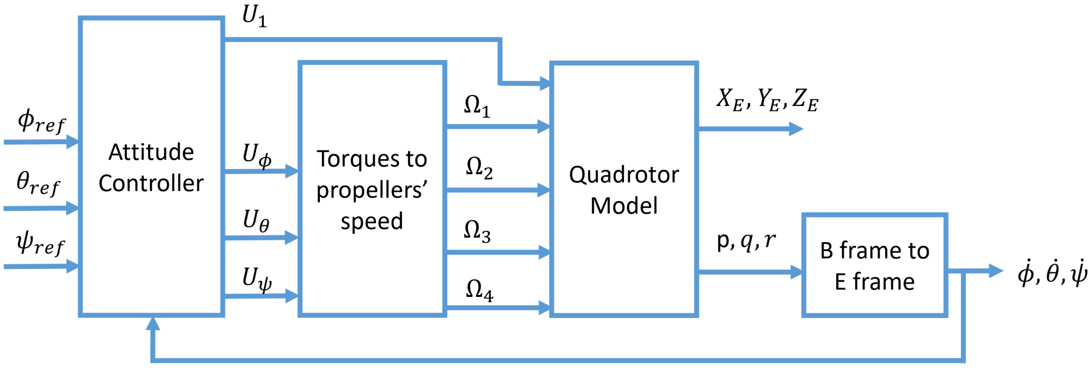

Figure 6.11 : Quadrotor attitude control scheme.

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b}U_1 - \frac{1}{2bl}U_3 - \frac{1}{4d}U_4 \\ \frac{1}{4b}U_1 - \frac{1}{2bl}U_2 + \frac{1}{4d}U_4 \\ \frac{1}{4b}U_1 + \frac{1}{2bl}U_3 - \frac{1}{4d}U_4 \\ \frac{1}{4b}U_1 + \frac{1}{2bl}U_2 + \frac{1}{4d}U_4 \end{bmatrix} \tag{6.43}$$

There are different approaches for quadrotor attitude control in literature. In this thesis, the equation 6.42 is used to define attitude controllers and force equation. The used force and controller equations for a PD controller are given below. The force equation is exactly the inverse of the forward equation. Some assumptions are used for the angular control signals [74].

$$\begin{bmatrix} U_1 \\ U_\phi \\ U_\theta \\ U_\psi \end{bmatrix} = \begin{bmatrix} (K_{z,p}(z_{ref} - z_E) + K_{z,d}(\dot{z}_{ref} - \dot{z}_E) + g)\frac{m}{c_\phi c_\theta} \\ (K_{\phi,p}(\phi_{ref} - \phi) + K_{\phi,d}(\dot{\phi}_{ref} - \dot{\phi}))I_{xx} \\ (K_{\theta,p}(\theta_{ref} - \theta) + K_{\theta,d}(\dot{\theta}_{ref} - \dot{\theta}))I_{yy} \\ (K_{\psi,p}(\psi_{ref} - \psi) + K_{\psi,d}(\dot{\psi}_{ref} - \dot{\psi}))I_{zz} \end{bmatrix} \tag{6.44}$$

where $K_{z,p}, K_{\phi,p}, K_{\theta,p}, K_{\psi,p}$ are proportional controller coefficients and $K_{z,d}, K_{\phi,d}, K_{\theta,d}, K_{\psi,d}$ are derivative controller coefficients.

Table 6.2 : Quadrotor PD coefficients

Parameters	K_p	K_d
z	3	1
ϕ	5	1
θ	5	1
ψ	5	1

The PD, NFC and FEL controllers are used for quadrotor control. The PD coefficients are given as stated in Table 6.2. The integral coefficient leads to oscillation and so it must be small. So, the integral controller is not used for attitude.

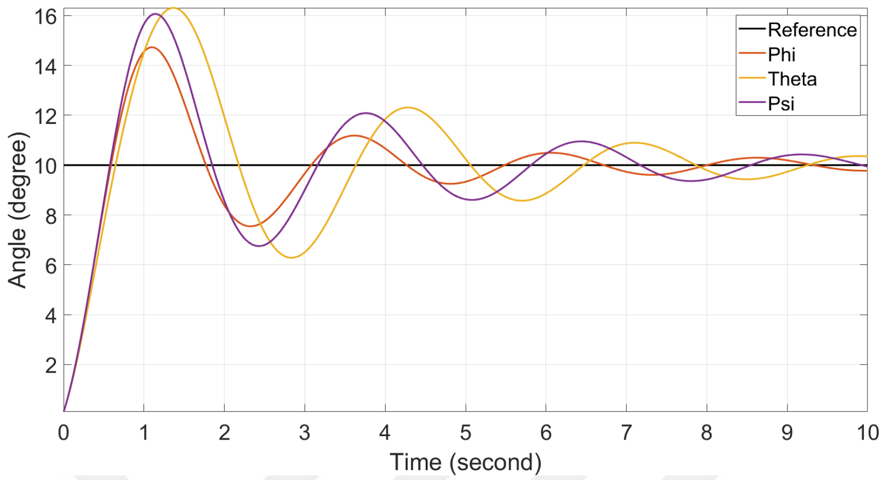

Figure 6.12 : Quadrotor attitude PD control results.

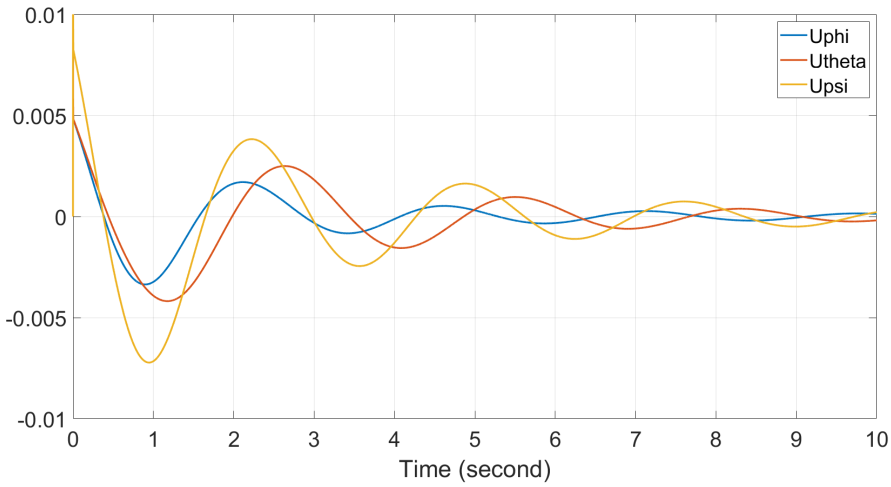

Figure 6.13 : PD control signals.

The Sugeno based Neuro-Fuzzy Controller (NFC) with three inputs (error, output derivative, sum of errors) is used in quadrotor attitude control. The NFC has no overshoot and settling time is small. However, control signals are very wavy. So, the NFC was used with PD. The results are given in Table 6.3 that NFC has no overshoot and has the smallest settling time. the NFC has the biggest mathematical load. The FEL has minimum RMSE values and computation time is less when compared to NFC results. The FEL is better in Settling time and overshoot than PD controller.

