ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

A MATLAB/SIMULINK TOOLBOX FOR INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS

M.Sc. THESIS

Ahmet TAŞKIN

Department of Control and Automation Engineering

Control and Automation Engineering Programme

MAY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

A MATLAB/SIMULINK TOOLBOX FOR INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS

M.Sc. THESIS

Ahmet TAŞKIN
(504121101)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Asst. Prof. Dr. Tufan KUMBASAR

MAY 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ARALIK DEĞERLİ TİP-2 BULANIK MANTIK SİSTEMLER İÇİN BİR MATLAB/SIMULINK ARAÇ KUTUSU

YÜKSEK LİSANS TEZİ

Ahmet TAŞKIN
(504121101)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Tufan KUMBASAR

MAYIS 2015

Ahmet Taşkın, a **M.Sc.** student of ITU **Graduate School of Science Engineering And Technology** student ID **504121101**, successfully defended the **thesis/dissertation** entitled **"A MATLAB/SIMULINK TOOLBOX FOR INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS",** which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Asst. Prof. Dr. Tufan KUMBASAR**
Istanbul Technical University

Jury Members : **Prof. Dr. İbrahim EKSİN**
Istanbul Technical University

Asst. Prof. Dr. Gürkan SOYKAN
Bahçeşehir University

Date of Submission : 04 May 2015
Date of Defense : 28 May 2015

To my family and dear friends,

FOREWORD

I would like to show my gratitude to my thesis advisor Asst. Prof. Dr. Tufan Kumbasar for his encouragements, sincerity and guidance during the period of my university education. I would like to thank Asst. Prof. Dr. Engin Yeşil for his help, suggestions and vision during the period of my thesis and laboratory studies. I would also like to thank Hazal Kurtuluş for her presence and consistent support. Finally, I would like to thank my parents and my friends for their assistance and confidence.

May 2015

Ahmet TAŞKIN
(Software Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

BMM	: Begain-Melek-Mendel
EIASC	: Enhanced Iterative Algorithm with Stop Condition
EKM	: Enhanced Karnik-Mendel
EODS	: Enhanced Opposite Direction Searching
FLC	: Fuzzy Logic Controller
FLS	: Fuzzy Logic System
FOPDT	: First Order plus Dead Time
FOU	: Footprint of Uncertainty
FPID	: Fuzzy PID
GUI	: Graphical User Interface
IASC	: Iterative Algorithm with Stop Condition
IT2-FLC	: Interval Type-2 Fuzzy Logic Controller
IT2-FLS	: Interval Type-2 Fuzzy Logic System
IT2-FPID	: Interval Type-2 Fuzzy PID
IT2-FS	: Interval Type-2 Fuzzy Set
KM	: Karnik-Mendel
LMF	: Lower Membership Function
MF	: Membership Function
NT	: Nie-Tan
ODS	: Opposite Direction Searching
PD	: Proportional-Derivative
PI	: Proportional-Integral
PID	: Proportional-Integral-Derivative
T1-FLS	: Type-1 Fuzzy Logic System
T1-FPID	: Type-1 Fuzzy PID
T1-FS	: Type-1 Fuzzy Set
T2-FLS	: Type-2 Fuzzy Logic System
T2-FS	: Type-2 Fuzzy Set
TR	: Type Reduction
UI	: User Interface
UMF	: Upper Membership Function
WM	: Wu-Mendel

LIST OF TABLES

Page

LIST OF FIGURES

Page

A MATLAB/SIMULINK TOOLBOX FOR TYPE-2 FUZZY LOGIC SYSTEMS

SUMMARY

The fuzzy logic has obtained attention of the researchers for last couple of decades. It has opened new scopes in both the academia and the industry site. Fuzzy logic was first introduced in 1965 by Prof. Lotfi A. Zadeh. Fuzzy logic is flexible application of classical logic rules and fuzzy sets are the extension of the classical logic set notation. Fuzzy logic defines the interval between the crisp values while classical logic is working with crisp values such as true or false. In addition, it is possible to define linguistic variables such as short, very short, tall, or very tall with fuzzy logic.
Fuzzy logic sets proposed by Prof. Lotfi A. Zadeh in 1965 are knows as type-1 (ordinary) fuzzy sets. The systems that include at least one type-1 fuzzy set are called as type-1 fuzzy logic systems. These type of systems are used in many areas like robotics, modeling and control of nonlinear systems or image processing. Lately, it has been demonstrated that type-1 fuzzy sets might be inadequate to cover uncertainties and nonlinearities due to defining their members as a crisp number in an interval [0,1].
Type-2 fuzzy logic sets are also introduced by Prof. Lotfi A. Zadeh in 1975. The type-2 fuzzy sets are extension of the type-1 fuzzy sets. The systems that include at least one type-2 fuzzy sets are called as type-2 fuzzy logic systems. The studies have shown that, type-2 fuzzy logic systems are more successful than type-1 fuzzy logic systems to describe uncertainties and nonlinear behaviors. However, working with type-2 fuzzy logic systems are much more complicated than working with type-1 fuzzy logic systems. There are many additional computational costs while working with type-2 fuzzy logic systems. Therefore, interval type-2 fuzzy sets are proposed. Interval type-2 fuzzy sets are the special case of the type-2 fuzzy sets. In literature, there are many successful studies in interval type-2 fuzzy logic sets and systems.
Interval type-2 fuzzy systems consists of five components: fuzzifier, inference, rules, type reducer and defuzzifier. The only different component between type-1 fuzzy logic systems and type-2 fuzzy logic systems is the type reducer. Interval type-2 fuzzy logic systems need to the type reducer block to convert type-2 fuzzy sets to type-1 fuzzy sets before defuzzifier. However, type reducer component brings some computation costs to the type-2 fuzzy logic systems. There are many type reduction method proposed in literature. The most widely used common type reduction method is the iterative Karnik and Mendel algorithm. The Karnik-Mendel algorithm aims to reduce the type-2 sets to type-1 sets by finding the optimal switching points iteratively. In literature, there are some studies that aim to enhance the Karnik Mendel Algorithm to improve the performance such as enhanced Karnik-Mendel algorithm. In addition, there some closed form type-reduction methods that find the solution without needing any iterative.
In this thesis, firstly, type-1 fuzzy sets and systems are explained. Then, type-2 fuzzy sets and systems are introduced and interval type-2 fuzzy sets and systems are explained more in detail. In addition, differences between type-1 fuzzy sets and systems and interval type-2 fuzzy sets and systems are explained. The components of an interval type-2 fuzzy logic system are mentioned and the most known type reduction methods are explained. Interval type-2 fuzzy logic systems are quite complicated systems and it needs many steps to implement them from the first phase to implementation phase. Therefore, in this thesis, a Matlab/Simulink toolbox is

developed and proposed to cover all phases of an interval type-2 fuzzy logic system design from the first description phase to the final implementation phase through type reduction. The proposed interval type-2 fuzzy logic toolbox allows users to design an interval type-2 fuzzy logic system by using the user interfaces. It makes the design of the interval type-2 fuzzy logic system so easy and understandable.

The design of the interval type-2 fuzzy logic system toolbox starts by creating a structure in the Matlab workspace to save the information of the interval type-2 fuzzy logic system. Type reduction method, input and output variable types, the membership function parameters, the rules and the other information are saved to this structure. The interval type-2 fuzzy logic systems toolbox consists of four main pages: main editor, membership functions editor, rule editor and surface viewer. The input and output variables number of the interval type-2 fuzzy logic design is determined in main editor page. In addition, it is possible to choose the desired type reduction method from the main editor. The all type reduction methods that are explained in this thesis are implemented as a Matlab function and embedded to the toolbox via a pop-up menu in main editor. The user can select his desired type reduction method by using this menu easily. The membership function editor page allows to define the upper and lower membership functions of the each input and output variables. To provide this, the all membership functions of the Matlab fuzzy logic toolbox have been reused by adding an additional parameter into the end. The last parameter provides the opportunity to define the height of the lower membership functions. In addition, it is possible to define the membership functions of the output variables in either crisp or interval. Defining rules for the interval type-2 fuzzy logic system design is possible in the rule editor page. In addition, it is possible to view the surface of the current design from the surface viewer page. The interval type-2 fuzzy logic systems toolbox is designed to be working with Matlab/Simulink. To provide this feature, firstly a new Simulink library has been created for the interval type-2 fuzzy logic toolbox. The created Simulink library has two blocks. The first one in used to simulate the designed interval type-2 fuzzy logic controller, and the second one gives the user an opportunity to choose desired type reduction method. Also, it is possible to export the interval type-2 fuzzy logic system design from the toolbox to Simulink automatically by clicking a button. Then, it creates a Simulink model with current design and the user can start the simulations easily.

In the last section of this thesis, an interval type-2 fuzzy logic system is created by using the proposed toolbox and some performance analysis are done. Firstly, an interval type-2 fuzzy logic controller is designed by using the toolbox. Then, the control surfaces of the different type reduction methods are given. The interval type-2 fuzzy logic controller exported to the Simulink and closed loop control system is created with a first order plus dead time system. Then, the performances of the same interval type-2 fuzzy logic controller with different type reduction methods are compared for the nominal system parameters. After that, the system parameters are perturbed several times and the simulations are repeated to compare the performances of the different type reduction methods In addition, the computational time of the each type reduction methods are measured and compared.

In summary, a Matlab/Simulink toolbox for interval type-2 fuzzy logic systems is designed and proposed in this thesis. The proposed interval type-2 fuzzy logic toolbox covers the all phases of an interval type-2 fuzzy logic system design from the initial description phase to the final implementation phase including type reduction component. Then, an interval type-2 fuzzy logic controller is designed by using the toolbox and the simulations are done to compare the control performance of the different type reduction methods for the same system. In addition, the computational times of the each type reduction methods are measured and compared.

ARALIK DEĞERLİ TİP-2 BULANIK SİSTEMLER İÇİN BİR MATLAB/SIMULINK ARAÇ KUTUSU

ÖZET

Son yıllarda, bulanık mantık konusu bir çok araştırmacının dikkatini çekmektedir. Literatürde ve sanayine, bulanık mantık ile birçok yeni ufuklar açılmıştır. Bulanık mantık ilk olarak 1965 yılında Prof. Lotfi. A. Zadeh tarafından önerilmiştir. Bulanık mantık, klasik mantık yapısının daha esnek halidir ve bulanık mantık kümeler, klasik mantık kümelerin genişlemiş halidir. Klasik mantık doğru ve yanlış gibi keskin ifadelerle işlemler yaparken, bulanık mantık, bu keskin ifadelerin aralarında kalan değerlerle de çalışır. Bunlara ek olarak, bulanık mantık ile, kısa, daha kısa, uzun, daha uzun gibi dilsel ifadeleri de tanımlamak mümkün olmuştur.

Prof. Lotfi. A. Zadeh tarafından 1965 yılında önerilmiş olan bulanık mantık kümeler, klasik (tip-1) bulanık mantık kümeler olarak bilinir. En az bir adet tip-1 bulanık küme içeren sistemler de tip-1 bulanık sistemler olarak bilirnirler. Tip-1 bulanık sistemlerin robotikte, modelleme ve kontrolde, karar verme konularında, görüntü işlemede ve daha bir çok konuda bir çok başarışı uygulaması vardır. Ancak, son yıllarda yapılan çalışmalar, tip-1 bulanık kümelerinde bazı belirsizlikleri ve lineer olmayan davranışları ifade etmede yetersiz kalabileceğini göstermiştir. Bunun sebebi de, tip-1 bulanık kümelerde, üyelik fonksiyonunlarının aitliklerinin keskin sayılarla ifade ediliyor olmasıdır.

Tüm bunların ışığında, Prof. Lotfi A. Zadeh 1975 yılında tip-2 bulanık kümeler ile ilgili ilk çalışmayı yayınlamıştır. Aslında tip-2 bulanık kümeler, tip-1 bulanık kümelerin genişletilmiş halidir. Çalışmalar göstermiştirki, tip-2 bulanık kümeler, belirsizlikleri ve lineer olmayan davranışları ifade etmede, tip-1 bulanık kümelere göre oldukça başarılıdırlar. Bunun yanında, tip-2 kümeler ile çalışmak, tip-1 kümeler ile çalışmaktan çok daha karmaşıktır. Tip-2 bulanık kümeler ile çalışmak bir çok işlem yükünü de beraberinde getirmektedir. Bu sebeple, nispeten işlem yükü daha düşük olan aralık değerli tip-2 bulanık kümeler önerilmiştir. Aralık değerli tip-2 bulanık kümeleri, tip-2 bulanık kümelerin özel bir halidir. Aralık değerli tip-2 bulanık kümelerde, giriş değişkenlerinin aitlikleri aralıklar şeklinde ifade edilir. Bir başka değişle, aralık değerli tip-2 bulanık kümelerde herhangi bir giriş değişkenin aitliği keskin bir değer yerine, bir aralık olarak ifade edilir. Bu sayede, belirsizlikler ve lineer olmayan ifadeler aralik değerli tip-2 bulanık kümelerde daha iyi ifade edilirler. En az bir adet aralık değerli tip-2 küme içeren sistemlere, aralık değerli tip-2 sistemler denir. Literatürde, aralık değerli tip-2 bulanık kümeler ve sistemlerin bir çok başarılı uygulaması bulunmaktadır.

Temel olarak aralık değerli tip-2 sistemler bulanıklaştırıcı, çıkarım mekanizması, kurallar, tip indirgeyici ve durulaştırıcı olmak üzere 5 adet komponentten oluşmaktadır. Aralık değerli tip-2 sistemlerde giriş işaretlerine ilk olarak bulanıklaştırıcı uygulanır. Daha sonra, o anki giriş işaretleri için tanımlanmış olan kurallar kullanılarak çıkarım mekanizması tarafından çıkarımlar yapılır. Aralık değerli tip-2 bulanık sistemlerde çıkarım mekanizmasının çıkışı tip-2 kümelerdir ve bu tip-2 kümelerin durulaştırıcı öncesinde tip-1 bulanık kümelere dönüştürülmesi gerekmektedir. Bu sebeple, çıkarım mekanizmasdan sonra elde edilen tip-2 kümeler, tip indirgeyici blok ile tip-1 kümelere dönüştürülür.Tip-1 bulanık mantık sistemler ile tip-2 bulanık mantık sistemler arasındaki en büyük fark bu tip indirgeme bloğudur. Tip indirgeme işlemi aralık değerli tip-2 sistemlerde hesaplama yükünü arttıran

komponentlerden biridir. Literatürde, tip indirgeme işlemi için bir çok yöntem önerilimiştir. Bu yöntemlerden en çok bilineni ve yaygın olarak kullanılanı Karnik-Mendel algoritmasıdır. Karnik-Mendel algoritması, optimum anahtarlama noktalarını tekrarlamalı olarak bulmayı hedefler. Ancak Karnik-Mendel algoritmasının çözümü tekrarlamalı olarak bulmasından dolayı aralık değerli tip-2 sisteme oldukça hesaplama yükü getirmektedir. Bu sebeple, literatürde, Karnik-Mendel algoritmasının iterasyon sayısını düşürerek, performansını artırarak optimum anahtarlama noktalarını bulmayı hedefleyen bir çok çalışma vardır. Ayrıca, literatürde, tip indirgeme işlemini tekrarlamalı olmadan kapalı formda gerçekleştiren bir çok çalışma da bulunmaktadır.

Bu tez çalışmasında, öncelikle tip-1 bulanık kümeler ve tip-1 bulanık sistemler anlatılmıştır. Daha sonra, tip-2 kümelerden bahsedilerek, aralık değerli tip-2 kümeler ve sistemler detaylıca incelenmiştir. Ayrıca, tip-1 bulanık kümeler ile aralık değerli tip-2 kümelerin ve sistemlerin temel farklılıklarından da bahsedilmiştir. Aralık değerli tip-2 sistemlerin temel bileşenlerinden bahsedilerek, yaygın olarak bilinen ve kullanılan bir çok tip indirgeme yöntemi açıklanmıştır. Aralık değerli tip-2 kümeler ve sistemler, anlaşılması ve uygulanması oldukça zor ve karmaşık sistemlerden. Buradan yola çıkarak, bu tez çalışmasında, aralık değerli tip-2 bulanık bir sistemin tasarımından uygulanmasına kadarki bütün süreçleri kapsayan ve uygulanabilmesine imkan sağlayan bir Matlab/Simulink araç kutusu geliştirilmiş ve önerilmiştir. Bu geliştirilen aralık değerli tip-2 sistemler Matlab/Simulink araç kutusu ile, bir aralık değerli tip-2 bulanık mantık sistemi tasarlamak ve benzetim aşamasına geçmek oldukça kolay ve anlaşılır bir şekilde yapılabilmektedir. Bütün tasarım aşamaları, ilk adımdan benzetim adımına kadar araç kutusunun arayüzleri sayesinde yapılabilmektedir. Ayrıca, tez içerisinde anlatılmış olan tüm tip indirgeme yöntemleri de araç kutusuna eklenmiş olup, tasarımcı sadece arayüzü kullanarak istediği tip indirgeme yöntemini seçebilmektedir. Ayrıca, geliştirilmiş olan araç kutusu Simulink ile de çalışabilecek şekilde tasarlanmıştır. Kullanıcı arayüzü kullanarak tasarlamış olduğu aralık değerli tip-2 bulanık sistemi kolayca Simulink ortamına aktarabilmektedir.