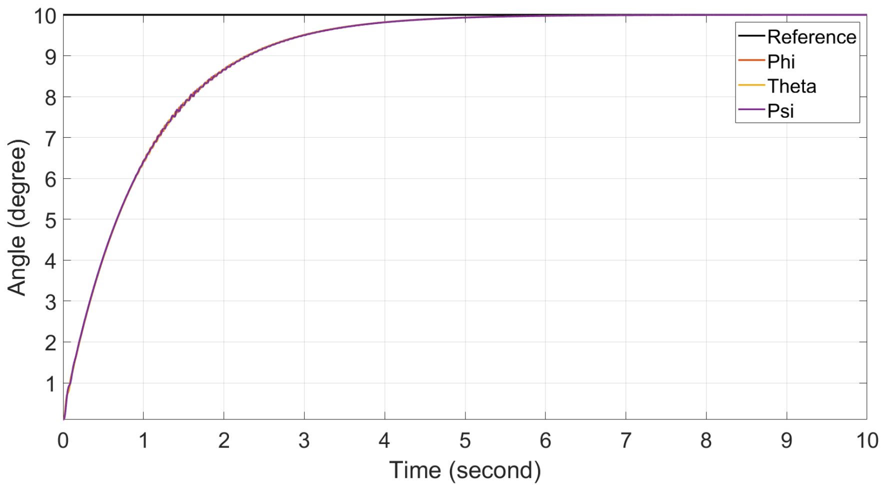

Figure 6.14 : Quadrotor attitude NFC results.

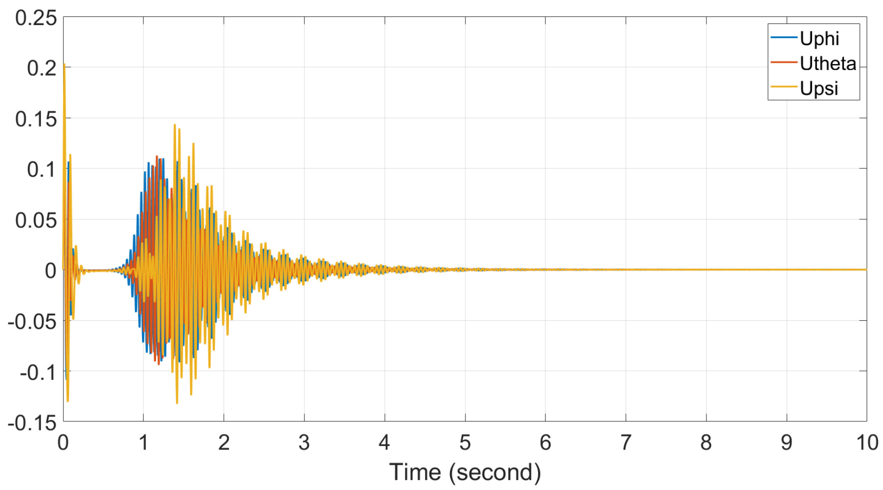

Figure 6.15 : NFC control signals.

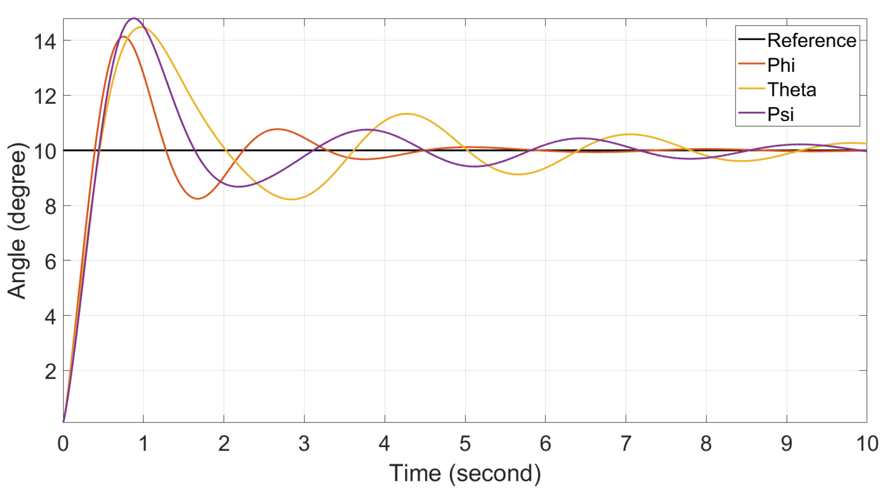

Figure 6.16 : Quadrotor attitude FEL control results.

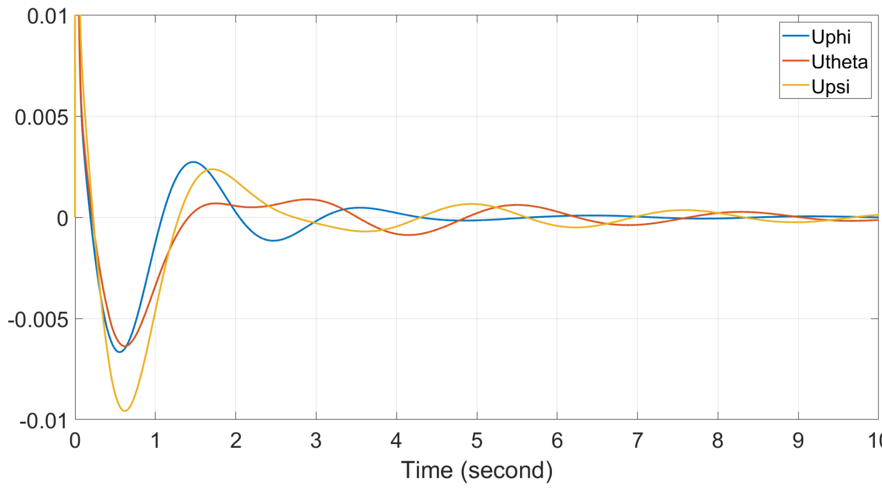

Figure 6.17 : FEL control signals.

Table 6.3 : Comparison of NFC, PD and FEL.

Inputs	%OS	T_s	RMSE (ϕ)	RMSE (θ)	RMSE (ψ)	Computation Time (second)
PD	63.1	11.510	0.0351	0.0461	0.0417	0.3480
T1 NFC (Sugeno)	0	3.907	0.0383	0.0384	0.0383	4.5641
T1 FEL (Sugneo)	48	10.140	0.0266	0.0322	0.0307	1.6181

6.4 Quadrotor Position Control

The quadrotor position control scheme is given in Figure (6.18). The desired position is given to the quadrotor. The quadrotor defines the necessary angles to reach the desired position. The position controller block consists of two stages as shown in Figure (6.19). The first stage represents the controller and the second stage represents the inverse kinematic.

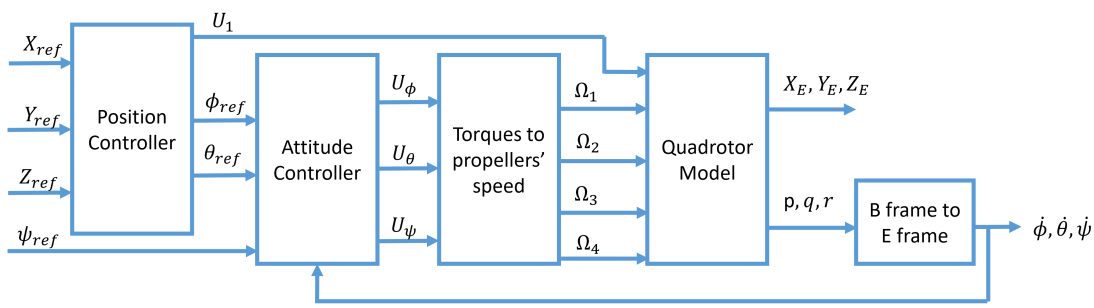

Figure 6.18 : Quadrotor position control scheme.

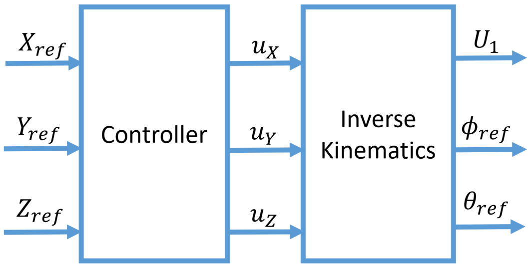

Figure 6.19 : Quadrotor position controller block.