Geliştirilmiş olan aralık değerli tip-2 bulanık sistemler Matlab/Simulink araç kutusunun tasarımı Matlab çalışma uzayında bir tip-2 çıkarım yapısı oluşturarak başlar. Tip indirgeme yöntemi, giriş ve çıkış değişkenlerine ait üyelik fonksiyonu tipleri, parametreleri, oluşturulan kurallar ve diğer tüm bilgiler bu yapıda saklanır. Araç kutusu ile aralık değerli tip-2 bulanık system tasarımı yapılırken, her adımda bu yapı güncellenerek tasarımın korunmasına ve daha sonra da kullanıbilmesine olanak sağlar. Bu aralık değerli tip-2 bulanık sistemler için tasarlanan araç kutusu temel olarak dört kullanıcı arayüzünden oluşmaktadır. Geliştirilmiş olan aralık değerli tip-2 bulanık sistemler araç kutusu ilk olarak açıldığında tasarıma ana ekran ile başlanır. Bu arayüzden, tasarlanacak olan aralık değerli tip-2 bulanık sistemin giriş ve çıkış değişkenleri sayıları belirlenebilir. Ayrıcı tip indirgeme için, bu tezde anlatılan tüm yöntemler birer Matlab fonksiyonu haline getirilip araç kutusunu entegre edilmiştir. Bu ana arayüzden istenilen tip indirgeme yöntemi kolayca seçilebilecek şekilde uygulanmıştır. Ayrıca, tasarımı kaydetme, kaydedilmiş önceki bir tasarımı açma veya tasarlanmış olan aralık değerli tip-2 bulanık sistemi Simulink'e aktarma işlemleri de yine bu araç kutusunun ilk arayüzü olan ana ekran ile kolayca yapılabilir. Diğer bir arayüz ise üyelik fonksiyonları editörüdür. Bu arayüz ile giriş ve çıkış değişkenlerine ait üyelik fonksiyonları tasarlanır. Üyelik fonksiyonu tanımlamaları Matlab bulanık mantık araç kutusu fonksiyonlarına yeni parameter tanımlayarak uygulanmıştır. Bu yeni eklenen parametre ile alt üyelik fonksiyonlarının yükseklikleri kolayca tanımlanabilir. Bu geliştirilmiş olan aralık değerli tip-2 bulanık sistemin en büyük artılarından biri de bu üyelik fonksiyonları editörü arayüzü ile yapılan üyelik fonksiyonu tasarımlarıda kullanıcıya sağladığı üstün serbestliktir. Tasarımcı, bu arayüz ile herhangi bir üyeik fonksyionu tanımlarken alt ve üst üyelik fonksiyonlarını istediği gibi kolayca tanımlayabilir. Hatta, alt ve üst üyelik fonksiyonlarının farklı tiplerde tanımlanması da mümkündür. Örneğin, bir üyelik fonksyionu tanımlamasında üst üyelik fonksiyonu gauss tipinde olurken, alt üyelik fonksiyonu üçgen olarak tanımlanabilir. Üyelik fonksyionu tanımlamadaki tek kısıt, alt üyelik fonksiyonunun aitliklerinin her değer için üst üyelik fonksiyonu aitliklerinden küçük olması gerektiğidir. Bu koşul göz önüne alınarak üyelik fonksiyonu tanımlamaları istenilen şekilde yapılabilir. Geliştirilen ve önerilen bu aralık değerli tip-2 bulanık mantık sistemler Matlab/Simulinkk araç kutusunun bir diğer arayüzü ise kural tanımlamarının yapılmasına olanak sağlayan kural editörüdür. Giriş ve çıkış değişkenleri bu arayüzde otomatik olarak gelir ve tasarlanmış olan kural tablosu kolayca uygulanır. Bir diğer arayüz ise, tasarlanmış olan aralık değerli tip-2 bulanık sistemin giriş değişkenleri için elde edilecek olan yüzeyin incelenmesine olanak sağlar. Belirli aralıklarla giriş değişkenleri için aitlikler hesaplanarak bu yüzey elde edilir ve bu arayüze çizdirilir. Tasarlanan ve önerilen bu araç kutusunun en önemli özelliklerinden birisi de Simulink ile entegre olarak geliştirilmiş olmasıdır. Bu amaçla öncelikle araç kutusu için yeni bir Simulink kütüphanesi oluşturulmuştur. Bu kütüphane de iki adet blok bulunmaktadır. İlk blok, araç kutusu ile tasarlanmış olan aralık değerli tip-2 sistemin benzetimde koşturulmasına olanak sağlar. İkinci blok ise bunun yanı sıra tip indirgeme yönteminin seçilmesine imkan sağlar. Aralık değerli tip-2 bulanık sistem Matlab/Simulink araç kutusunun ana arayüzünden sadece bir düğmeye tıklanarak o anki tasarımın otomatik olarak Simulink'e aktarılması mümkündür. Bu şekilde, tasarlamış olan aralık değerli tip-2 bulanık sistem otomatik olarak Simulink ortamına aktarılır, tanımlanmış olan giriş değişkenleri otomatik olarak oluşturulur ve bağlantıları yapılır. Tasarımcı bu sayede istediği sistemi Simulink ortamında oluşturarak benzetim aşamasına kolayca geçebilir.

Son bölümde, geliştirilmiş olan bu aralık değerli tip-2 bulanık sistemler Matlab/Simulink araç kutusu kullanılarak bir uygulama yapılmış ve bazı analizler yapılarak sonuçları paylaşılmıştır. Yapılan uygulamada öncelikle geliştirilmiş olan araç kıtısı kullanılarak bir aralık değerli tip-2 bulanık kontrollör tasarlanmıştır. Bu amaçla öncelikle araç kutusunun ana arayüzünden giriş değişkenleri sayıları belirlenmiştir. Giriş ve çıkış değişkenleri belirlendikten sonra giriş ve çıkış değişkenlerine ait üyelik fonksiyonları üyelik fonksiyonları editörü kullanılarak tanımlanmıştır. Daha sonra, kural editörü arayüzünden kurallar tanımlanmıştır. Bu tasarlanmış olan aralık değerli tip-2 bulanık kontrolör için öncelikle farklı tip indirgeme yöntemleri kullanılarak her bir tip indirgeme yöntemi için kontrol yüzeyleri çizdirilmiş ve sonuçları paylaşılmıştır. Beklendiği gibi Karnik-Mendel algroritması ve onun genişletilmiş algoritmaları aynı yüzeyi vermekte ancak kapalı formdaki tip indirgeme yöntemleri kullanılarak elde edilen yüzeylerde küçük farklılar gözlemlenmiştir. Daha sonra, tasarlanmış olan bu aralık değerli tip-2 bulanık kontrolör Simulink ortamına otomatik olarak aktarılmış ve bir adet de birinci dereceden ölü zamanlı bir sistem de eklenerek benzetimler yapılmıştır. Benzetimlerde tüm tip indirgeme yöntemleri ayrı ayrı benzetimlerde kullanılmış ve aşım, yerleşme zamanı gibi kontol performansları karşılaştırmalı olarak verilmiştir. Daha sonra sistem parametreleri bozulmaya zorlanarak bu durumda benzetimler yapılmış ve farklı tip indirgeme yöntemlerinin yine aşım, yerleşme zamanı gibi kriterler için kontrol performansları bozulmuş sistem için karşılaştırmalı olarak verilmiştir. Bunlara ek olarak, farklı tip indirgeme yöntemleri için hesaplama süreleri giriş değişkenlerinin belirli bir aralıktaki tüm değerleri için eşit sayıda koşarak ölçülmüş, ve her bir tip indirgeme yöntemi için hesaplama süreleri karşılaştırmalı olarak verilmiştir.

Özetle, bu tezde öncelikle tip-1 bulanık mantık kümeler ve sistemler anlatılmıştır. Daha sonra aralık değerli tip-2 bulanık mantık kümeler ve sistemler anlatılarak yaygın olarak kullanılan tip indirgeme yöntemleri detaylı olarak anlatılmıştır. Bu aşamadan sonra aralık değerli tip-2 bulanık mantık sistemler için bir Matlab/Simulink araç kutusu geliştirilmiş ve önerilmiştir. Bu araç kutusu ile, oldukça karmaşık olan aralık değerli tip-2 bulanık sistem tasarımı ilk aşamasında son aşamasına araç kutusunun arayüzleri kullanılarak kolayca ve anlaşılabilir bir şekilde gerçekleştirilebilir. Geliştirilen araç kutusunun nasıl geliştirildiği ve kullanılan arayüzler detaylı olarak anlatılmıştır. Daha sonra geliştiren araç kutusunun Simulink ortamı ile nasıl çalıştığı anlatılmış ve araç kutusu için geliştirilmiş olan Simulink kütüphanesi açıklanarak tasarlanmış olan aralık değerli tip-2 bulanık sistemin Simulink ortamına nasıl aktarılacağı açıklanmıştır. Son bölümde, geliştirilmiş olan bu aralık değerli tip-2 bulanık mantık sistemler Matlab/Simulink araç kutusu ile bir uygulama geliştirilmiş ve farklı tip indirgeme yöntemlerinin kontrol performansları incelemiştir. Ayrıca buna ek olarak farklı tip indirgeme yöntemlerinin hesaplama performansları da incelenerek sonuçları tablolar şekilde verilmiştir.

1. INTRODUCTION

Fuzzy Logic has obtained attention of researchers for last couple of decades. It has opened new scopes both in the academia and the industry site. Fuzzy Logic sets was first introduced by Zadeh in 1965 (Zadeh, 1965). The fuzzy logic sets covers the interval between crisp values while classical logic approach works with only crisp values like true or false. In other words, it is possible to define the linguistic variables like ‘short’, ‘very short’, ‘tall’ and ‘very tall’ with fuzzy logic sets. The fuzzy logic sets proposed by Zadeh in 1965 are known as the type-1 (ordinary) fuzzy sets. The systems that have at least one type-1 fuzzy set are known as Fuzzy Logic Systems (FLSs). The first industrial application example of the type-1 fuzzy logic systems by using the linguistic control rules was introduced by Mamdani in 1974 (Mamdani, 1974). Then, many successful studies of type-1 fuzzy logic systems were introduced in many areas like robotics, modelling and control of the nonlinear systems and making decisions. However, type-1 fuzzy sets are inadequate for defining some uncertainties and nonlinear behaviors because of the crisp membership grades of the employed type-1 fuzzy sets. Therefore, Zadeh introduced type-2 fuzzy sets in 1975 (Zadeh, 1975). Type-2 fuzzy sets are the extension of the ordinary (type-1) fuzzy sets. Also, type-2 fuzzy sets are known as the ‘fuzzy-fuzzy’ sets. The systems that have at least one type-2 fuzzy set are known as type-2 fuzzy logic systems. Type-2 fuzzy sets have one more degree of freedom which suites them for uncertain environments but higher degree of freedom brings more computation complex. Due to high computational cost of the type-2 fuzzy sets, interval type-2 fuzzy sets are proposed as an extension of the type-2 fuzzy sets. Recently, interval type-2 fuzzy sets and systems has obtained more interest on the fuzzy logic theory and related areas.

1.1 Purpose of Thesis

In literature, there are many studies on the interval type-2 fuzzy sets and systems in different areas such as liquid level control, robotics, modelling and control of the nonlinear systems. However, in literature, there is still a lack of useful toolbox to facilitate the use interval type-2 fuzzy logic systems. Therefore, in this thesis, a toolbox for the interval type-2 fuzzy logic systems is introduced to cover the all steps of the implementation of an interval type-2 fuzzy logic system and make it more understandable.

Firstly, type-1 fuzzy sets and systems are explained. Then, interval type-2 fuzzy sets and systems are explained more in detail. The most known type reduction methods are mentioned. The each type reduction method is given more in detail and differences between the type reduction methods are explained.

The proposed toolbox are explained in the next sections. The each user interfaces of the interval type-2 fuzzy systems toolbox is explained. The user interfaces of the toolbox give the opportunity of defining membership functions of the input and output variables, defining rules, and viewing surfaces of the interval type-2 design. In addition, the type reduction methods that are explained in this thesis have been implemented as a separate function in Matlab and embedded to the toolbox. Therefore, desired type reduction method can be applied by choosing it from the menus of the toolbox. Another possibility of the proposed toolbox is automatic exporting the current design to Simulink. It makes easier to start the simulations after completing the interval type-2 fuzzy logic system design.

In summary, the interval type-2 fuzzy logic systems are quite complicated to understand and implement from the initial description phase to the final implementation phase. The proposed toolbox covers the all phases of an interval type-2 fuzzy logic system. The proposed toolbox makes the interval type-2 fuzzy logic system design possible by using the user interfaces of the toolbox. In addition, the different type reduction methods are explained more in detail and the performances of the different type reduction methods compared for different performance criteria.

1.2 Literature Review

A new research area has been opened since the first paper on fuzzy sets published by Zadeh (Zadeh, 1965). The fuzzy logic sets covers the interval between crisp values while classical logic approach works with only crisp values like true or false. In other words, it is possible to define the linguistic variables like ‘short’, ‘very short’, ‘tall’ and ‘very tall’ with fuzzy logic sets. The fuzzy logic sets proposed by Zadeh in 1965 are known as the type-1 (ordinary) fuzzy sets. The systems that have at least one type-

1 fuzzy set are known as type-1 fuzzy logic systems. The first industrial application example of the T1-FLSs by using the linguistic control rules was introduced by Mamdani in 1974 (Mamdani, 1974). The systems that proposed in this study are known as Mamdani-type systems. The another most well known FLSs are Tagaki-Sugeno-Kang FLSs (Takagi and Sugeno, 1985). The Takagi-Sugeno-Kang type systems are widely used in control applications. In addition, Tagaki-Sugeno-Kang type FLSs makes designing and analyzing the fuzzy systems more obvious than Mamdani type (Sugeno and Kang, 1988). Therefore, there are some studies on fuzzy logic control and theory in literature to adopt the Takagi-Sugeno-Kang structure (Eksin et al., 2001; Guzelkaya et al., 2003; Yeşil et al., 2004; Kumbasar et al., 2008). In addition to these studies, there are various successful studies that aim to modelling and controlling the systems by using the fuzzy logic theory (Babuska et al., 2002; Guzelkaya et al., 2003; Precup et. al, 2004; Duan et al., 2008; Ahn et al., 2009; Karasakal et. al, 2013). However, type-1 fuzzy sets are inadequate for defining some uncertainties and nonlinear behaviors because of the crisp membership grades of the employed type-1 fuzzy sets. In literature, there are some studies that demonstrate the limitations of the type-1 fuzzy sets to cover the uncertainties and nonlinear behaviors (Hagras 2004, Wu, 2006). The reason of this limitation is occurring due to crisp value of the membership grade for each input value (Mendel, 2007). Therefore, Zadeh introduced type-2 fuzzy sets in 1975 (Zadeh, 1975). Type-2 fuzzy sets are the extension of the ordinary (type-1) fuzzy sets. In addition, type-2 fuzzy sets are known as the 'fuzzy-fuzzy' sets. The systems that have at least one type-2 fuzzy set are known as type-2 fuzzy logic systems (Mendel, 2000). It has been shown that T2-FSs are much more powerful to cover uncertainties and nonlinearities compared to T1-FLSs (Liang et al., 2000 and Coupland et al., 2007). On the other hand, type-2 fuzzy sets have one more degree of freedom which suites them for uncertain environments but higher degree of freedom brings more computation complex (Liang and Mendel, 2000; Wu and Mendel, 2002; Mendel et al. 2006). Due to high computational cost of the T2-FSs, Interval Type-2 Fuzzy Sets (IT2-FSs) are proposed as an extension of the T2-FSs (Liang at al. 2000; Mendel, 2000). Recently, IT2-FSs and Interval Type-2 Fuzzy Logic Systems (IT2-FLSs) has obtained more interest on the fuzzy logic theory and related areas. The type-reduction mechanism has to be used to map the IT2-FSs into T1-FSs and afterwards, defuzzification operation is applied to obtain a crisp quantity (Liang and Mendel, 2000; Mendel, 2000; Wu and Mendel, 2011). The most widely known

type reduction algorithm is Karnik-Mendel algorithm (Karnik and Mendel, 1999). The fundamental of the Karnik-Mendel algorithm is to find the optimal switching points to reduce the type. The Karnik-Mendel algorithm has to calculate the type-reduced set iteratively because of the impossibility of predetermining the switching points as explicit functions of the inputs (Liang and Mendel, 2000; Wu and Mendel, 2007). Therefore, there are many studies in literature to reduce the computational cost of the Karnik-Mendel (KM) algorithm. The earliest enhancement to the original KM algorithm is Enhanced Karnik-Mendel (EKM) Algorithm (Mendel and Wu, 2009). EKM algorithm have three improvements over the KM algorithm. First of the improvements, EKM algorithms have a better initialization to reduce the number of iterations. Secondly, EKM algorithms made a change for the termination condition of the iterations to remove one unnecessary iteration. Finally, a better computing technique is used to reduce the computational cost of each iteration. The Iterative Algorithm with Stop Condition (IASC) algorithm was proposed as an alternative solution to compute the generalized centroid of an IT2-FS (Duran at al., 2008). First experimental results showed that, IASC algorithm would be faster than EKM algorithms to find the solution. An enhancement of the IASC algorithm was proposed by Wu and Nie (2011). The Enhanced Iterative Algorithm with Stop Condition (EIASC) has two improvements over the IASC algorithm. First of the improvements, EIASC algorithm has new stop condition. And secondly, IASC algorithm has new starting points for iterations. Another enhancement of the KM algorithm is Enhanced Opposite Direction Searching (EODS) algorithm (Hu et al. 2012). EODS algorithm aim to compute the centroid of an IT2-FS by speeding up the KM algorithm. Except of the enhancement of the KM algorithm, there are some studies in literature that finds the closed form solution without need making any iteration. One of these types of methods is the uncertainty bound method (Wu and Mendel, 2002). Another closed-form TR method is Nie-Tan (NT) method (Nie and Tan, 2008). NT method saves the the robustness performance of an IT2 FLS. Therefore, NT methob could be useful for real time applications due to these advantages. The one of the other closed-form TR and defuzzification method for IT2-FLSs is Begain-Melek-Mendel (BMM) method (Begain at al., 2008). BMM has two adjustable parameters and the type reduction performance can be adjusting easily by changing these two parameters. Although there are many studies in literature on IT2-FSs and IT2-FLSs, there is a lack of useful toolboxes to facilitate the use of IT2-FLSs. A toolbox for IT2-FLSs was proposed to literature (Castro at al., 2007). Another toolbox for IT2-FLSs was proposed Ozek and Zuhtu in 2008. In addition, a toolbox has been introduced for the IT2-FLSs in the Matlab environment in 2008 by Zamani et al. However, there is still a lack of useful toolboxes to cover all steps of an implementation of the IT2-FLSs from the initial description phase the final implementation phase through implementing the type reduction methods and exporting to a simulation environment.