Inverse kinematic I: In the open literature, there are different ways to define the inverse kinematic. In [72], the inverse kinematic equations are derived from equation 6.42. To show simpler, the equations are written in matrix form as given below.

$$\begin{bmatrix} \ddot{X}_E \\ \ddot{Y}_E \\ \ddot{Z}_E \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{U_1}{m} \begin{bmatrix} s_\psi s_\phi + c_\psi s_\theta c_\phi \\ -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ c_\theta c_\phi \end{bmatrix} \tag{6.45}$$

$$U_1 \begin{bmatrix} s_\psi s_\phi + c_\psi s_\theta c_\phi \\ -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ c_\theta c_\phi \end{bmatrix} = \begin{bmatrix} m\ddot{X}_E \\ m\ddot{Y}_E \\ m(\ddot{Z}_E + g) \end{bmatrix} = R^T(3,:) \begin{bmatrix} m\ddot{X}_E \\ m\ddot{Y}_E \\ m(\ddot{Z}_E + g) \end{bmatrix} \tag{6.46}$$

$$U_1 = m[\ddot{X}_E(s_\psi s_\phi + c_\psi s_\theta c_\phi) + \ddot{Y}_E(-c_\psi s_\phi + s_\psi s_\theta c_\phi) + (\ddot{Z}_E + g)(c_\theta c_\phi)] \tag{6.47}$$

$$\phi_{ref} = arcsin\left(\frac{\ddot{X}_E s_\psi - \ddot{Y}_E c_\psi}{\ddot{X}_E^2 + \ddot{Y}_E^2 + (\ddot{Z}_E + g)^2}\right) \quad \theta_{ref} = arctan\left(\frac{\ddot{X}_E c_\psi + \ddot{Y}_E s_\psi}{\ddot{Z}_E + g}\right) \tag{6.48}$$

The inverse kinematic equations can be written as;

$$U_1 = m[u_x(s_\psi s_\phi + c_\psi s_\theta c_\phi) + u_y(-c_\psi s_\phi + s_\psi s_\theta c_\phi) + (u_z + g)(c_\theta c_\phi)] \tag{6.49}$$

$$\phi_{ref} = arcsin\left(\frac{u_x s_\psi - u_y c_\psi}{u_x^2 + u_y^2 + (u_z + g)^2}\right) \quad \theta_{ref} = arctan\left(\frac{u_x c_\psi + u_y s_\psi}{u_z + g}\right) \tag{6.50}$$

Inverse kinematic II: In [15], the inverse kinematic equations are derived as given below. In this approach, Newton's second rule has been taken into account.

$$U_1 = m\sqrt{u_x^2 + u_y^2 + (u_z + g)^2} \tag{6.51}$$

$$\phi_{ref} = arcsin\left(-\frac{mu_y}{U_1}\right) \quad \theta_{ref} = arctan\left(\frac{u_x}{u_z + g}\right) \tag{6.52}$$

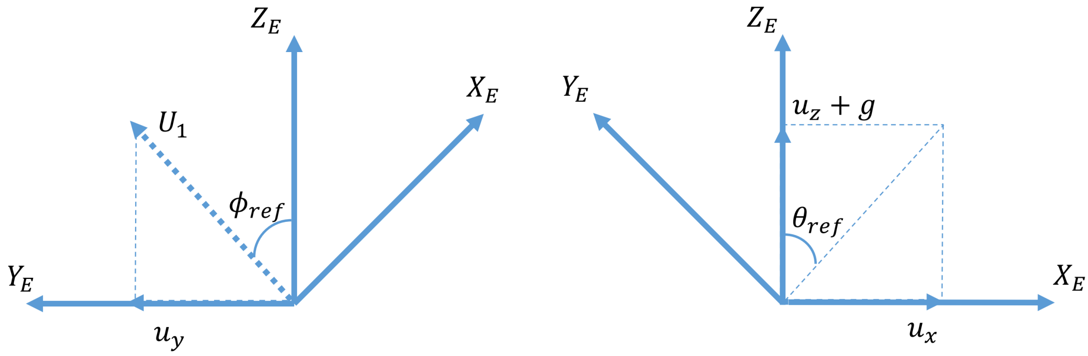

Figure 6.20 : Quadrotor inverse kinematic calculation.

For the ϕ_{ref} angle two force vectors are considered in Z_E direction and Y_E direction. The sin function of the vectors defines the ϕ_{ref} angle.

$$\phi_{ref} = arcsin\left(-\frac{mu_y}{m\sqrt{u_x^2 + u_y^2 + (u_z + g)^2}}\right) = arcsin\left(-\frac{u_y}{\sqrt{u_x^2 + u_y^2 + (u_z + g)^2}}\right) \tag{6.53}$$

In the ϕ_{ref} angle calculation, the sinus defines the force around the X_E axis. In the θ_{ref} angle calculation, the tangent defines the force around the Y_E axis. In this thesis, the second inverse kinematic approach is used. The used PID coefficients are given in Table 6.4. The position control of the quadrotor is tested for PID, Neuro-Fuzzy Controller (NFC) and ANFIS based Feedback Error Learning (FEL). The NFC control signals are very wavy and so it is not compared to others.

Table 6.4 : Quadrotor position control PID coefficients

Parameters	K_p	K_i	K_d
x	3	0.13	1.1
y	3	0.13	1.1
z	3	0.13	1.1
ϕ	6	0.6	10
θ	6	0.6	10
ψ	6	0.6	10

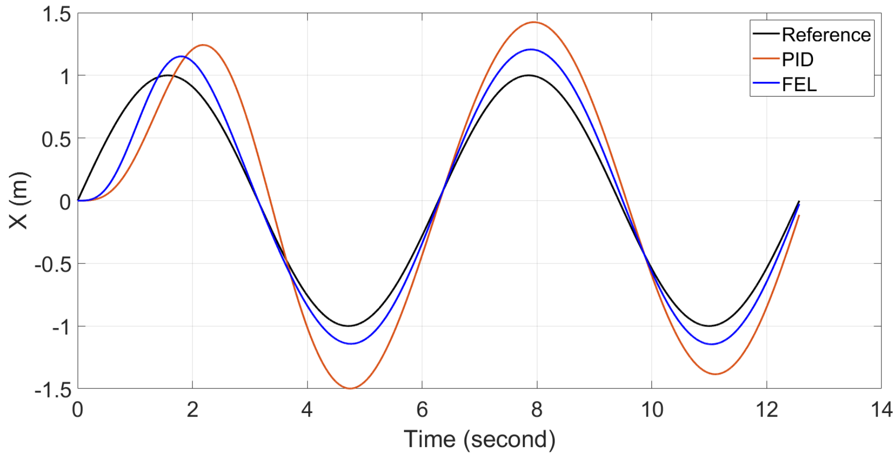

Figure 6.21 : Trajectory tracking performance on X axis of the different controllers.

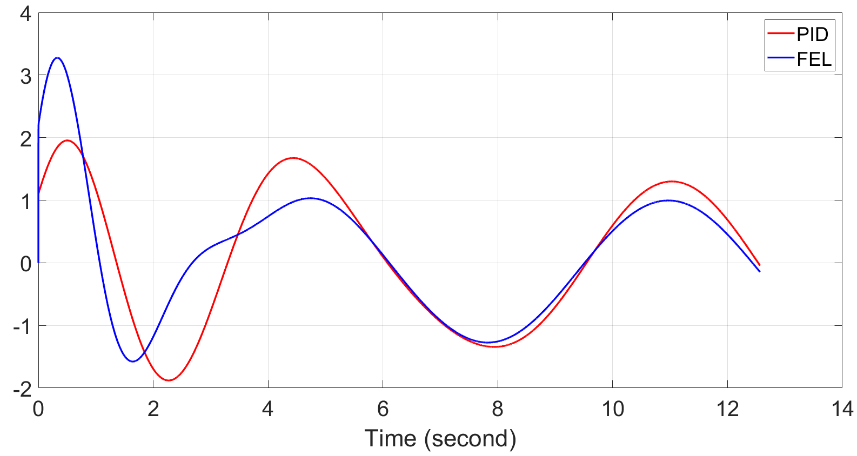

Figure 6.22 : PID and FEL control signals for X axis.