2. TYPE-1 FUZZY SETS AND FUZZY LOGIC SYSTEMS

Professor Lotfi A. Zadeh at the University of California at Berkeley introduced the fuzzy theory into the scientific literature in 1965. He published his study with a paper titled "Fuzzy Sets" in the journal Information and Control (Zadeh, 1965). In this study, he proposed a fuzzy set theory that operated between the range [0, 1]. The difference between Boolean logic and fuzzy logic is Boolean logic results can be '0' or '1', however, fuzzy logic results can take any value between '0' and '1'. In other words, the values between sharp results like true and false are defined by the fuzzy. Therefore, fuzzy sets can define some commonly used concept in daily life like "very tall", "tall", "short" and "very short". Fuzzy logic uses linguistic variables and it can define level and degrees of linguistic variables.

2.1 Type-1 Fuzzy Sets

The basis of the fuzzy logic is the fuzzy sets. A fuzzy set is a set that can include elements with only a partial degree of membership. Therefore, a fuzzy set does not have a crisp boundary. When the universe, D_X, is continuous and infinite, a fuzzy set X in D_X can be defined as follows:

$$X = \{(x, \mu_X(x)) | x \epsilon D_X\} \quad \textbf{(2.1)}$$

In the equation given above, $\mu_X\ (x)$ is the MF of x in X, and it represents the degree of membership of the x belongs to X. The MF $(\mu_X\ (x))$ maps each component in D_X to a continuous unit interval [0, 1]. If $\mu_X\ (x)$ value is close to zero, it means that belonging of the x to fuzzy set X is lower degree. If $\mu_X\ (x)$ value is close to one, it means that belonging of the x to to fuzzy set X is higher degree.

Many different Membership Functions (MFs) defined in literature. The most known and used MFs are triangular, trapezoidal and Gaussian MFs.

A type-1 (ordinary) triangular MF is defined by three parameters:$\{a, b, c\}$. Here, a indicates the left endpoint, b indicates the cenral point, and c indicates the right endpoint of the triangular. A type-1 triangular MF can be seen in Figure 2.1.

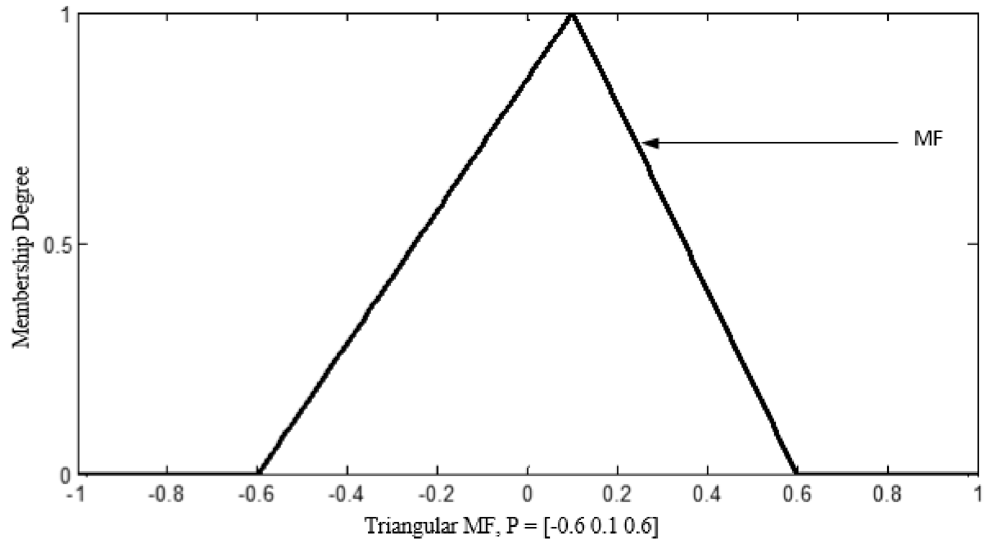


Figure 2.1 : Triangular type-1 MF.

A type-1 trapezoidal MF is defined by four parameters: $\{a, b, c, d\}$. The a, b, c, d (where $a < b < c < d$) parameters define the corner points of the trapezoidal MF as it can be seen in Figure 2.2.

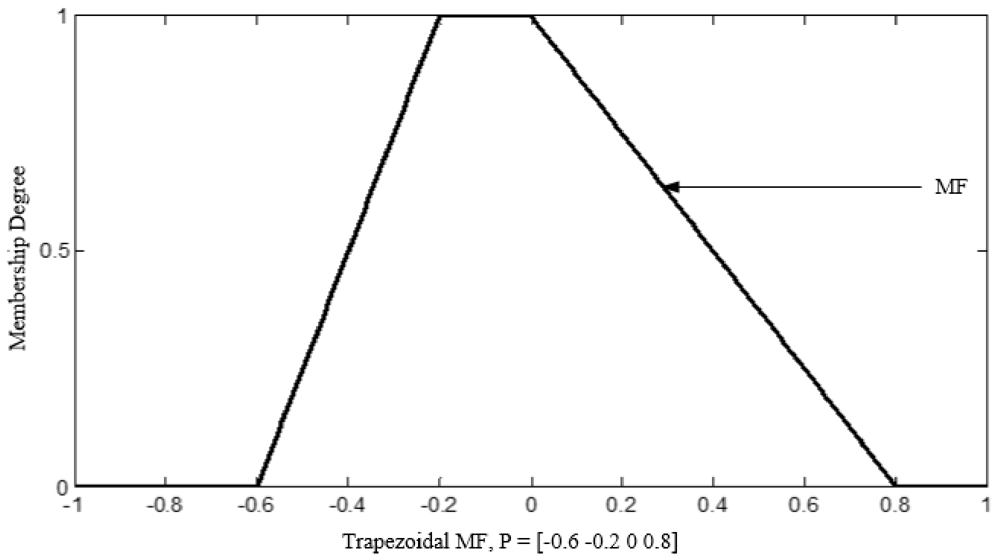


Figure 2.2 : Trapezoidal type-1 MF.

A type-1 Gaussian MF is defined by two parameters: $\{c, \sigma\}$. Here, x represents the center of the MF and σ represents the width of the MF. A type-1 Gaussian MF can be seen in Figure 2.3.

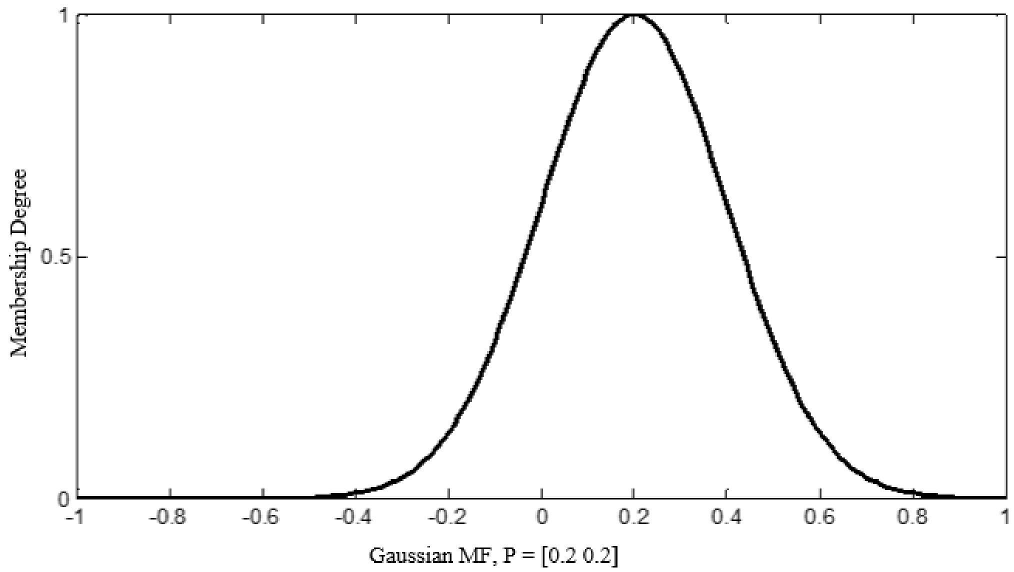


Figure 2.3 : Gaussian type-1 MF.

2.2 Type-1 Fuzzy Logic System

In this section, Type-1 Fuzzy Logic Systems (T1-FLSs) are described. T1-FLSs operate human's knowledge by using the fuzzy sets and IF-THEN rules. A classical type-1 fuzzy system structure consists of four components:

- Fuzzification
- Rule base
- Inference mechanism
- Defuzzification

The general structure of the T1-FLS is illustrated in the Figure 2.4.

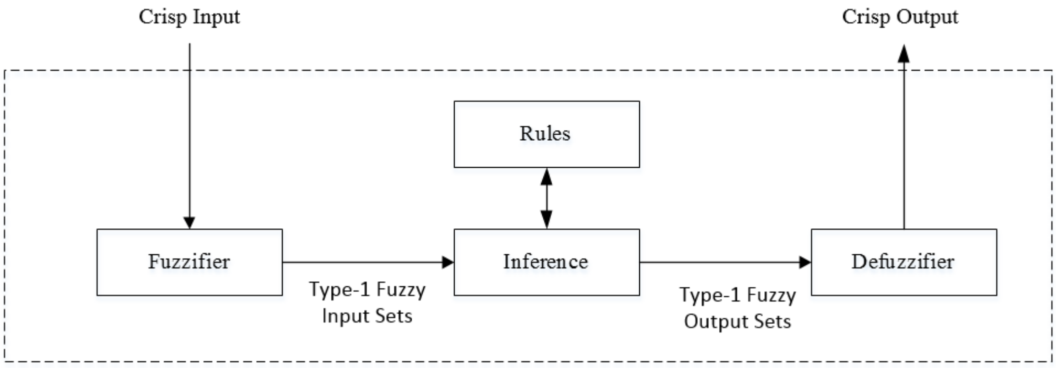


Figure 2.4 : Type-1 fuzzy logic system structure

The fuzzification operation is implemented by the fuzzifier block. The fuzzifier block converts crisp values to degree of membership for linguistic terms of fuzzy sets. Each MF by a membership degree represents a related linguistic.

The fuzzy rule base consists of IF-THEN fuzzy rules. Fuzzy rules provide the relationship between inputs and outputs of the fuzzy system. Let "x" be the input variables and "y" is the output variable, an example fuzzy rule (R_i) for this system can be obtained like this:

$$\text{Rule } \tilde{R}^n\text{: If } x_1 \text{is } \tilde{X}_1^n \text{ and } \dots \text{ and } x_I \text{is } \tilde{X}_I^n \text{ , then } y \text{ is } Y^n \tag{2.2}$$

In this example fuzzy rule, $\tilde{X}_1^n$ and … and $\tilde{X}_l^n$ represent fuzzy sets of the input variables, and Y^n represents the membership set of output variable.

One of the components of the type-1 fuzzy system is inference mechanism. Fuzzy inference mechanism is one of the most important component of the fuzzy systems. Generally, fuzzy inference mechanism makes inferences like decision, suggestion and inference of human. Fuzzy inference mechanism identifies the firing of the each rule in the rule base and calculates the outputs for a specific time period by using the input variable.

The output value that was calculated by inference mechanism is fuzzy set. This fuzzy output value is converted to crisp value by the defuzzification block. In literature, there are different methods that are used for defuzzification. The most known methods are center of area, mean of maximum and height defuzzification.

3. INTERVAL TYPE-2 FUZZY SETS AND FUZZY LOGIC SYSTEMS

Zadeh proposed the type-2 fuzzy sets (T2-FSs) as an extension of the type-1 fuzzy sets (T1-FSs) in 1975 (Zadeh, 1975). The memberships in a T1-FS are crisp values; however, the memberships in a T2-FS are T1-FS (Wagner and Hagras, 2010). Recently, T2-FSs and type-2 fuzzy logic systems (T2-FLSs) are used in many areas. However, it is not so easy to describe mathematically the T2-FSs due to their additional dimension (Mendel, 2007). Therefore, people are interested in IT2-FSs whose memberships are interval instead of T1-FSs in a general T2-FS (Mendel, 2000). This brings many advantages like simplicity and reduced computational cost.

3.1 Interval Type-2 Fuzzy Sets

IT2-FSs are special case of the T2-FSs. Due to their reduced computational cost, there are many study about the IT2-FSs in literature. In this section, background materials on the IT2-FSs are given. The input variable (x) which was introduced in T1-FSs is also defined as the primary variable (x), has domain $D_{\tilde{X}}$, for T2-FSs and IT2-FSs. An IT2-FS $\tilde{X}$ is characterized by its MF $\mu_{\tilde{X}}(x,u)$. An IT2-FS $\tilde{X}$ can be expressed as follows:

$$\tilde{X} = \int_{x \in D_{\tilde{X}}} \int_{u \in J_x \subseteq [0,1]} \frac{\mu_{\tilde{X}}(x,u)}{(x,u)} \qquad \textbf{(3.1)}$$

In this equation, x is called as the primary variable and it has domain $D_{\tilde{X}}$, $\mu \in [0,1]$ is called as the secondary variable and it has domain $J_x \subseteq [0,1]$ at each $x \in D_{\tilde{X}}$. J_x is called the support of the secondary MF, and, the amplitude of $\mu_{\tilde{X}}(x,u)$, called a secondary grade of $\tilde{X}$, equals 1 for $\forall x \in D_{\tilde{X}}$ and $\forall u \in J_x \subseteq [0,1]$ (Wu, 2013).

An example of an IT2-FS can be seen in Figure 3.1. As seen in the Figure 3.1, unlike a T1-FS, the membership of an IT2-FS is an interval. In addition, the IT2-FS is bounded from above and below by two T1-FS. The bound of the above is called as

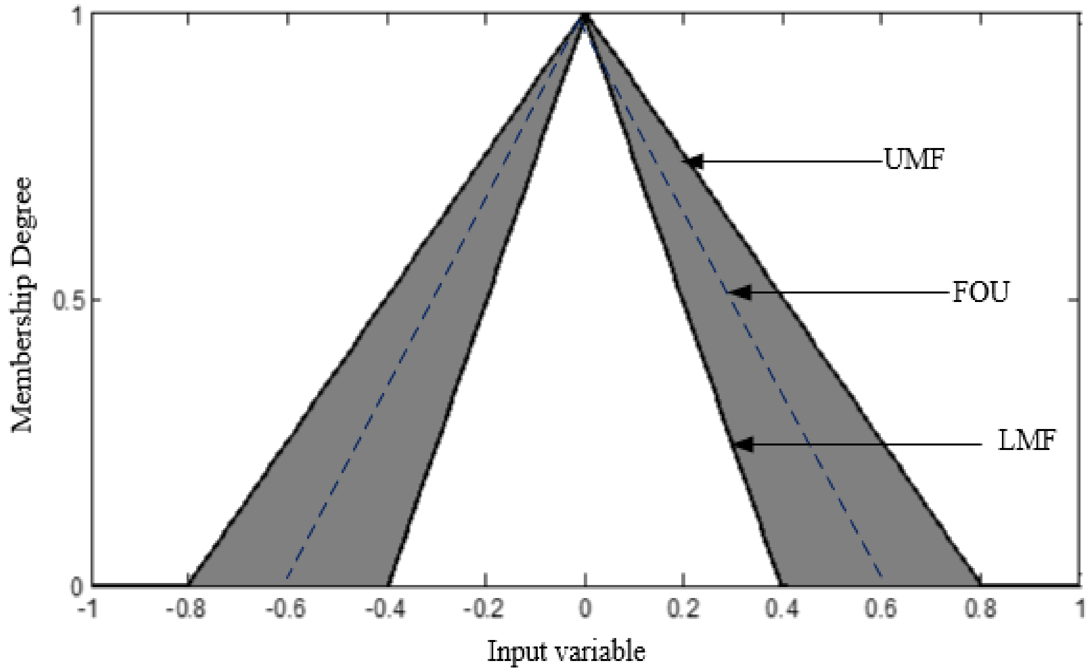


Figure 3.1 : Triangular interval type-2 MF.

Upper Membership Function (UMF) and the bound which locates below is called as Lower Membership Function (LMF). The area between UMF and LMF is called as Footprint of Uncertainty (FOU). Trapezoidal and Gaussian IT2-FSs can be seen in Figure 3.2 and Figure 3.3, respectively.

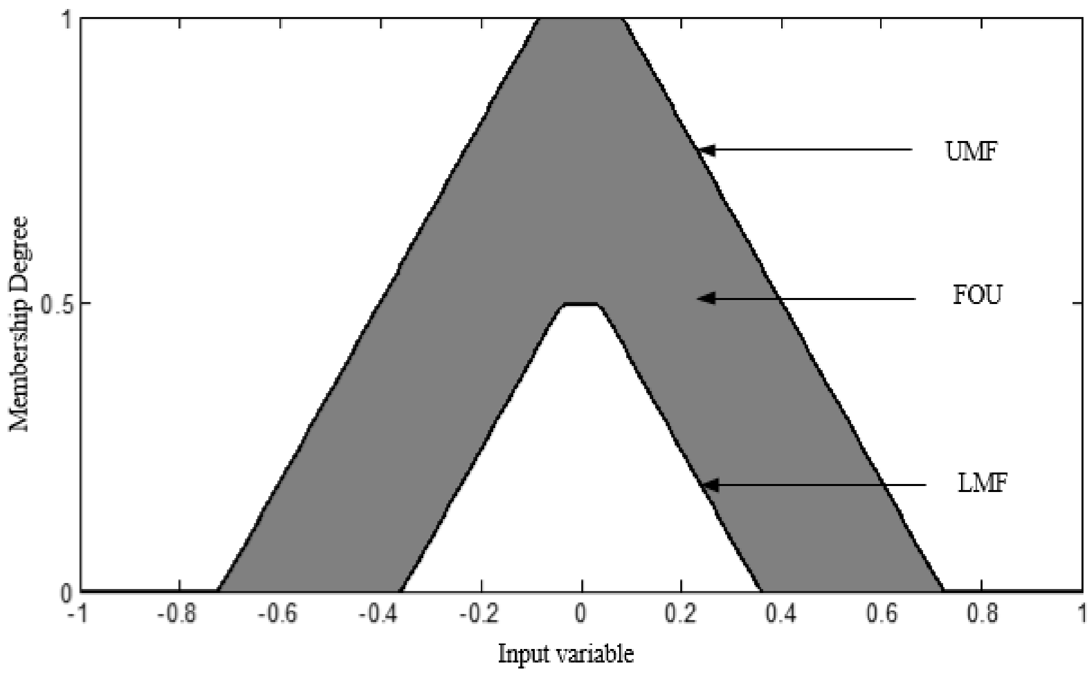


Figure 3.2 : Trapezoidal interval type-2 MF.

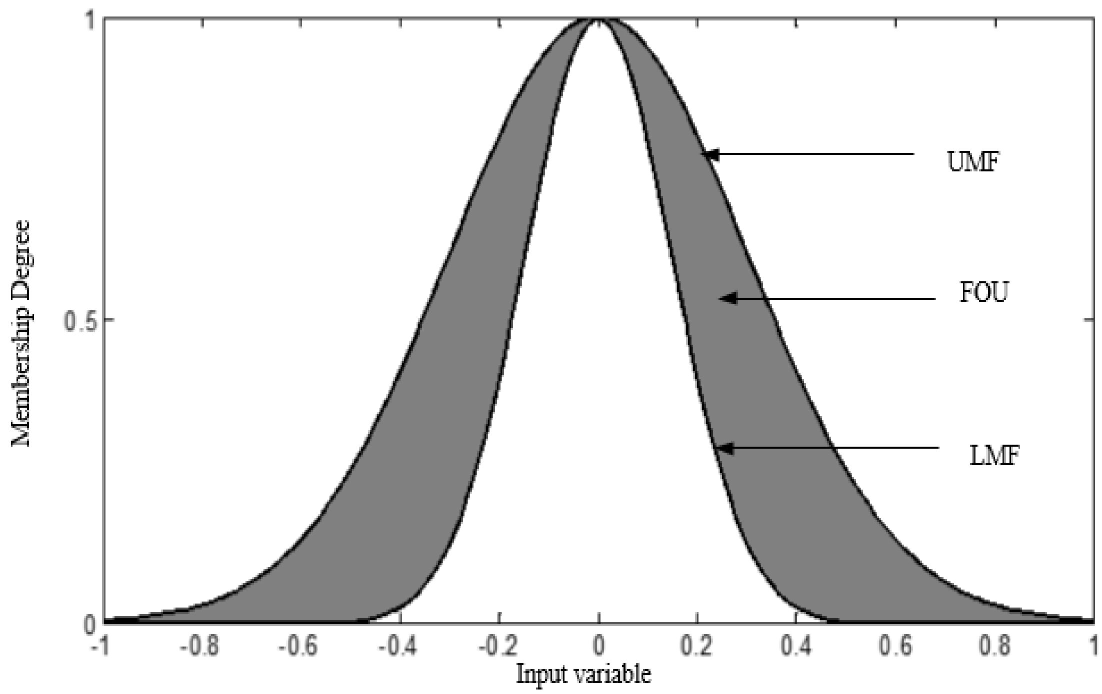


Figure 3.3 : Gaussian interval type-2 MF.

3.2 Interval Type-2 Fuzzy Logic Systems

An IT2-FLS is a FLS that contains at least one IT2-FSs. IT2-FLSs (or T2FLSs) consist of five main parts:

- Fuzzification
- Rule base
- Inference mechanism
- Type Reducer
- Defuzzification

As can be seen, IT2-FLSs have an additional part that is called as type-reducer, while T1-FLSs do not have type-reducer. Type-reducer is necessary for the IT2-FLSs because defuzzifier block can operates by using the type-1 information, so type-2 information have to be converted to type-1 information before the defuzzifier block. The output of the type-reducer block is called as type-reduced set and then defuzzifier uses the type-reduced set for defuzzification process. A simple block diagram of an IT2-FLS which is special case of T2-FLSs can be seen in Figure 3.4.

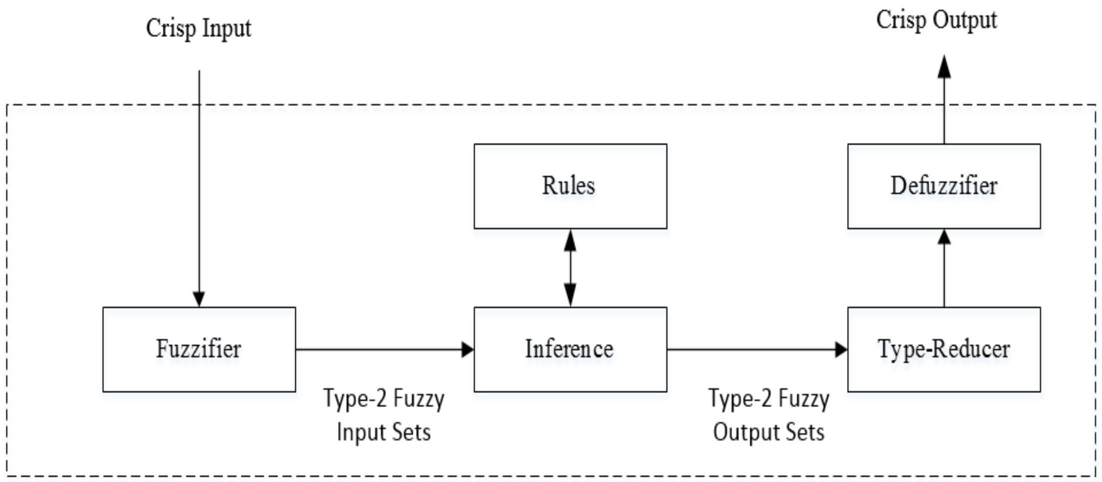


Figure 3.4 : Type-2 fuzzy logic system structure.

Let's consider a rule base of an IT2-FLS that contains N rules as below:

$$\tilde{R}^n\text{: If } x_1 \text{is } \tilde{X}_1^n \text{ and } \dots \text{ and } x_I \text{is } \tilde{X}_I^n \text{ , then } y \text{ is } Y^n \quad \textbf{(3.2)}$$

Where $\tilde{X}_i^n(i = 1, \dots, I)$ are IT2-FSs and $Y^n = [\underline{y}^n, \overline{y}^n]$ is the interval output. For an input vector $x = (x_1,\ x_2, \dots, x_3)$ classical steps of the IT2-FLS can be summarized below:

1. Compute the membership interval of each x_i' on each $\tilde{X}_i^n$, i.e. $[\underline{\mu}_{\tilde{X}_i^n}(x_i'), \overline{\mu}_{\tilde{X}_i^n}(x_i')], i = 1,2, \dots I, n = 1,2, \dots N$.

2. Calculate the rule firing interval of the n^{th}, $F^n(\boldsymbol{x}')$:

$$F^n(\boldsymbol{x}') \equiv \left[\underline{f}^n, \overline{f}^n\right], n = 1, \dots, N \quad \textbf{(3.3)}$$

where

$$\begin{aligned} \underline{f}^n &= [\underline{\mu}_{\tilde{X}_1^n}(x_1') * \dots \times \underline{\mu}_{\tilde{X}_I^n}(x_I')] \\ \underline{f}^n &= [\overline{\mu}_{\tilde{X}_1^n}(x_1') \times \dots \times \overline{\mu}_{\tilde{X}_I^n}(x_I')] \end{aligned} \quad \textbf{(3.4)}$$

3. Perform TR to combine $F^n(\boldsymbol{x}')$ and the corresponding rule consequents. The most commonly used TR is the center-of-sets type reducer defined as:

$$Y_{cos}(\boldsymbol{x}') = \bigcup_{\substack{f^n \in F^n(x') \\ y^n \in Y^n}} \frac{\sum_{n=1}^N y^n f^n}{\sum_{n=1}^N f^n} = [y_l, y_r] \quad \textbf{(3.5)}$$

where y_l and y_r are defined as:

$$y_l = \frac{\sum_{n=1}^{L} \underline{y}^n \overline{f}^n + \sum_{n=L+1}^{N} \underline{y}^n \underline{f}^n}{\sum_{n=1}^{L} \overline{f}^n + \sum_{n=L+1}^{N} \underline{f}^n} \quad \textbf{(3.6)}$$

$$y_r = \frac{\sum_{n=1}^{R} \overline{y}^n \underline{f}^n + \sum_{n=R+1}^{N} \overline{y}^n \overline{f}^n}{\sum_{n=1}^{R} \underline{f}^n + \sum_{n=R+1}^{N} \overline{f}^n} \quad \textbf{(3.7)}$$

Here, R and L are the switching points that can be found via the iterative KM algorithm or its enhancements mentioned in this thesis.

4. Compute the defuzzified (crisp) output as follows:

$$y = \frac{y_l + y_r}{2} \quad \textbf{(3.8)}$$

3.3 Type Reduction Methods

Type reducer converts the T2-FSs to T1-FSs before the defuzzifier. There are many Type Reduction (TR) method proposed in literature for IT2-FLSs. In this section, most widely used type reduction methods are explained. The most widely used common type reduction method is the iterative Karnik and Mendel algorithm (Karnik and Mendel, 1999). In literature, there are some studies that aim to enhance the KM algorithm to improve the performance. Some of them are explained in detail in the next subsections. In addition, there are some studies that aim to reduce the type-2 sets to type-1 sets by using the approximations. These type of methods find approximations results and does not need any iteration. Therefore, they find the approximation results faster.

3.3.1 Karnik-Mendel algorithm

The most known TR method that is used to reduce the type is KM algorithm (Karnik and Mendel, 1999). The KM algorithm consists of two main parts. The aim of the one part is to compute the left point of the output (y_l) and the other part is to compute the right point of the output (y_r). The calculation of the y_l and y_r includes following steps.

Define:

$$f_l(k) = \frac{\sum_{n=1}^{k} \underline{y}^n \overline{f}^n + \sum_{n=k+1}^{k} \underline{y}^n \underline{f}^n}{\sum_{n=1}^{k} \overline{f}^n + \sum_{n=k+1}^{k} \overline{f}^n} \quad \textbf{(3.6)}$$

$$f_r(k) = \frac{\sum_{n=1}^{k} \overline{y}^n \underline{f}^n + \sum_{n=k+1}^{k} \overline{y}^n \overline{f}^n}{\sum_{n=1}^{k} \underline{f}^n + \sum_{n=k+1}^{k} \overline{f}^n} \quad \textbf{(3.7)}$$

where k is an integer in $[1, N - 1]$, and $\{\underline{y}^n\}$ and $\{\overline{y}^n\}$ are sorted to be increasing form, respectively. Then, y_l and y_r can be expressed as (Mendel, 2011):

$$y_l = \underbrace{min}_{k \in [1, N-1]} f_l(k) = f_l(L) \quad \textbf{(3.8)}$$

$$y_l = \frac{\sum_{n=1}^{L} \underline{y}^n \overline{f}^n + \sum_{n=L+1}^{N} \underline{y}^n \underline{f}^n}{\sum_{n=1}^{k} \overline{f}^n + \sum_{n=L+1}^{N} \overline{f}^n} \quad \textbf{(3.9)}$$

$$y_r = \underbrace{max}_{k \in [1, N-1]} f_r(k) = f_r(R) \quad \textbf{(3.10)}$$

$$y_r = \frac{\sum_{n=1}^{R} \overline{y}^n \underline{f}^n + \sum_{n=R+1}^{N} \overline{y}^n \overline{f}^n}{\sum_{n=1}^{R} \underline{f}^n + \sum_{n=R+1}^{N} \overline{f}^n} \quad \textbf{(3.11)}$$

where L and R are the switching points satisfying:

$$\underline{y}^L \leq y_l \leq \underline{y}^{L+1} \quad \textbf{(3.12)}$$

$$\overline{y}^R \leq y_r \leq \overline{y}^{R+1} \quad \textbf{(3.13)}$$

In addition, two mentioned inequalities in the study that published by Mendel (2011) are:

$$\underline{y}^L \leq y_l \leq \underline{y}^{L+1} \quad \textbf{(3.14)}$$

$$\overline{y}^R \leq y_r \leq \overline{y}^{R+1} \quad \textbf{(3.15)}$$

because there can be some elements that duplicated for $\{\underline{y}^n\}$ and $\{\overline{y}^n\}$.

The steps pf the KM algorithm to compute y_l and y_r is given in Table 3.1. As seen in this table, the main aim of the KM algorithm is to find the switch points for y_l and y_r.

Table 3.1 : Karnik-Mendel algorithm.