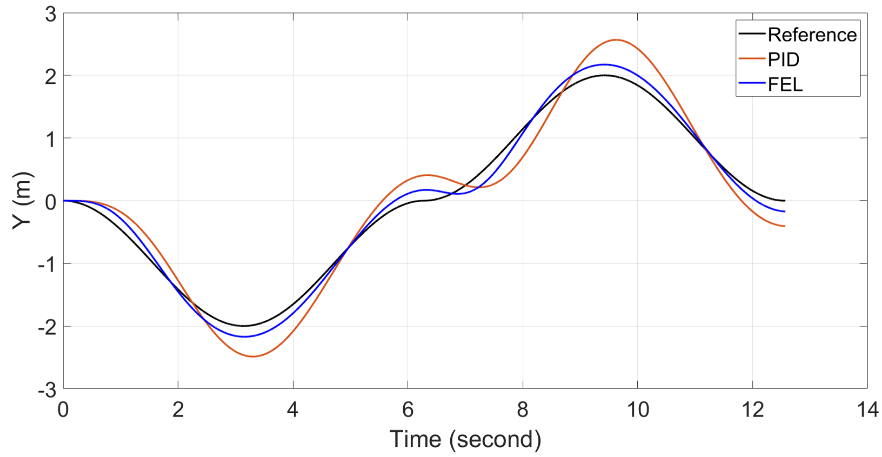

Figure 6.23 : Trajectory tracking performance on Y axis of the different controllers.

The FEL controller is composed of conventional controller and the NFC. So, the FEL gives faster reaction then conventional controllers as seen in Figure (6.21, 6.23, 6.25).

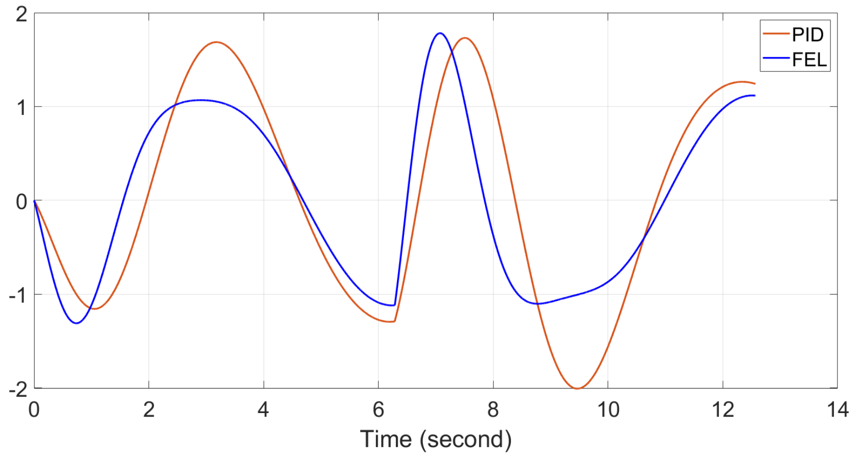

Figure 6.24 : PID and FEL control signals for Y axis.

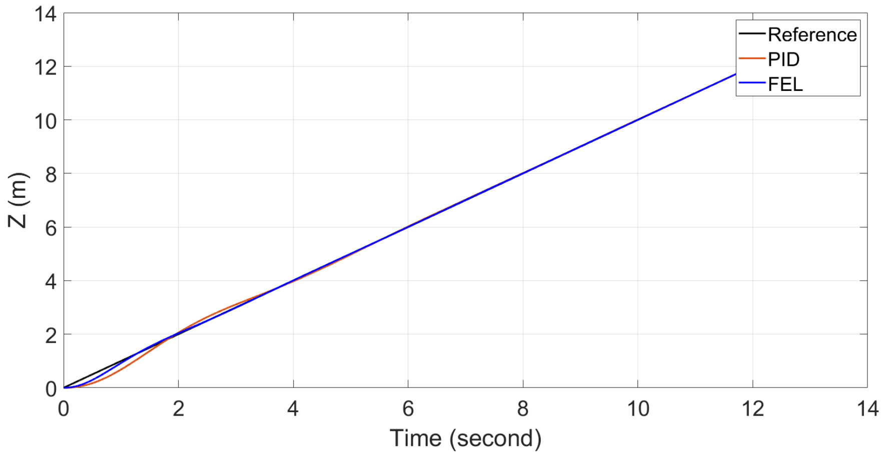

Figure 6.25 : Trajectory tracking performance on Z axis of the different controllers.

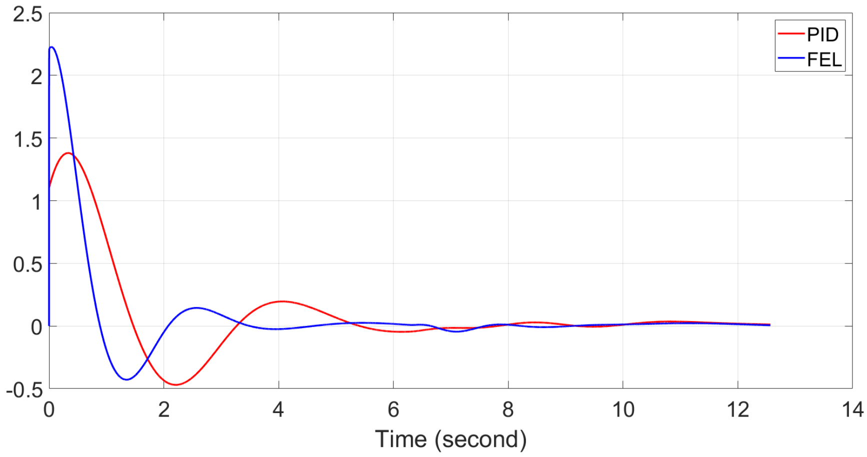

Figure 6.26 : PID and FEL control signals for Z axis.

As seen in Figures (6.22, 6.24, 6.26), the FEL control signals is more harsh answers

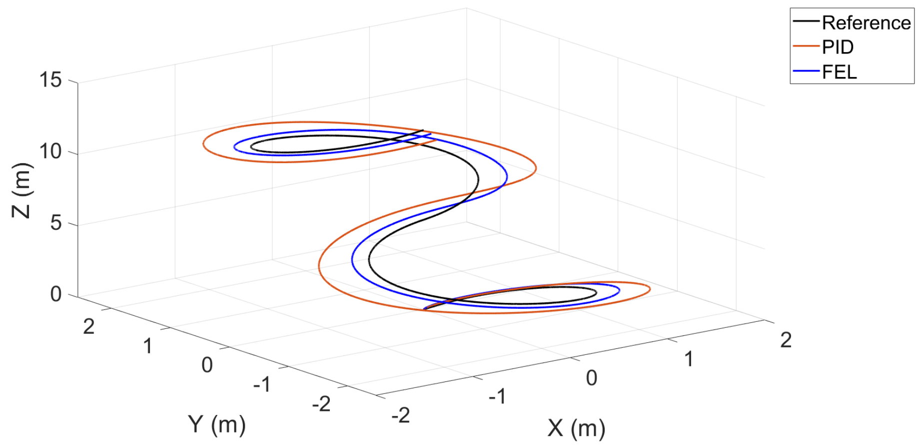

Figure 6.27 : Trajectory tracking performance of the different controllers.

than PID for errors. However, this much more powerful reactions does not cause to larger overshoots.