Step	For computing y_l	For computing y_r
1.	Initialize $f^n = \frac{\underline{f}^n + \overline{f}^n}{2}$ And compute $y = \frac{\sum_{n=1}^{N} \underline{y}^n f^n}{\sum_{n=1}^{N} f^n}$	Initialize $f^n = \frac{\underline{f}^n + \overline{f}^n}{2}$ And compute $y = \frac{\sum_{n=1}^{N} \overline{y}^n f^n}{\sum_{n=1}^{N} f^n}$
2	Find $l \in [1, N-1]$ s.t. $\underline{y}^l \leq y_l \leq \underline{y}^{l+1}$	Find $r \in [1, N-1]$ s.t. $\overline{y}^r \leq y_r \leq \overline{y}^{r+1}$
3	Set $f^n = \begin{cases} \overline{f}^n, n \leq l \\ \underline{f}^n, n > l \end{cases}$ And compute $y' = \frac{\sum_{n=1}^{N} \underline{y}^n f^n}{\sum_{n=1}^{N} f^n}$	Set $f^n = \begin{cases} \underline{f}^n, n \leq r \\ \overline{f}^n, n > r \end{cases}$ And compute $y' = \frac{\sum_{n=1}^{N} \overline{y}^n f^n}{\sum_{n=1}^{N} f^n}$
4	If $y' = y$, stop set $y_l = y$ and $L = l$; otherwise, set $y = y'$ otherwise, set $y = y$ and go to Step 2.	If $y' = y$, stop set $y_r = y$ and $R = r$; otherwise, set $y = y'$ otherwise, set $y = y$ and go to Step 2.

3.3.2 Enhanced Karnik-Mendel algorithm

The earliest enhancement to the original KM algorithm is the EKM algorithm (Mendel and Wu, 2009). The original KM algorithm make iteration to converge to the results. EKM algorithm aim to reduce the number of iterations and computational cost of original KM algorithm. The steps of the EKM algorithm are given in Table 3.2.

Table 3.2 : Enhanced Karnik-Mendel algorithm.

Step	For computing y_l	For computing y_r
1	Set $l = [N/2.4]$ (the nearest integer to $[N/2.4]$) and compute $a = \sum_{n=1}^{l} \underline{y}^n \overline{f}^n + \sum_{n=l+1}^{N} \underline{y}^n \underline{f}^n$ $b = \sum_{n=1}^{l} \overline{f}^n + \sum_{n=l+1}^{N} \underline{f}^n$ $y = a/b$	Set r= $[N/1.7]$ (the nearest integer to $[N/1.7]$) and compute $a = \sum_{n=1}^{r} \overline{y}^n \underline{f}^n + \sum_{n=l+1}^{N} \overline{y}^n \overline{f}^n$ $b = \sum_{n=1}^{r} \underline{f}^n + \sum_{n=r+1}^{N} \overline{f}^n$ $y = a/b$
2	Find $l' \in [1, N-1]$ such that $\underline{y}^{l'} < y < \underline{y}^{l'+1}$	Find $r' \in [1, N-1]$ such that $\overline{y}^{r'} < y < \overline{y}^{r'+1}$
3	If $l' = l$, stop and set $y_l = y$ and $L = l$; otherwise, continue.	If $r' = r$, stop and set $r = y$ and $R = r$; otherwise, continue.
4	Compute $s = sign(l' - l)$ $a' = a + s \sum_{n=\min(l,l')+1}^{\max(l,l')} \underline{y}^n (\overline{f}^n - \underline{f}^n)$ $b' = b + s \sum_{n=\min(l,l')+1}^{\max(l,l')} \overline{f}^n - \underline{f}^n$ $y' = a'/b'$	Compute $s = sign(r' - r)$ $a' = a + s \sum_{n=\min(r,r')+1}^{\max(r,r')} \overline{y}^n (\overline{f}^n - \underline{f}^n)$ $b' = b + s \sum_{n=\min(r,r')+1}^{\max(r,r')} \overline{f}^n - \underline{f}^n$ $y' = a'/b'$
5	Set $y = y'$, $a = a'$, $b = b'$ and $l = l'$. Go to Step 2.	Set $y = y'$, $a = a'$, $b = b'$ and $r = r'$. Go to Step 2.

EKM algorithm has three improvements over the KM algorithm. First of the improvements, EKM algorithm has a better initialization to reduce the number of iterations. Secondly, EKM algorithm made a change for the termination condition of the iterations to remove one unnecessary iteration. Finally, a better computing technique is used to reduce the computational cost of each iteration.. Simulations result show that EKM algorithm saves the computational time from %23 and up to %39 (Wu and Mendel, 2009).

3.3.3 Iterative algorithm with stop condition

The steps of the IASC algorithm is illustrated Table 3.3. The IASC algorithm for

Table 3.3 : Iterative algorithm with stop condition.

Step	For computing y_l	For computing y_r
1	Initialize	Initialize
	$a = \sum_{n=1}^{N} \underline{y}^n \underline{f}^n$	$a = \sum_{n=1}^{N} \overline{y}^n \overline{f}^n$
	$b = \sum_{n=1}^{l} \underline{f}^n$	$b = \sum_{n=1}^{l} \overline{f}^n$
	$y_l = \underline{y}^N$	$y_r = \overline{y}^l$
	$l = 0$	$r = 0$
2	Compute	Compute
	$l = l + 1$	$r = r + 1$
	$a = a + \underline{y}^l (\overline{f}^l - \underline{f}^l)$	$a = a + \overline{y}^l (\overline{f}^l - \underline{f}^l)$
	$b = b + \overline{f}^l - \underline{f}^l$	$b = b - \overline{f}^r + \underline{f}^r$
	$c = a/b$	$c = a/b$
3	If $c > y_l$, set $L = l - 1$ and stop; otherwise,	If $c < y_r$, set $R = r - 1$ and stop; otherwise,
	$y_l = c$ and go to step 2.	$y_r = c$ and go to step 2.

computing the generalized centroid of an IT2-FS is proposed by Duran, Bernal and Melgarejo in 2008. This algorithm based on that $f_l(k)$ monotonically decreases firstly and then monotonically increases with the increase of k, and $f_r(k)$ monotonically increases firstly and then monotonically decreases with the increase of k. By using this, the IASC algorithm counts the switch point for y_l from $1\ to\ N-1$ until $f_l(k)$ stops decreasing, at which point y_l is obtained. Similarly, same steps for computing the y_r are applied. The IASC algorithm was proposed as an alternative solution to compute the generalized centroid of an IT2-FS. First experimental results showed that, IASC algorithm would be faster than EKM algorithm (Duran at al., 2008).

3.3.4 Enhanced iterative algorithm with stop condition

The steps of the EIASC algorithm is illustrated in Table 3.4. The EIASC algorithm

Table 3.4 : Enhanced iterative algorithm with stop condition.

Step	For computing y_l	For computing y_r
1	Initialize $a=\sum_{n=1}^{N}\underline{y}^n\underline{f}^n$ $b=\sum_{n=1}^{l}\underline{f}^n$ $L=0$	Initialize $a=\sum_{n=1}^{N}\overline{y}^n\underline{f}^n$ $b=\sum_{n=1}^{l}\underline{f}^n$ $R=N$
2	Compute $L=L+1$ $a=a+y^L(\overline{f}^L-\underline{f}^L)$ $b=b+\overline{f}^L-\underline{f}^L$ $y_1=a/b$	Compute $L=L+1$ $a=a+y^L(\overline{f}^L-\underline{f}^L)$ $b=b+\overline{f}^L-\underline{f}^L$ $y_1=a/b$
3	If $y_l\le y^{L+1}$, stop; otherwise, go to step 2.	If $y_l\le y^{L+1}$, stop; otherwise, go to step 2.

which is the enhancement of the IASC algorithm was proposed by Wu and Nie (2011). The EIASC has two improvements over the IASC algorithm. First of the improvements, EIASC algorithm has new stop condition. And secondly, IASC algorithm starts from the switch point 1 to compute the y_l and y_r. However, EIASC algortihm starts form the switch point $L = N - 1$ to compute the y_l and $R = N - 1$ to compute the y_r.

3.3.5 Enhanced opposite direction searching algorithm

EODS algorithms aim to compute the centroid of an IT2-FS by speeding up the KM algorithm (Hu et al. 2012) .proposed two algorithms to speed up the KM algorithms. The steps of the EODS algorithm are given in Table 3.5. This algorithm is an enhanced version of the original Opposite Direction Searching (ODS) algorithm (Hu at al. 2010). The EODS algorithms improved the ODS algorithms by using the new two high-speed formulae for calculating the centroid endings. Some of the advantages of the EODS algorithms can be summarized as follows:

i. The computation and comparison complexity of the EODS algorithm is O(N);

ii. EODS algorithm saves the computational costs about 50% when compared with ODS algorithm.

iii. EODS algorithm saves the computational costs about 67% to 80% when compared with EKM, EODS algorithms.

iv. The computational cost of EODS algorithm is very closely to that of an estimated lower bound.

If another algorithm exists that is more efficient than EODS, then the computational cost that can be reduced would not be more than N additions or subtractions (Hu at al., 2012).

Table 3.5 : Enhanced opposite direction searching algorithm.

Step	For computing y_l	For computing y_r
1	Initialize $m = 2, n = N - 1$ and compute	Initialize $m = 2, n = N - 1$ and compute

Table 3.6 : Enhanced opposite direction searching algorithm (continued).

	$S_l = (\underline{y}^m - \underline{y}^1)\overline{f}^1$ $S_r = (\underline{y}^N - \underline{y}^n)\overline{f}^N$ $F_l = \underline{f}^N, F_r = \overline{f}^1$	$S_l = (\overline{y}^m - \overline{y}^1)\underline{f}^1$ $S_r = (\overline{y}^N - \overline{y}^n)\underline{f}^N$ $F_l = \underline{f}^1,\ F_r = \overline{f}^N$
2	If $m = n$ then go to step 4.	If $m = n$ then go to step 4.

3	If $S_l > S_r$, then
	$F_l = F_l + \underline{f}^n$
	$n = n - 1$
	$S_r = S_r + F_l(\underline{y}^{n+1} - \underline{y}^n)$
	else
	$F_r = F_r + \overline{f}^m$
	$m = m + 1$
	$S_l = S_l + F_r(\underline{y}^m - \underline{y}^{m-1})$
	Go to step 2.
		If $S_l > S_r$, then
		$F_r = F_r + \overline{f}^n$
		$n = n - 1$
		$S_r = S_r + F_r(\overline{y}^{n+1} - \overline{y}^n)$
		else
		$F_l = F_l + \underline{f}^m$
		$m = m + 1$
		$S_l = S_l + F_l(\overline{y}^m - \overline{y}^{m-1})$
		Go to step 2.

4	If $S_l \leq S_r$, $L = m$ $F_r = F_r + \overline{f}^m$ else $L = m - 1$ $F_l = F_l + \underline{f}^m$	If $S_l \leq S_r$, $R = m$ $F_l = F_l + \underline{f}^m$ else $R = m - 1$ $F_r = F_r + \overline{f}^m$
5	$y_l = \underline{y}^m + \dfrac{S_r - S_l}{F_r + F_l}$	$y_r = \overline{y}^m + \dfrac{S_r - S_l}{F_r + F_l}$

3.3.6 Wu-Mendel uncertainty bound method

The uncertainty bound method was proposed by Wu and Mendel (2002). This method computes the output of the IT2 FLS by using the following equations:

$$y_l = \frac{\underline{y}_l + \overline{y}_l}{2} \tag{3.16}$$

$$y_r = \frac{\underline{y}_r + \overline{y}_r}{2} \tag{3.17}$$

Where

$$\overline{y_l} = min\left\{\underline{y}^{(0)}, \underline{y}^{(N)}\right\} \tag{3.18}$$

$$\overline{y_r} = max\left\{\overline{y}^{(0)}, \overline{y}^{(N)}\right\} \tag{3.19}$$

$$\underline{y}_l = \overline{y}_l - \frac{\sum_{n=1}^{N}(\overline{f}^n - \underline{f}^n)}{\sum_{n=1}^{N}\overline{f}^n \sum_{n=1}^{N}\underline{f}^n} * \frac{\sum_{n=1}^{N}\underline{f}^n(\underline{y}^n - \underline{y}^1)\sum_{n=1}^{N}\overline{f}^n(\underline{y}^N - \underline{y}^n)}{\sum_{n=1}^{N}\underline{f}^n(\underline{y}^n - \underline{y}^1) + \sum_{n=1}^{N}\overline{f}^n(\underline{y}^N - \underline{y}^n)} \tag{3.20}$$

$$\overline{y}_r = \underline{y}_r - \frac{\sum_{n=1}^{N}(\overline{f}^n - \underline{f}^n)}{\sum_{n=1}^{N}\overline{f}^n \sum_{n=1}^{N}\underline{f}^n} * \frac{\sum_{n=1}^{N}\overline{f}^n(\overline{y}^n - \overline{y}^1)\sum_{n=1}^{N}\underline{f}^n(\overline{y}^N - \overline{y}^n)}{\sum_{n=1}^{N}\overline{f}^n(\overline{y}^n - \overline{y}^1) + \sum_{n=1}^{N}\underline{f}^n(\overline{y}^N - \overline{y}^n)} \tag{3.21}$$

in which

$$\underline{y}^{(0)} = \frac{\sum_{n=1}^{N}\underline{y}^n\underline{f}^n}{\sum_{n=1}^{N}\underline{f}^n} \tag{3.22}$$

$$\underline{y}^{(\underline{N})} = \frac{\sum_{n=1}^{N}\underline{y}^n\overline{f}^n}{\sum_{n=1}^{N}\overline{f}^n} \tag{3.23}$$

$$\overline{y}^{(0)} = \frac{\sum_{n=1}^{N}\overline{y}^n\underline{f}^n}{\sum_{n=1}^{N}\underline{f}^n} \tag{3.24}$$

$$\overline{y}^{(\underline{N})} = \frac{\sum_{n=1}^{N}\overline{y}^n\overline{f}^n}{\sum_{n=1}^{N}\overline{f}^n} \tag{3.25}$$

The uncertainty bound method does not require $\{\underline{y}_n\}$ and $\{\overline{y}_n\}$ to be sorted unlike the KM algorithm. However, the method still needs to identify the minimum and maximum of $\{\underline{y}_n\}$ and $\{\overline{y}_n\}$. This method is not an iterative method, so it calculates the output very fast.

3.3.7 Nie-Tan method

Another closed-form TR method is NT method (Nie and Tan, 2008). The output is calculating by using the equation below:

$$y = \frac{\sum_{n=1}^{N} y^n(\underline{f}^n + \overline{f}^n)}{\sum_{n=1}^{N}(\underline{f}^n + \overline{f}^n)} \quad \textbf{(3.26)}$$

NT method does not require $\{y^n\}$ to be sorted. NT method is a computationally efficient alternative to the KM algorithm. Difference between the defuzzified outputs calculated by using the KM algorithm and NT method is minimal. In addition, NT method saves the robustness performance of an IT2 FLS. Therefore, NT method could be useful for real time applications due to these advantages.

3.3.8 Begain-Melek-Mendel method

Another closed-form TR and defuzzification method for IT2-FLSs is BMM method (Begain at al., 2008). BMM method calculates the output by using the following equation:

$$y = \alpha \frac{\sum_{n=1}^{N} \underline{f}^n y^n}{\sum_{n=1}^{N} \underline{f}^n} + \beta \frac{\sum_{n=1}^{N} \overline{f}^n y^n}{\sum_{n=1}^{N} \overline{f}^n} \quad \textbf{(3.27)}$$

where α and β are adjustable coefficients. In addition, BMM method does not require *{yn}* to be sorted.

4. INTERVAL TYPE-2 FUZZY LOGIC SYSTEM TOOLBOX

T2-FLSs have been used in many areas recently. However, designing and implementing an IT2-FLS is more complex and needs more calculations than T1-FLS. In addition, designing an IT2-FLS needs some different steps like type-reduction and defuzzification. In this thesis, a Matlab/Simulink toolbox is proposed to design an IT2-FLS from beginning to simulation. This IT2-FLSs toolbox is implemented by reusing the Matlab® commercial Fuzzy Logic Toolbox and adding new functions for TR operations, improving the toolbox user interfaces, creating Simulink library, connecting the design of the toolbox to Simulink,.

The IT2FLS toolbox design starts by creating a structure in MATLAB. This structure is named as interval type-2 fuzzy inference structure (IT2-FIS). The IT2-FIS structure includes all information of the user’s design. In addition, when new MFs added or current MFs modified or some rules added, actually, at each action this it2fis structure is updated by the IT2FLS toolbox automatically. An example overview of the it2fis is given in the Figure 4.1.

The implementation of the IT2-FLSs Toolbox User Interface is similar to the Graphical User Interface (GUI) used for Type-1 FLS in the Matlab® Fuzzy Logic Toolbox. Thus, adapting to the toolbox is easy for the user familiar to the Matlab® Fuzzy Logic Toolbox.

The IT2-FLS Toolbox consists of four main User Interfaces (UIs). All pages are docked in one tab. So it easy to manage the pages for user. When the toolbox started the firstly the main editor page is opened. It is possible to add input variables or output variables to the current IT2-FLS design from this page. In addition, existing designs can be loaded or the new design can be saved via this page. TR method is chosen in this page and it is possible to open the other pages from the Main Editor. One of the other pages is Membership Edit Page. From this page, UMFs and LMFs are designed. Defining rules for the IT2-FLS design is possible in the page Rule Editor. The last page is to see the surface of the design. The all pages can be seen in the Figure 4.2.

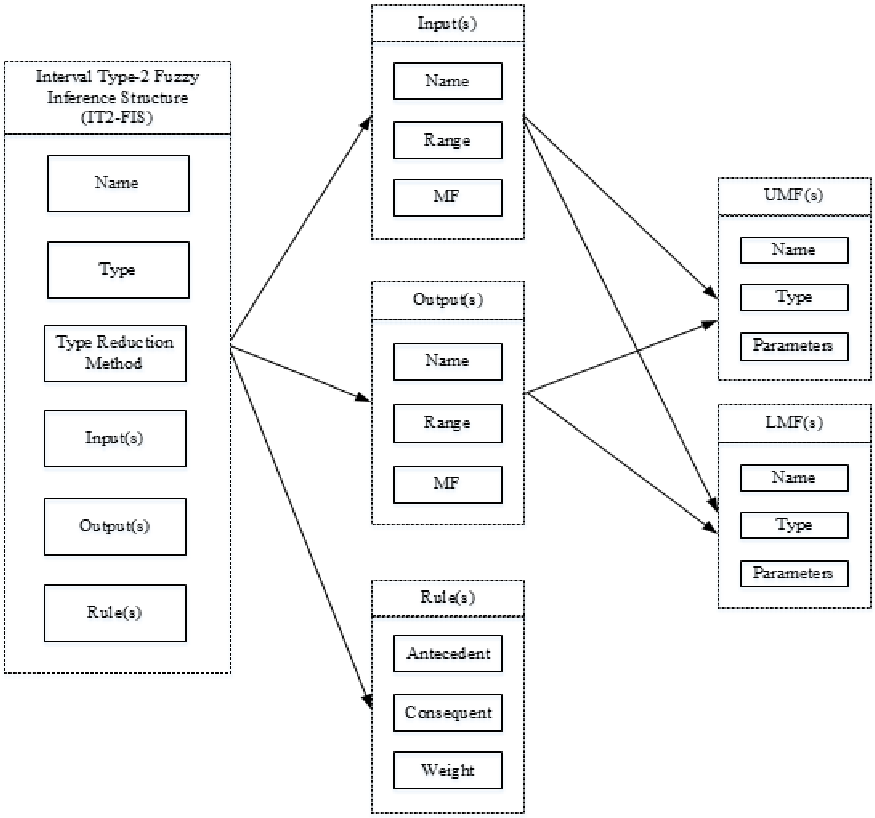


Figure 4.1 : The interval type-2 structure of the IT2-FLSs toolbox in Matlab workspace.

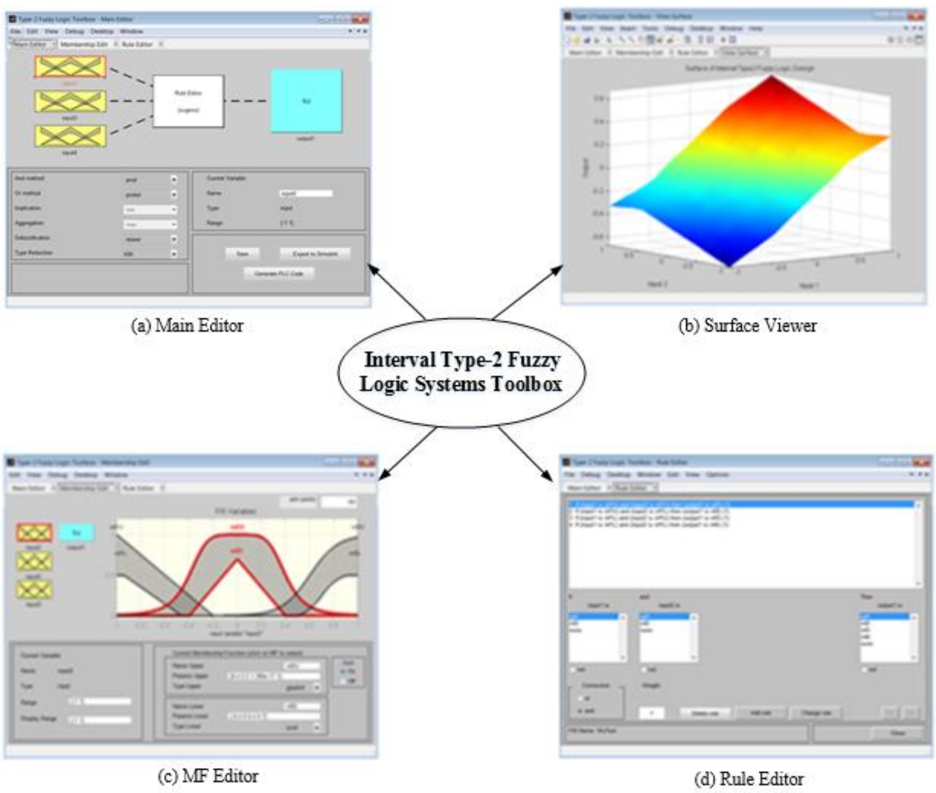


Figure 4.2 : User interfaces of the IT2-FLSs toolbox.

4.1 Main Editor

Main Editor page is the entry point of the IT2-FLS Toolbox. When the IT2-FLS Toolbox started, main editor is opened first. The other pages can be accessed from the Main Editor. An overview of the main editor is given in the Figure 4.3.

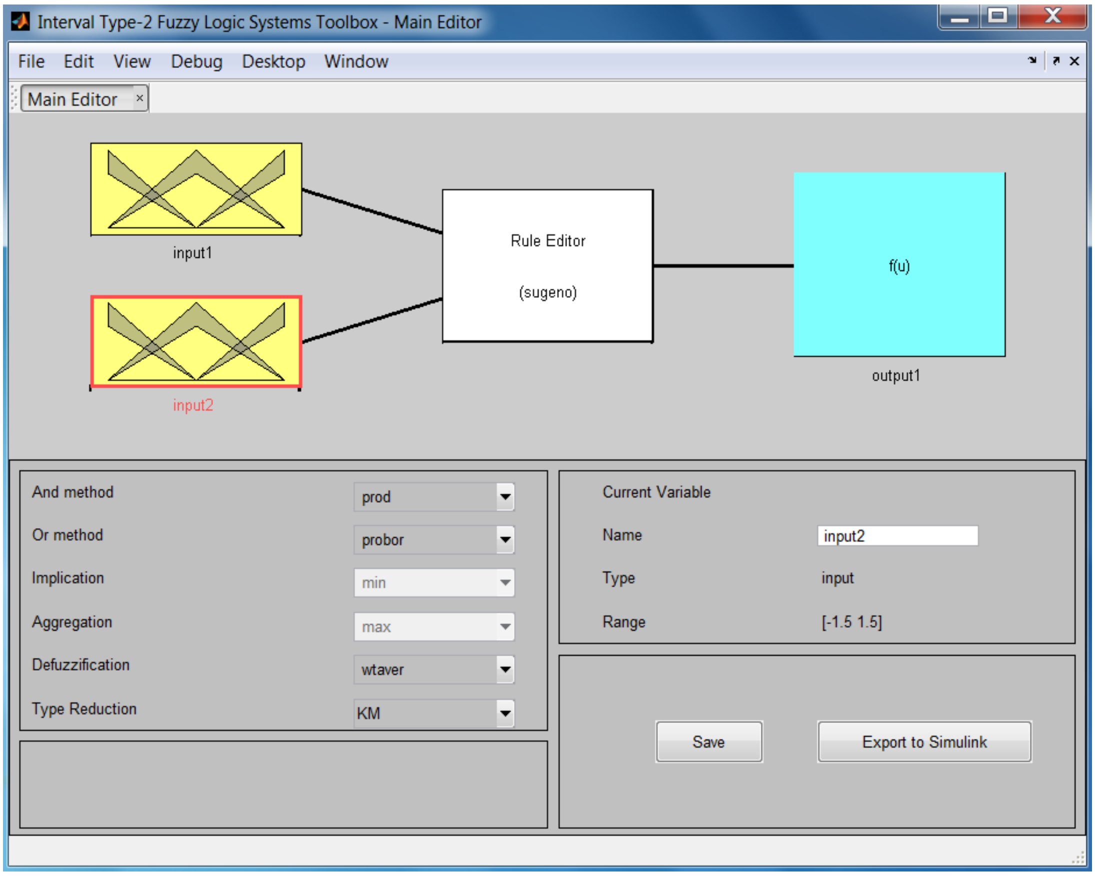


Figure 4.3 : Main editor of the IT2-FLS toolbox.

In the Main Editor, there are some menus in the top of the screen. The IT2-FLS toolbox saves the designs to a file with extension 't2fis' or to Matlab workspace as a structure. Saving and loading operations are available under the file tab. In addition, it is possible to open a new toolbox from the file menu.

Adding inputs and outputs is possible from the edit menu. Default is one input and one output. However, users can add more inputs and outputs as desired by using the edit menu in the main editor. In addition, naming the inputs and outputs are possible by clicking the desired input or output.

In addition, another opportunity of IT2-FLS toolbox is to export the current design to Matlab/Simulink automatically. There is button in the main editor for exporting. This topic is explained in detail in the section 5.

TR is one of the important operation for the IT2-FLS. The most common TR methods is explained in the subsection 3.4 detailed. The IT2-FLS toolbox includes the codes of the following TR methods:

- KM (Karnik-Mendel Algorithm)
- EKM (Enhanced Karnik-Mendel Algortihm)
- IASC (Iterative Algorithm with Stop Condition)
- EIASC (Enhanced Iterative Algorithm with Stop Condition)
- EODS (Enhanced Opposite Direction Searching Algorithm)
- WM (Wu-Mendel Uncertainty Bound Method)
- NT (Nie-Tan Method)
- BMM (Begain-Melek-Mendel Method)
- Custom (User defined functions)

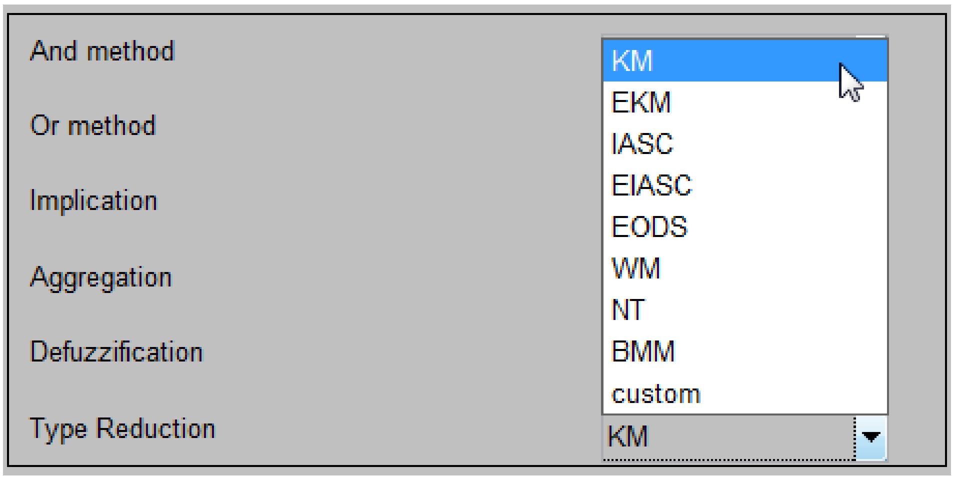


Figure 4.4 : Embedded TR methods to choose.

In addition, it is possible to use a custom function for TR for the users who would like to develop their own TR function. To do this, 'custom' must be selected from the pop-up menu then the desired function can be selected for TR.

4.2 Membership Function Editor

MFs of the IT2-FLS design can be edited from the membership function editor page. An overview of the membership function editor page is given in the Figure 4.5.

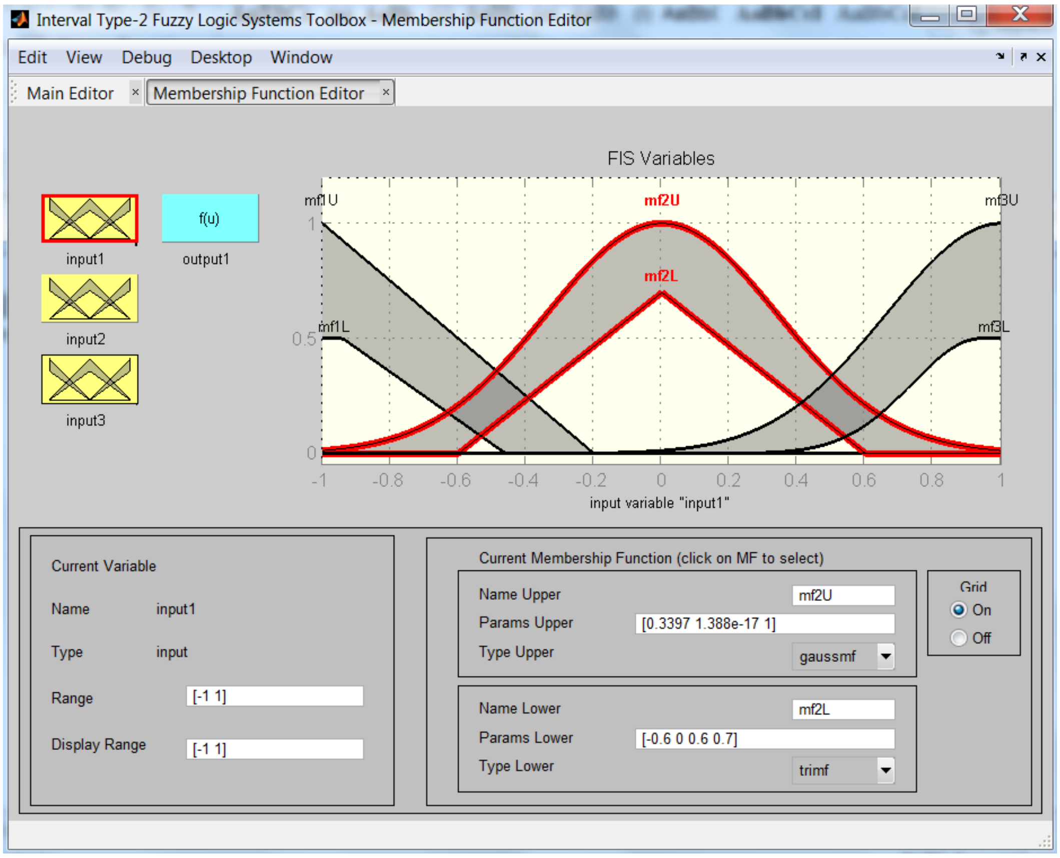


Figure 4.5 : MF editor of the input variables of the IT2-FLSs toolbox.

In this page, basically, range, types and values of the UMFs and LMFs are designed. MFs types are original functions of Matlab Fuzzy Logic Toolbox for UMFs and LMFs separately. The only difference, there is an additional parameters for each type of MFs to define the height. For instance, triangle MF is defined with four parameters as follows:

$$y = \text{trimf}(l,c,r,h) \quad \textbf{(3.26)}$$

First parameter defines the left point (l), second point defines the center point (c) and third parameter defines the right point (r) of the triangular MF. The last parameter (h) defines the height of the triangle MF.

Another advantage of the IT2-FLS toolbox is possibility of designing UMFs and LMFs in either same type or different type. For instance, LMF can be triangle while UMF is Gaussian. The only prerequisite is that all the membership degree values of the LMF have to be smaller than or equal to the UMF’s ones. Otherwise, the toolbox would give a warning message. The warning message is given in the Figure 4.6.

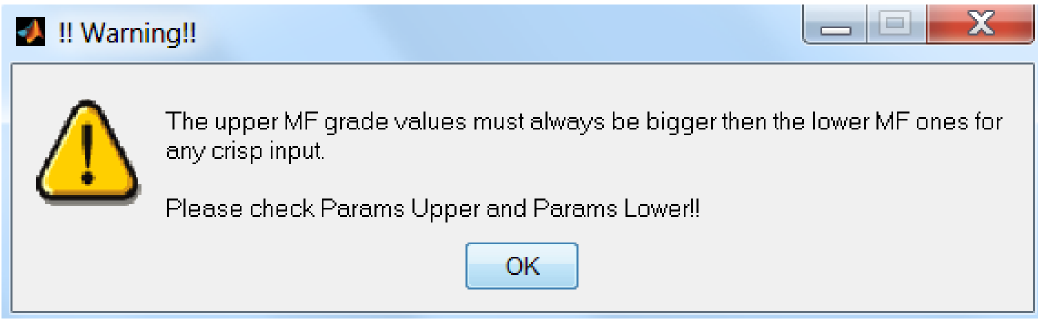


Figure 4.6 : Warning dialog box when the LMF is bigger than the UMF.

Designing output MFs by using the membership function editor page has some opportunities. It is possible to choose type of the output MFs either constant or linear. In addition, it is possible to define parameters either as a crisp or as an interval. A membership function editor page for output variables can be seen in Figure 4.7.