Table 6.5 : Comparison of PID and FEL controllers for Quadrotor position control.

Inputs	RMSE (x)	RMSE (y)	RMSE (z)	Computation Time (second)
PID	0.3389	0.3297	0.1034	0.9157
T1 FEL (Sugeno)	0.1507	0.1272	0.0470	3.6578
IT2 FEL (b=1)	0.1507	0.1272	0.0470	5.9418
IT2 FEL (b=0.5)	0.1507	0.1272	0.0470	6.6833

As seen in Table 6.5, FEL controller calculation load is more than PID. However, the FEL controller RMSE values are significantly smaller. The performance of the FEL can be seen from Figure 6.27. There is no uncertainty in the system, so the IT2 FLC and T1 FLC have the same results. Because of the inner iterations, the elapsed time of the IT2 FLC is much more than T1 FLC.

6.4.1 Wind Disturbance

The aircrafts have many disturbances while working like air resistance, wind disturbance, ground effect and ceiling effect if it is in a closed area [75]. The controllers' performance is examined under the wind disturbance effect. The wind disturbance effects are stated as ζ. The equations with wind disturbance are given below.

$$
\begin{bmatrix} \ddot{X}_E \\ \ddot{Y}_E \\ \ddot{Z}_E \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi)\frac{U_1}{m} + \zeta_1 \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi)\frac{U_1}{m} + \zeta_2 \\ -g + (c_\theta c_\phi)\frac{U_1}{m} + \zeta_3 \\ \frac{I_{yy}-I_{zz}}{I_{xx}} qr - \frac{J_r}{I_{xx}} q\Omega_r + \frac{U_2}{I_{xx}} + \zeta_4 \\ \frac{I_{zz}-I_{xx}}{I_{yy}} pr + \frac{J_r}{I_{yy}} p\Omega_r + \frac{U_3}{I_{yy}} + \zeta_5 \\ \frac{I_{xx}-I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}} + \zeta_6 \end{bmatrix} \tag{6.54}
$$

The wind disturbance on the aircraft is given as two different signals. Firstly, a fixed value is given for a second as shown in Figure 6.28.

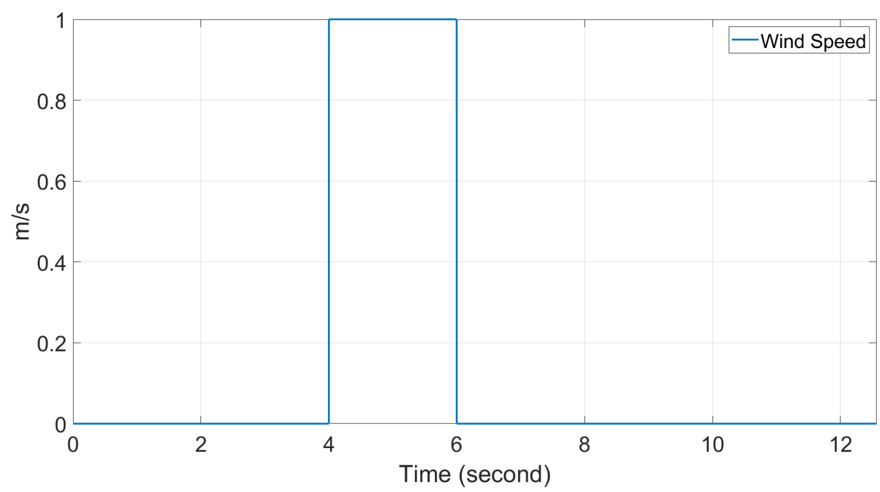

Figure 6.28 : Fixed wind speed.

Table 6.6 : Comparison of PID and FEL controllers for Quadrotor position control with fixed wind disturbance.

Inputs	RMSE (x)	RMSE (y)	RMSE (z)	Computation Time (second)
PID	0.4055	0.4202	0.1769	0.5139
T1 FEL (Sugeno)	0.1812	0.1499	0.0824	2.0105
IT2 FEL (b=1)	0.1812	0.1499	0.0824	6.7139
IT2 FEL (b=0.5)	0.1812	0.1499	0.0824	6.2762

The controller RMSE values for fixed wind disturbance are given in Table 6.6. As seen from Figure 6.29, the ANFIS based FEL controller is robust against to disturbances.

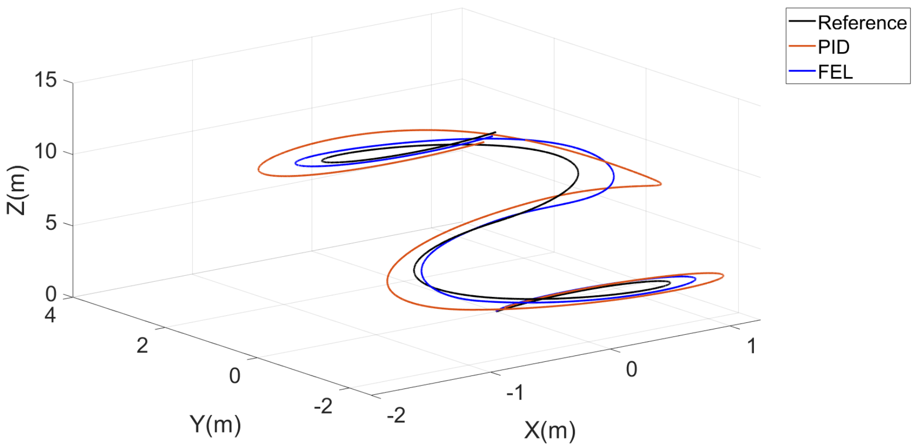

Figure 6.29 : Trajectory tracking performance of the different controllers for fixed wind noise.

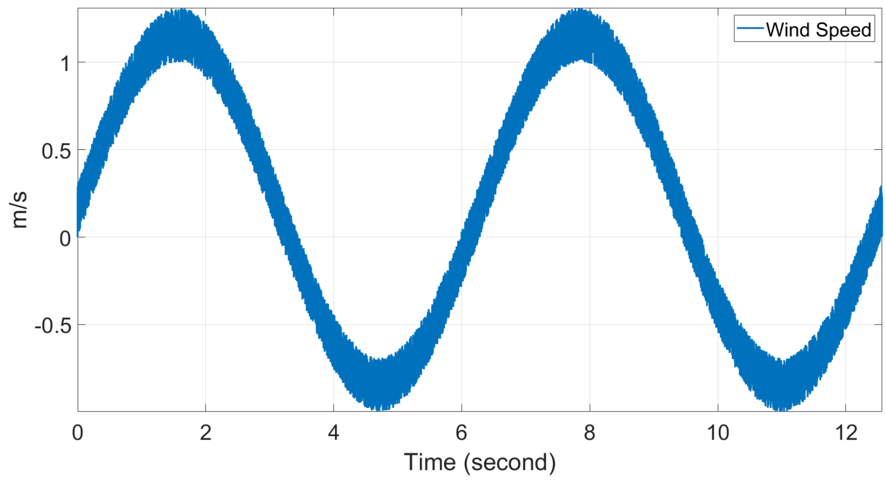

Figure 6.30 : Wind speed with sinusoidal white noise.

Table 6.7 : Comparison of PID and FEL controllers for Quadrotor position control with sinusoidal white noise.

Inputs	RMSE (x)	RMSE (y)	RMSE (z)	Computation Time (second)
PID	0.7980	0.4378	0.3707	0.5990
T1 FEL (Sugeno)	0.3292	0.1692	0.1477	1.9978
IT2 FEL (b=1)	0.3290	0.1689	0.1476	6.2987
IT2 FEL (b=0.5)	0.3290	0.1689	0.1475	7.0863

Secondly, a sinusoidal white noise is given as wind disturbance to the system as shown in Figure 6.30.