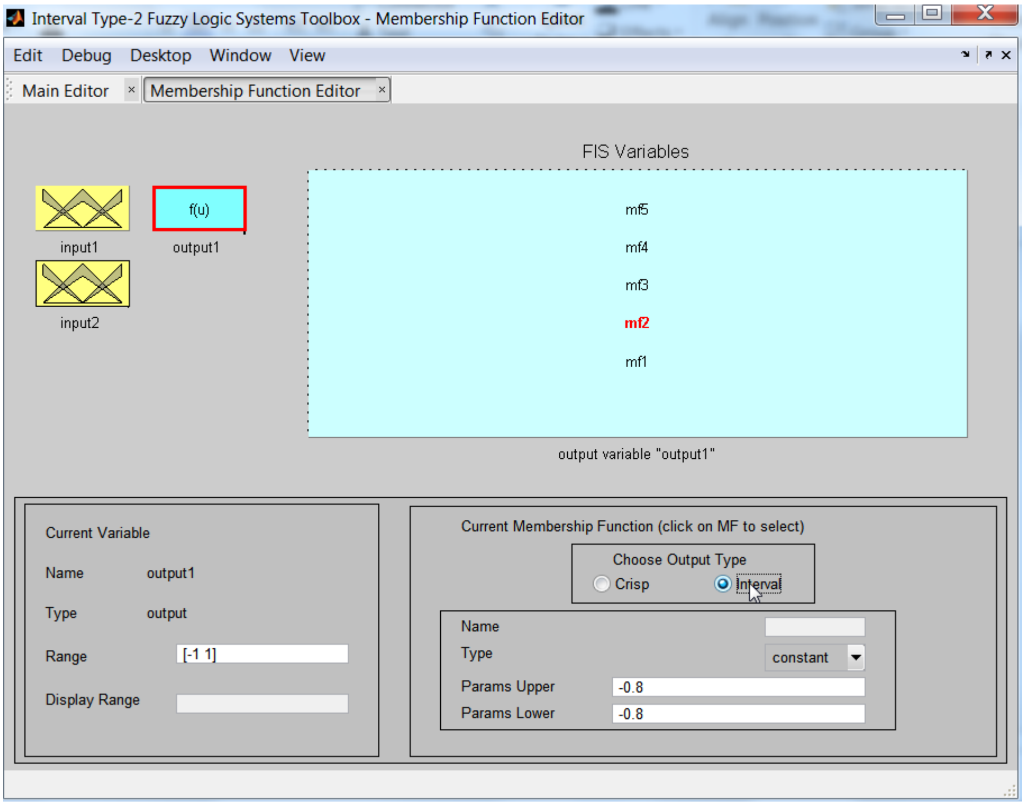


Figure 4.7 : MF editor of the output variables of the IT2-FLSs toolbox.

4.3 Rule Editor

Rule editor page allows users to defining the rules of the type-2 fuzzy logic design. Rule editor is implemented by reusing the Matlab® commercial Fuzzy Logic Toolbox. An overview of rule editor page is in the Figure 4.7.

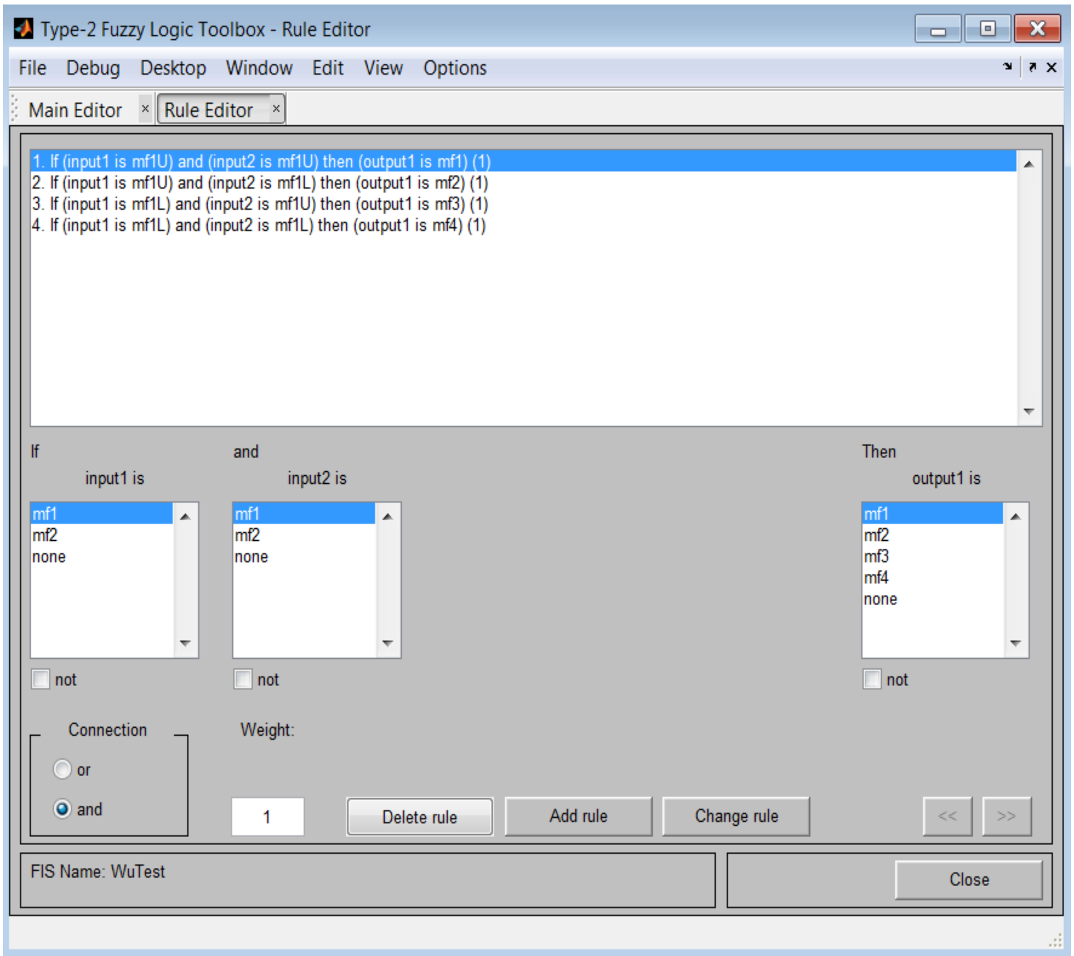


Figure 4.8 : Rule editor of the IT2-FLSs toolbox.

As seen in the Figure 4.7, input and output variables are available in the rule editor. Here, to add a new rule, desired input and output variables are chosen by clicking and then the new rule is added by clicking the add rule button. Then added new rule would be appear in the top. All desired rules added for the IT2-FLS design from the rule editor page. All added new rule information are written to the it2fis structure at the Matlab workspace automatically, and after the rule editor page closed, these information are hold by this structure.

4.4 Surface Viewer

After completing the IT2-FLS design, it is possible to see the surface of the output at each point. To see the surface, surface button from the view menu in the main editor is clicked. It is possible to edit or save the surface opened as desired. An example surface is given in the Figure 4.8.

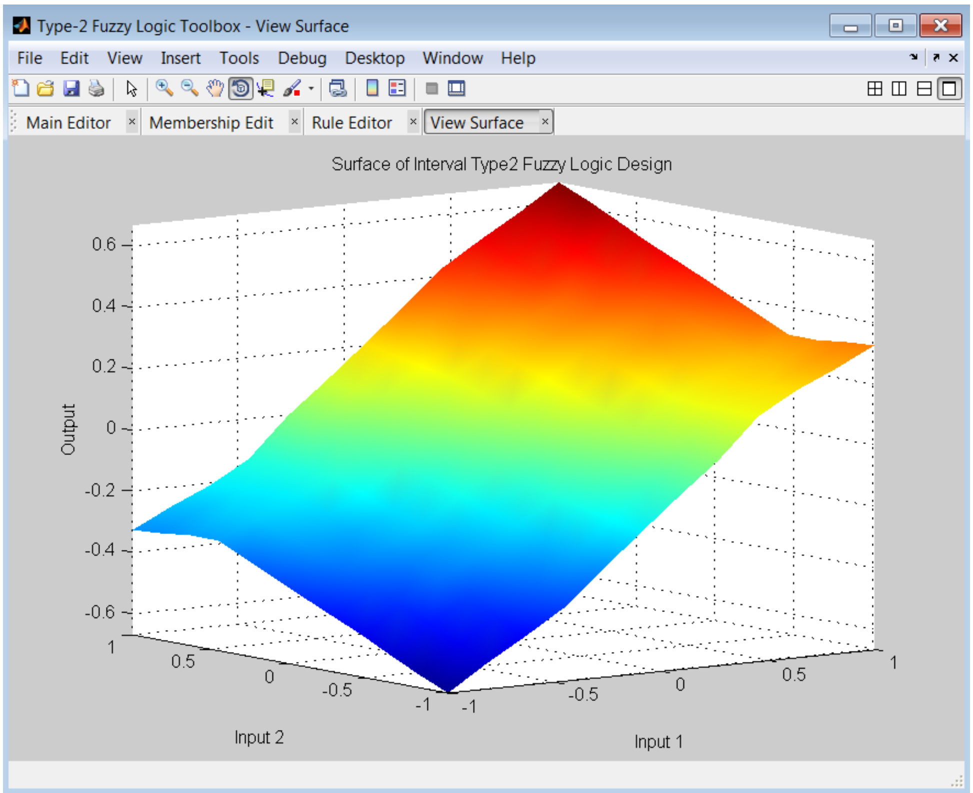


Figure 4.9 : Surface viewer of the IT2-FLSs toolbox.

5. WORKING WITH MATLAB/SIMULINK

The IT2-FLS toolbox is designed to be working with Matlab/Simulink. The one of the main features of the IT2-FLS toolbox is the automatic connection to Simulink. It is possible to use this feature by just clicking the button 'Export to Simulink' in the Main Menu. Then, current design would be exported to Simulink automatically. Firstly, a Simulink library of the IT2-FLS toolbox is created to connect the toolbox to Simulink. And then a function is created to move the current design to Simulink environment. These steps are explained in the next subsections.

5.1 Simulink Library of the IT2-FLS Toolbox

A Simulink library for the IT2-FLS toolbox is created first to ensure the connection between toolbox and Simulink. The IT2-FLSs toolbox library consists of 2 blocks. One of them can be used to embed the design to the block and the other also provides opportunity to choose TR method. The IT2-FLS toolbox is given in the Figure 5.1.

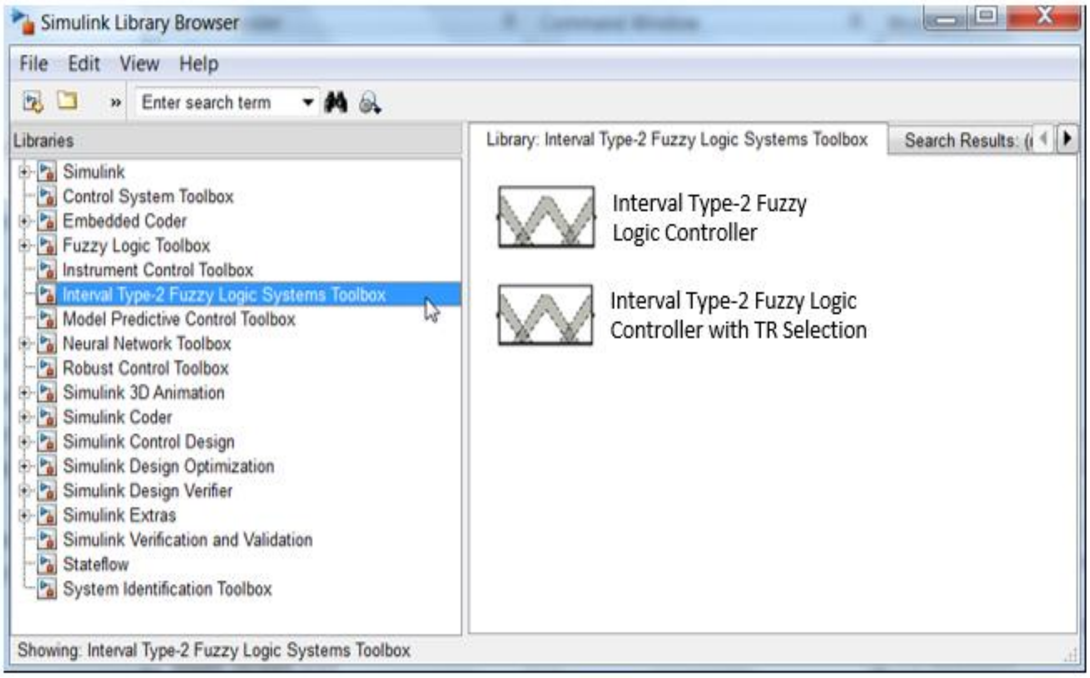


Figure 5.1 : Simulink library of the IT2-FLSs toolbox.

As seen in the Figure 5.1, there are 2 blocks in the IT2-FLS toolbox Simulink library. First block is used to embed the output of the IT2-FLS toolbox. The name of the saved design is written in this block and then inputs and outputs are connected to the block in Simulink. Then, the block is ready for simulation. The second block has an extension of opportunity to choose a TR method. Desired TR method for the current design can be chosen easily by using the second block of the IT2-FLS toolbox library. An overview of the second block is given in the Figure 5.2.

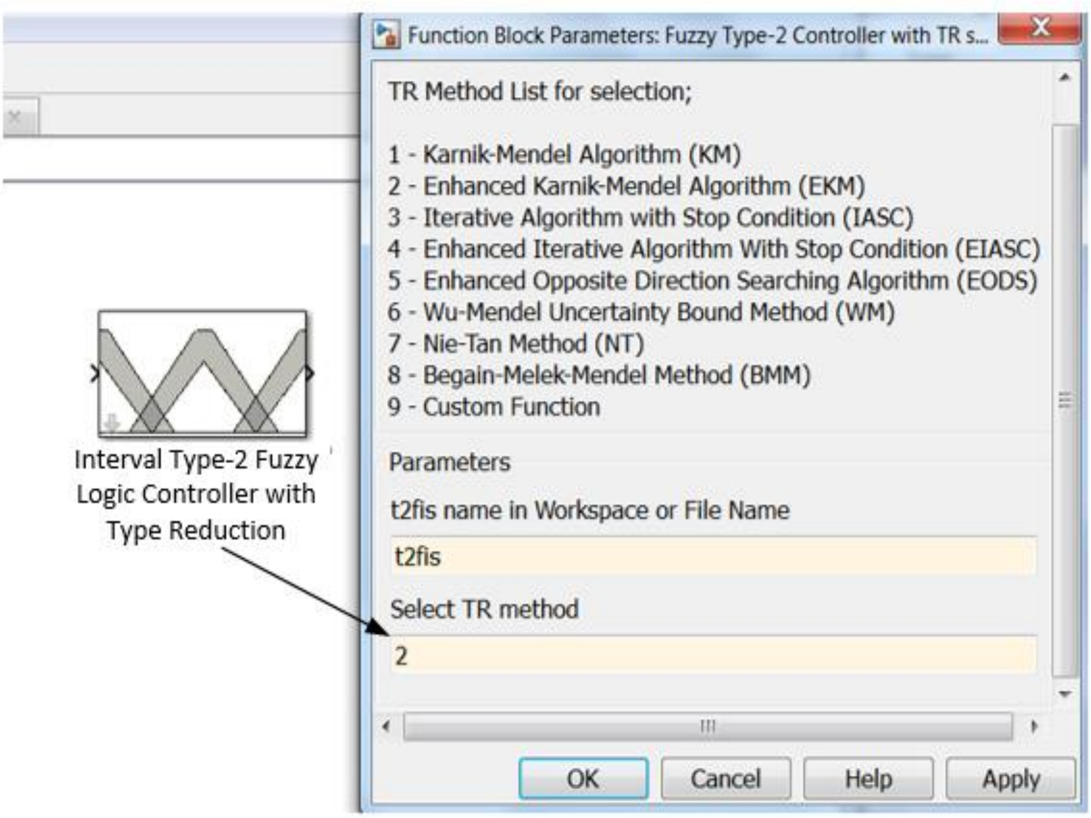


Figure 5.2 : Simulink library block of the IT2-FLSs toolbox for TR selection.

5.2 Exporting the IT2-FLS Design to Simulink

After completing the IT2-FLS design, it is possible to export it to Simulink automatically from the IT2-FLS toolbox. To do this, there is a button named 'Export to Simulink' in the main editor. After finishing the IT2-FLS design, it is possible to export it to Simulink automatically by clicking this button. Then, a Simulink model would be created and opened automatically by the IT2-FLS toolbox. An example Simulink model created by the toolbox is given in the Figure 5.3.