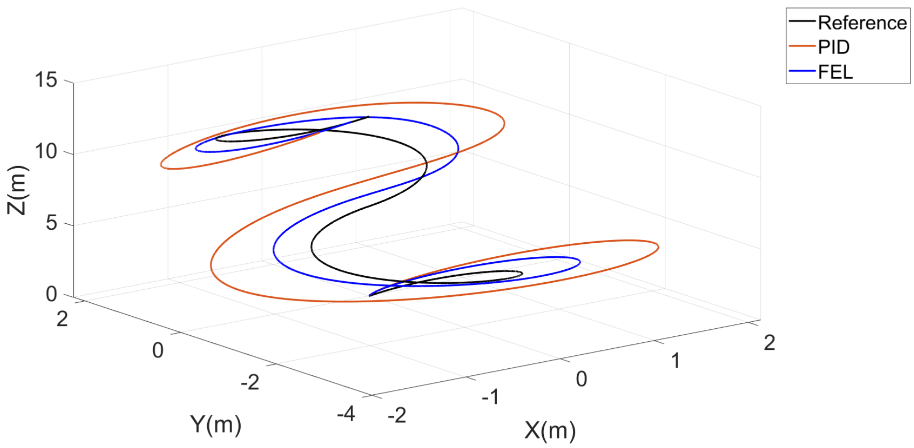

Figure 6.31 : Trajectory tracking performance of the different controllers for sinusoidal white noise.

The controller RMSE values for sinusoidal white noise disturbance are given in Table 6.7. The results show that the ANFIS based FEL controller is very effective to control any system under disturbances.

7. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, type-1 and interval type-2 fuzzy inference systems are examined and some modifications are implemented. New ANFIS models are proposed for Mamdani, Şen and interval type-2 FIS. The given ANFIS models are tested as controllers on SISO and MIMO systems.

Firstly, type-1 FISs are examined. The Mamdani ANFIS model is proposed and tested. The Şen FIS, which is found as SISO in the open literature, is designed as MIMO and the Şen ANFIS training model is proposed and tested. The proposed Mamdani and Şen ANFIS is compared with Sugeno ANFIS. It is seen that the LSE based Sugeno ANFIS has less RMSE value than others. However, the GD based Mamdani and ŞEN ANFIS has less RMSE value than Sugeno ANFIS, but the Mamdani ANFIS need to more computation time. In control studies, GD algorithm is used because FLS parameters must change smoothly for every single input vectors.

Secondly, the proposed type-1 ANFIS models are used in the NFCs. There are different NFC target functions in the literature. These target functions are tested and the best performing one selected. In addition, there are approaches with different controller inputs in the literature for the NFC structure. Five of them are compared to each other and one of them is chosen with respect to it's performance. The chosen NFC structure is used to compare Sugeno, Mamdani and Şen FISs. The Şen NFC has much less computational load than Mamdani NFC with same controller parameters. The Sugeno NFC has more overshoot than both of others but it has less computational load with less RMSE error. The Sugeno, Mamdani and Şen ANFIS controllers are compared for different NFC structures and Sugeno NFC is selected as the best. So, Sugeno ANFIS is compared to interval type-2 ANFIS studies in following studies.

Thirdly, the interval type-2 ANFIS are used in the NFC. As stated in the literature, the ANFIS uses exact parameter values, however, the IT2 FLS parameters include uncertainties. To eliminate the uncertainties in X matrix, we proposed the modified

Karnik-Mendel algorithm. The modified Karnik-Mendel algorithm is a different mathematical approach of Karnik-Mendel algorithm that is one of the type-reduction methods. The proposed IT2 ANFIS structure is tested over the function previously used for type-1 ANFIS. It is clear that the proposed IT2 ANFIS has less RMSE values than type-1 ANFIS for both LSE and GD method. At worst case, the IT2 ANFIS has the same RMSE results with T1 ANFIS. The development in this IT2 ANFIS has been a significant advance in solving problems involving uncertainties.

The solely NFC is very effective as controller but it is does not work regularly for a cascade structure like quadrotors. So, the FEL controller is tested on a quadrotor model. The results show that both of type-1 and interval type-2 based FEL controllers increased the performance of classical controllers. In IT2 ANFIS training, A2-C1 consequent parameters were trained. So, the type-1 and interval type-2 FEL controller results are similar. As future study, we continue on the IT2 ANFIS research on A2-C2 (Antecedent type-2 MF - Consequent type-2 MF) parameters training to decrease RMSE value and to increase controller performance. Because of that, the newly proposed direct approach type-reduction method is examined and being worked on to generate a new ANFIS model.

REFERENCES

[1] **Han, B.**, **Zhou, Y.**, **Deveerasetty, K.K. and Hu, C.** (2018). A review of control algorithms for quadrotor, *2018 IEEE International Conference on Information and Automation (ICIA)*, IEEE, pp.951–956.

[2] **Öztürk, M.** (2014). *Antropomorfik Robotların Dinamiği Ve Adaptif Kontrol Uygulamaları: Matlab/Simulink Modelleme, (Master Thesis)*, Istanbul Technical University.

[3] **Eberhart, R.C.** (1998). Overview of computational intelligence [and biomedical engineering applications], *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286)*, volume 3, IEEE, pp.1125–1129.

[4] **Zadeh, L.A.** (1973). Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on systems, Man, and Cybernetics*, *SMC-3*(1), 28–44.

[5] **Mamdani, E.H. and Assilian, S.** (1975). An experiment in linguistic synthesis with a fuzzy logic controller, *International journal of man-machine studies*, *7*(1), 1–13.

[6] **Sugeno, M.** (1985). *Industrial Applications of Fuzzy Control*, Elsevier Science Inc., New York, NY, USA.

[7] **Jang, J.S.** (1993). ANFIS: adaptive-network-based fuzzy inference system, *IEEE transactions on systems, man, and cybernetics*, *23*(3), 665–685.

[8] **Şahin, H.**, **Oktay, T. and Konar, M.** (2020). Anfis Based Thrust Estimation of a Small Rotary Wing Drone, *European Journal of Science and Technology*, (18), 738–742.

[9] **Hines, J.W.** (1997). *Fuzzy and Neural Approaches in Engineering Matlab Supplement*, John Wiley & Sons, Inc.

[10] **Raja, P. and Pahat, B.** (2016). A review of training methods of ANFIS for applications in business and economics, *International Journal of u-and e-Service, Science and Technology*, *9*(7), 165–172.

[11] **Neshat, M.**, **Adeli, A.**, **Masoumi, A. and Sargolzae, M.** (2011). A comparative study on ANFIS and fuzzy expert system models for concrete mix design, *International Journal of Computer Science Issues (IJCSI)*, *8*(3), 196.

[12] **Chai, Y.**, **Jia, L. and Zhang, Z.** (2009). Mamdani model based adaptive neural fuzzy inference system and its application, *International Journal of Computational Intelligence*, *5*(1), 22–29.

[13] **Öztürk, M.**, **Bayezit, İ. and Ozkol, İ.** (2021). Mamdani Model based ANFIS and Application in Prediction of Soot Emission, *European Journal of Science and Technology*, (23), 761–768.

[14] **Şen, Z.** (1998). Fuzzy algorithm for estimation of solar irradiation from sunshine duration, *Solar Energy*, *63*(1), 39–49.

[15] **Kayacan, E. and Maslim, R.** (2016). Type-2 fuzzy logic trajectory tracking control of quadrotor VTOL aircraft with elliptic membership functions, *IEEE/ASME Transactions on Mechatronics*, *22*(1), 339–348.

[16] **Jha, S.**, **Gaur, P. and Yadav, A.K.** (2016). Various intelligent control techniques for attitude control of an aircraft system, *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, pp.2476–2480.

[17] **Shmelova, T.**, **Shmelov, Y. and Vladov, S.** (2020). Concept of Building Intelligent Control Systems for Aircraft, Unmanned Aerial Vehicles and Aircraft Engines, *2020 IEEE 6th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*, IEEE, pp.14–19.

[18] **Karr, C.L. and Gentry, E.J.** (1993). Fuzzy control of pH using genetic algorithms, *IEEE Transactions on Fuzzy Systems*, *1*(1), 46.

[19] **Qin, S.J. and Borders, G.** (1994). A multiregion fuzzy logic controller for nonlinear process control, *IEEE Transactions on Fuzzy Systems*, *2*(1), 74–81.

[20] **D'Angelo, G.**, **Presti, M.L. and Rizzotto, G.** (1994). Fuzzy controller design to drive an induction motor, *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, IEEE, pp.1502–1507.

[21] **Mauer, G.F.** (1995). A fuzzy logic controller for an ABS braking system, *IEEE Transactions on Fuzzy Systems*, *3*(4), 381–388.

[22] **Chang, W.**, **Moon, J.H. and Lee, H.J.** (2017). Fuzzy model-based output-tracking control for 2 degree-of-freedom helicopter, *Journal of Electrical Engineering and Technology*, *12*(4), 1649–1656.

[23] **Li, W.** (1998). Design of a hybrid fuzzy logic proportional plus conventional integral-derivative controller, *IEEE transactions on fuzzy systems*, *6*(4), 449–463.

[24] **Ndzabukelwayo, S.M.**, **Mpofu, K. and Tlale, N.** (2012). An intelligent fuzzy PID controller for a reconfigurable machine tool, *2012 International Conference on Information Security and Intelligent Control*, IEEE, pp.99–102.

[25] **Xu, J.X.**, **Pok, Y.M.**, **Liu, C. and Hang, C.C.** (1998). Tuning and analysis of a fuzzy PI controller based on gain and phase margins, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *28*(5), 685–691.

[26] **Mann, G.K.**, **Hu, B.G. and Gosine, R.G.** (1999). Analysis of direct action fuzzy PID controller structures, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *29*(3), 371–388.

[27] **Zhao, Z.Y.**, **Tomizuka, M. and Isaka, S.** (1993). Fuzzy gain scheduling of PID controllers, *IEEE transactions on systems, man, and cybernetics*, *23*(5), 1392–1398.

[28] **Uddin, M.N. and Wen, H.** (2007). Development of a self-tuned neuro-fuzzy controller for induction motor drives, *IEEE Transactions on industry applications*, *43*(4), 1108–1116.

[29] **Aras, A.C.**, **Kayacan, E.**, **Oniz, Y.**, **Kaynak, O. and Abiyev, R.** (2010). An adaptive neuro-fuzzy architecture for intelligent control of a servo system and its experimental evaluation, *2010 IEEE International Symposium on Industrial Electronics*, IEEE, pp.68–73.

[30] **Aras, A.C. and Kaynak, O.** (2014). Trajectory tracking of a 2-DOF helicopter system using neuro-fuzzy system with parameterized conjunctors, *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, pp.322–326.

[31] **Öztürk, M. and Özkol, İ.** (2021). Comparison of self-tuned Neuro-Fuzzy controllers on 2 DOF helicopter: an application, *SN Applied Sciences*, *3*(1), 1–14.

[32] **Bresciani, T.** (2008). Modelling, identification and control of a quadrotor helicopter, *MSc Theses*.

[33] **Kawato, M.**, (1990). Feedback-error-learning neural network for supervised motor learning, Advanced neural computers, Elsevier, pp.365–372.

[34] **Miyamura, A. and Kimura, H.** (2002). Stability of feedback error learning scheme, *Systems & control letters*, *45*(4), 303–316.

[35] **Nakanishi, J. and Schaal, S.** (2004). Feedback error learning and nonlinear adaptive control, *Neural Networks*, *17*(10), 1453–1465.

[36] **Gomi, H. and Kawato, M.** (1993). Neural network control for a closed-loop system using feedback-error-learning, *Neural Networks*, *6*(7), 933–946.

[37] **Ruan, X.**, **Ding, M.**, **Gong, D. and Qiao, J.** (2007). On-line adaptive control for inverted pendulum balancing based on feedback-error-learning, *Neurocomputing*, *70*(4-6), 770–776.

[38] **Khanesar, M.A.**, **Kayacan, E.**, **Reyhanoglu, M. and Kaynak, O.** (2015). Feedback error learning control of magnetic satellites using type-2 fuzzy neural networks with elliptic membership functions, *IEEE transactions on cybernetics*, *45*(4), 858–868.

[39] **Hagras, H.A.** (2004). A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, *IEEE Transactions on Fuzzy systems*, *12*(4), 524–539.

[40] **Zadeh, L.A.** (1975). The concept of a linguistic variable and its application to approximate reasoning—I, *Information sciences*, *8*(3), 199–249.

[41] **Taşkın, A.** (2015). *A Matlab/simulink Toolbox For Interval Type-2 Fuzzy Logic Systems, (Ph.d. Thesis)*, Istanbul Technical University.

[42] **Liang, Q. and Mendel, J.M.** (2000). Interval type-2 fuzzy logic systems: theory and design, *IEEE Transactions on Fuzzy systems*, *8*(5), 535–550.

[43] **Karnik, N.N.**, **Mendel, J.M. and Liang, Q.** (1999). Type-2 fuzzy logic systems, *IEEE transactions on Fuzzy Systems*, *7*(6), 643–658.

[44] **Wu, D. and Mendel, J.M.** (2008). Enhanced karnik–mendel algorithms, *IEEE Transactions on Fuzzy Systems*, *17*(4), 923–934.

[45] **Duran, K.**, **Bernal, H. and Melgarejo, M.** (2008). Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set, *NAFIPS 2008-2008 Annual Meeting of the North American Fuzzy Information Processing Society*, IEEE, pp.1–5.

[46] **Wu, D. and Nie, M.** (2011). Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems, *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, IEEE, pp.2131–2138.

[47] **Chen, C.**, **John, R.**, **Twycross, J. and Garibaldi, J.M.** (2017). A direct approach for determining the switch points in the Karnik–Mendel algorithm, *IEEE Transactions on Fuzzy Systems*, *26*(2), 1079–1085.

[48] **Wu, D.** (2012). On the fundamental differences between interval type-2 and type-1 fuzzy logic controllers, *IEEE Transactions on Fuzzy Systems*, *20*(5), 832–848.

[49] **Sakalli, A.**, **Beke, A. and Kumbasar, T.** (2018). Analyzing the control surfaces of type-1 and interval type-2 FLCs through an experimental study, *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp.1–6.

[50] **Kumbasar, T.**, **Eksin, I.**, **Guzelkaya, M. and Yesil, E.** (2011). Interval type-2 fuzzy inverse controller design in nonlinear IMC structure, *Engineering Applications of Artificial Intelligence*, *24*(6), 996–1005.

[51] **Kumbasar, T.**, **Eksin, I.**, **Guzelkaya, M. and Yesil, E.** (2017). An inverse controller design method for interval type-2 fuzzy models, *Soft Computing*, *21*(10), 2665–2686.

[52] **Zeghlache, S.**, **Kara, K. and Saigaa, D.** (2014). Type-2 fuzzy logic control of a 2-DOF helicopter (TRMS system), *Open Engineering*, *4*(3), 303–315.

[53] **Kumbasar, T.** (2014). A simple design method for interval type-2 fuzzy PID controllers, *Soft Computing*, *18*(7), 1293–1304.

[54] **Kumbasar, T.**, **Eksin, I.**, **Guzelkaya, M. and Yesil, E.** (2012). Type-2 fuzzy model based controller design for neutralization processes, *ISA transactions*, *51*(2), 277–287.

[55] **Kumbasar, T. and Hagras, H.** (2014). Big Bang–Big Crunch optimization based interval type-2 fuzzy PID cascade controller design strategy, *Information Sciences*, *282*, 277–295.

[56] **Kumbasar, T. and Hagras, H.** (2015). A gradient descent based online tuning mechanism for pi type single input interval type-2 fuzzy logic controllers, *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp.1–6.

[57] **Khanesar, M.A. and Kayacan, E.**, (2015). Controlling the pitch and yaw angles of a 2-dof helicopter using interval type-2 fuzzy neural networks, Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics, Springer, pp.349–370.

[58] **Khanesar, M.A.**, **Kayacan, E. and Kaynak, O.** (2015). Optimal sliding mode type-2 tsk fuzzy control of a 2-dof helicopter, *2015 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, IEEE, pp.1–6.

[59] **Kayacan, E. and Khanesar, M.A.** (2016). Recurrent interval type-2 fuzzy control of 2-DOF helicopter with finite time training algorithm, *IFAC-PapersOnLine*, *49*(13), 293–299.

[60] **Chen, C.**, **John, R.**, **Twycross, J. and Garibaldi, J.M.** (2016). An extended ANFIS architecture and its learning properties for type-1 and interval type-2 models, *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp.602–609.

[61] **Chen, C.**, **John, R.**, **Twycross, J. and Garibaldi, J.M.** (2017). Type-1 and interval type-2 ANFIS: A comparison, *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp.1–6.

[62] **Öztürk, M. and Özkol, İ.** (2021). Comparison of modified Karnik-Mendel algorithm-based interval type-2 ANFIS and type-1 ANFIS, *Aircraft Engineering and Aerospace Technology*, *93*(10), 1526–1532.

[63] **Cuevas, E.**, **Díaz, P.**, **Avalos, O.**, **Zaldivar, D. and Pérez-Cisneros, M.** (2018). Nonlinear system identification based on ANFIS-Hammerstein model using Gravitational search algorithm, *Applied Intelligence*, *48*(1), 182–203.

[64] **Şen, Z.** (2009). *Bulanık mantık ilkeleri ve modelleme:(Mühendislik ve sosyal bilimler)*, Su Vakfı.

[65] **Zadeh, L.A.** (1965). Fuzzy sets, *Information and control*, *8*(3), 338–353.

[66] **Kumbasar, T.** (2012). *Type-2 Fuzzy Model Inversion Methods and Fuzzy Model Based Controller Design, (Ph.d. Thesis)*, Istanbul Technical University.

[67] **De los Angeles Hernandez, M.**, **Melin, P.**, **Méndez, G.M.**, **Castillo, O. and López-Juarez, I.** (2015). A hybrid learning method composed by the orthogonal least-squares and the back-propagation learning algorithms for interval A2-C1 type-1 non-singleton type-2 TSK fuzzy logic systems, *Soft Computing*, *19*(3), 661–678.

[68] **Mendel, J.M.**, (2017). Uncertain rule-based fuzzy systems, Introduction and new directions, Springer, p.684.

[69] **Antão, R.J.M.M.** (2016). *Type-2 Fuzzy Logic: Uncertain Systems' Modeling and Control*, Universidade de Aveiro (Portugal).

[70] **Karnik, N.N. and Mendel, J.M.** (2001). Centroid of a type-2 fuzzy set, *Information Sciences*, *132*(1-4), 195–220.

[71] **Kumbasar, T.**, **Eksin, I.**, **Guzelkaya, M. and Yesil, E.** (2011). Type-2 fuzzy model inverse controller design based on BB-BC optimization method, *IFAC Proceedings Volumes*, *44*(1), 5308–5313.

[72] **Luukkonen, T.** (2011). Modelling and control of quadcopter, *Independent research project in applied mathematics, Espoo*, *22*, 22.

[73] **Stroud, K.A. and Booth, D.J.** (2003). *Advanced engineering mathematics*, Palgrave.

[74] **Dikmen, İ.C.**, **Arisoy, A. and Temeltas, H.** (2009). Attitude control of a quadrotor, *2009 4th International Conference on Recent Advances in Space Technologies*, IEEE, pp.722–727.

[75] **Sun, C.**, **Liu, M.**, **Liu, C.**, **Feng, X. and Wu, H.** (2021). An Industrial Quadrotor UAV Control Method Based on Fuzzy Adaptive Linear Active Disturbance Rejection Control, *Electronics*, *10*(4), 376.

CURRICULUM VITAE

Name Surname: Muhammet ÖZTÜRK

EDUCATION:

- **B.Sc.:** 2011, Istanbul Technical University, Faculty of Aeronautic and Astronautic, Astronautical Engineering
- **M.Sc.:** 2014, Istanbul Technical University, Faculty of Aeronautic and Astronautic, Aeronautical and Astronautical Engineering

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Öztürk, M.** and Özkol, İ., 2021. Comparison of modified Karnik-Mendel algorithm-based interval type-2 ANFIS and type-1 ANFIS, *Aircraft Engineering and Aerospace Technology, 93(10)*, 1526–1532.
- **Öztürk, M.** and Özkol, İ., 2021. Comparison of self-tuned Neuro-Fuzzy controllers on 2 DOF helicopter: an application, *SN Applied Sciences, 3(1)*, 1-14.
- **Öztürk, M.** and Özkol, İ., 2021. Mamdani Model based ANFIS and Application in Prediction of Soot Emission, *European Journal of Scence and Technology, (23)*, 761-768.
- **Öztürk, M.,** Yılmaz, M. U., Ozgur, E. and Akataş, N., 2017. Using fuzzy logic approach on evaporation modeling. *International Journal of Natural and Engineering Sciences, 11(3)*, 04-06.
- **Öztürk, M.** and Özkol, İ., 2020. Comparison of Type-1 and Type-2 Fuzzy Inference Systems at Aircraft Pitch Control, *Euro Asia 7th Internaitonal Congress on Applied Sciences*, August 21-22, Trabzon.
- **Öztürk, M.** and Bayezit, İ., 2017. Neuro-Fuzzy Approach to Adapt Vehicle Calibration Maps, *International Conference on Mathematics and Mathematics Education*, May 11-13, Harran University, Şanlıurfa.
- **Öztürk, M.**, Bayezit, İ., Okudan, M.E. and Özkol, İ., 2017. Sugeno and Mamdani Based Fuzzy Inference Approaches for Aircraft Pitch Control, *International Conference on Mathematics and Mathematics Education*, May 11-13, Harran University, Şanlıurfa.
- **Öztürk, M.**, Bitirgen, R., Bayezit, İ. and Özkol, İ., 2016. Fuzzy Inference System Design for Linear Quadratic Regulator based 2-DOF Helicopter Control, *Turkey Otomatic Control Conference*, 29 September-1 October, Eskişehir Osmangazi University, Eskişehir.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Kalkan, A., Göçmen, F.E. and **Öztürk, M.**, 2020. Front and Rear Spar Design on Aircraft Wings, *Euro Asia 7th Internaitonal Congress on Applied Sciences*, August 21-22, Trabzon.

- Baylan, K., **Öztürk, M.** and Bayezit, İ., 2015. 2 Serbestlik Dereceli Helikopterin Modelleme Ve Kontrolü, *III. Ulusal Havacılık Teknolojisi ve Uygulamaları Kongresi*, October 23-24, Ege University, İzmir.

- **Öztürk, M.** and Caferov, E., 2014. Antropomorfik Robotların Dinamiği ve Uyarlamalı Kontrol Uygulamaları, *Turkey otomatic Control Conference*, September 23-24, Kocaeli University, Kocaeli.