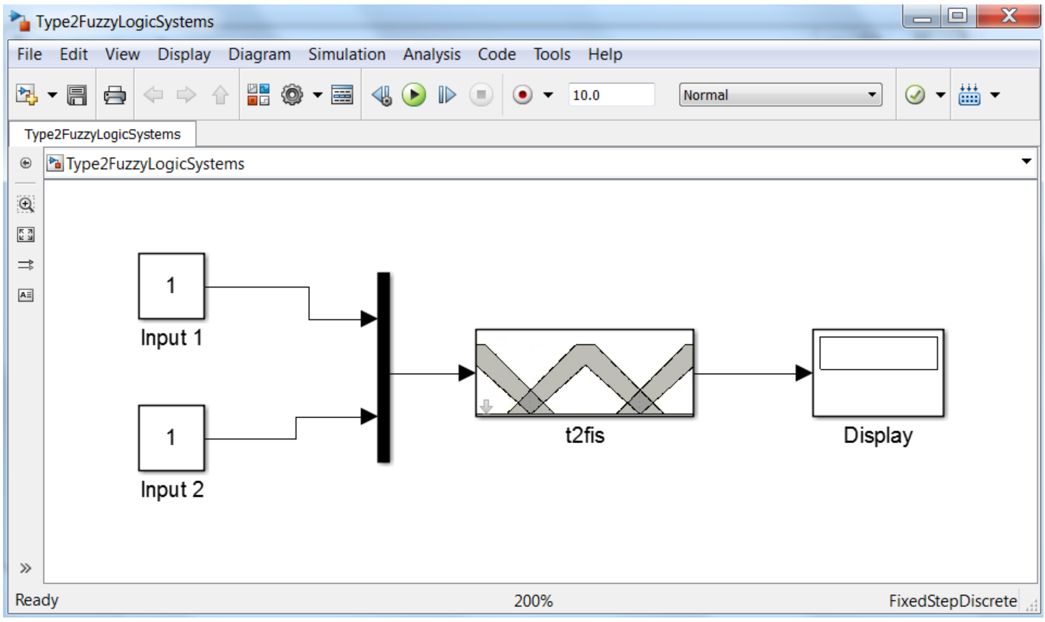


Figure 5.3 : An example Simulink model that is created by the IT2-FLSs toolbox.

6. ILLUSTRATIVE EXAMPLES

In this section, several simulations have been done to examine and compare the different TR methods mentioned in this thesis, and the results are given. For this purpose, firstly, an Interval Type-2 Fuzzy Logic Controller (IT2-FLC) is created by using the IT2-FLS toolbox. Then, simulations are done to compare the control performance of the different TR methods for the same closed loop control system. The IT2-FLC is designed with two input variables and one output variable. The block diagram of the IT2-FLC is given in Figure 6.1.

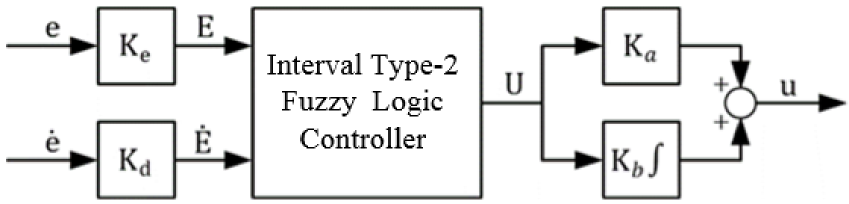


Figure 6.1 : IT2-FPID controller structure.

The first input of the IT2-FLC is the error signal the second input of the IT2-FLC is the derivative of the error signal. The MFs of the input variables are chosen same and as triangle. The height of the LMFs are found by using the genetic algorithm. The designed MFs for the Interval Type-2 Fuzzy PID (IT2-FPID) controller are given in the Figure 6.2.

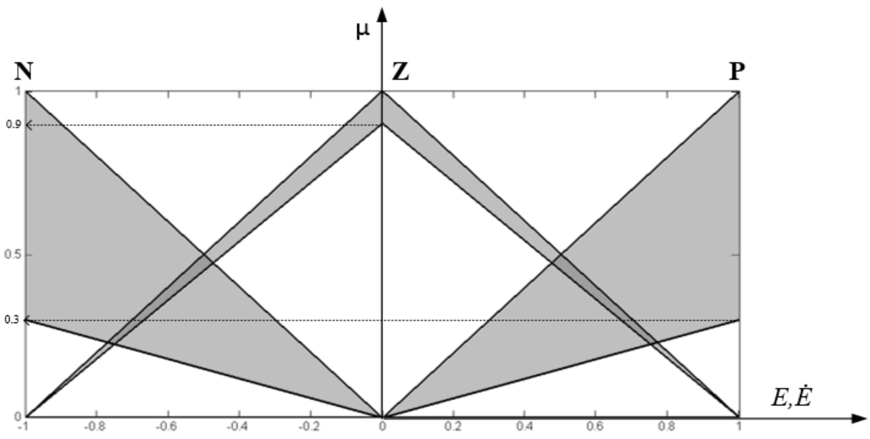
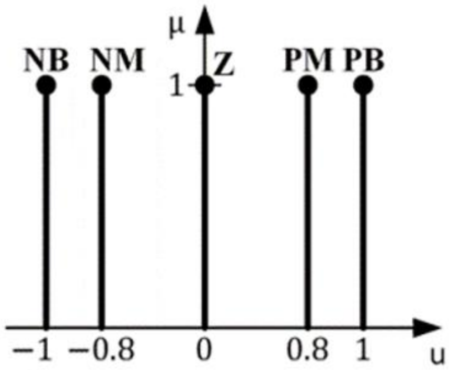


Figure 6.2: The inputs MFs used for the IT2-FPID controller.

The output MFs and the rule base of the IT2-FPID is given in Figure 6.3.

E \ $\dot{E}$	**N**	**Z**	**P**
N	NB	NM	Z
Z	NM	Z	PM
P	Z	PM	PB

(a)



(b)

Figure 6.3: (a) The fuzzy rule base, (b) the consequent MFs used for the IT2-FPID controllers.

The IT2-FPID MFs of the input variables can be seen in Figure 6.2.

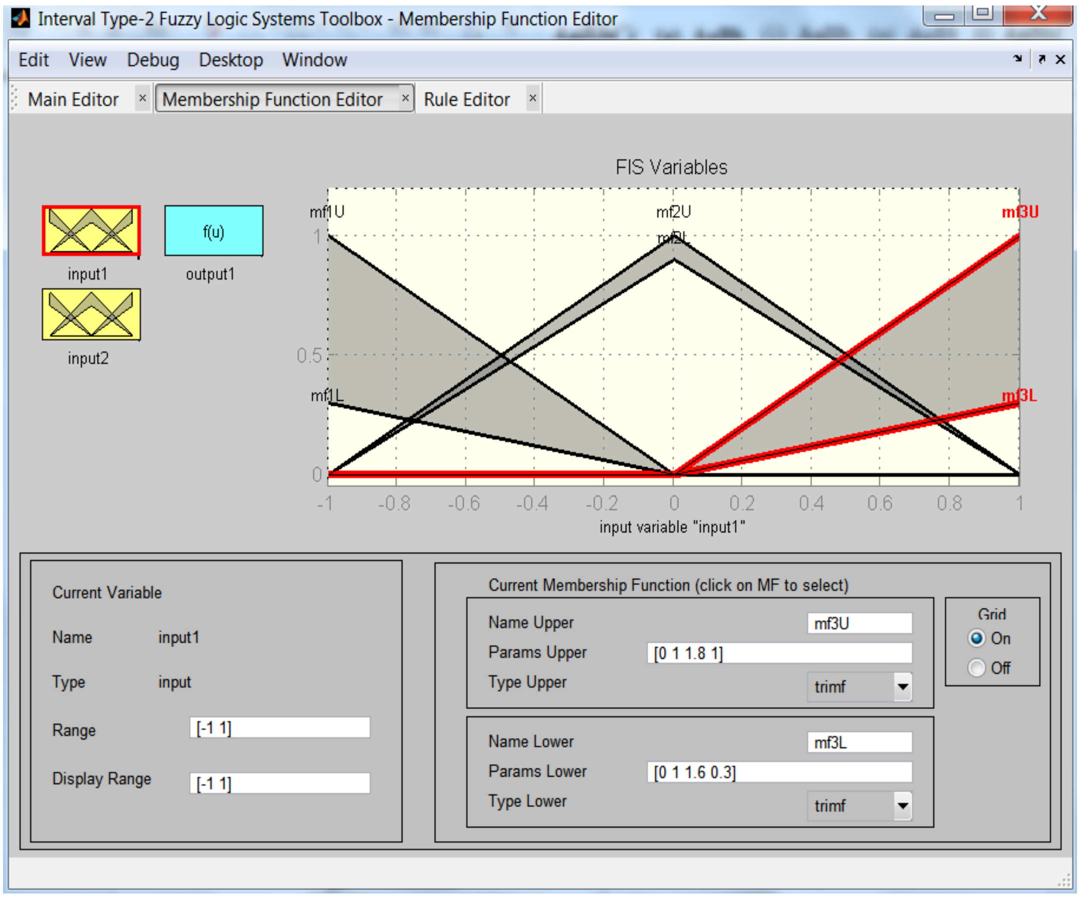


Figure 6.2 : Input and output MFs of the IT2-FPID controller.

Then, rules are defined for the IT2-FPID as seen in Figure 6.3.

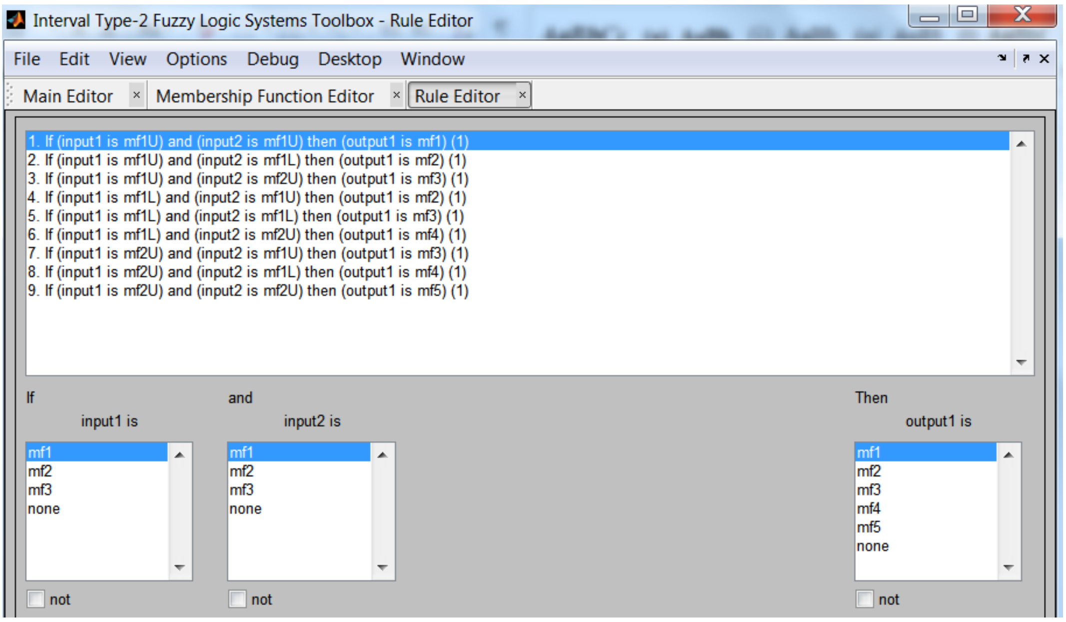


Figure 6.3 : Rules of the IT2-FPID controller.

After defining the MFs of the input and output variables and defining the rules, firstly, the surfaces are viewed for different TR methods. The control surface would be the same for Karnik-Mendel algorithm and its enhancements, so, just four different surfaces are viewed. The IT2-FPID surface for the KM algorithm is given in Figure 6.4.

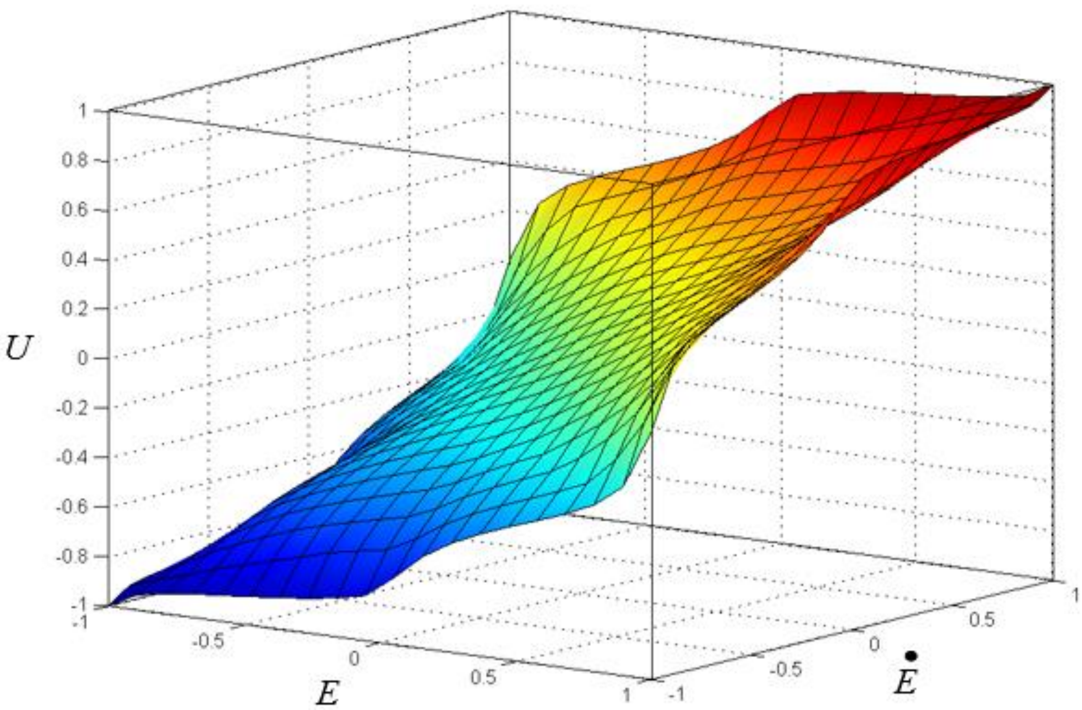


Figure 6.4 : Surface of the IT2-FPID controller with KM algorithm.

The surfaces for the WM method, NT method and BMM method are given in the Figure 6.5, Figure 6.6 and Figure 6.7, respectively.

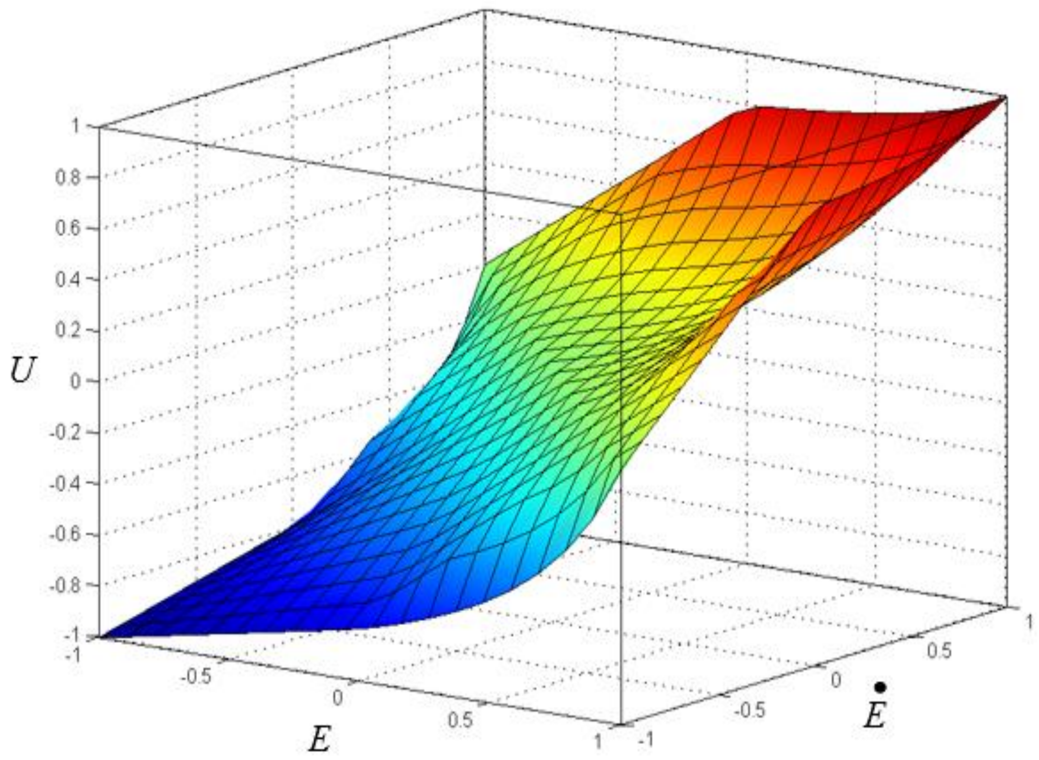


Figure 6.5 : Surface of the IT2-FPID controller with WM algorithm.

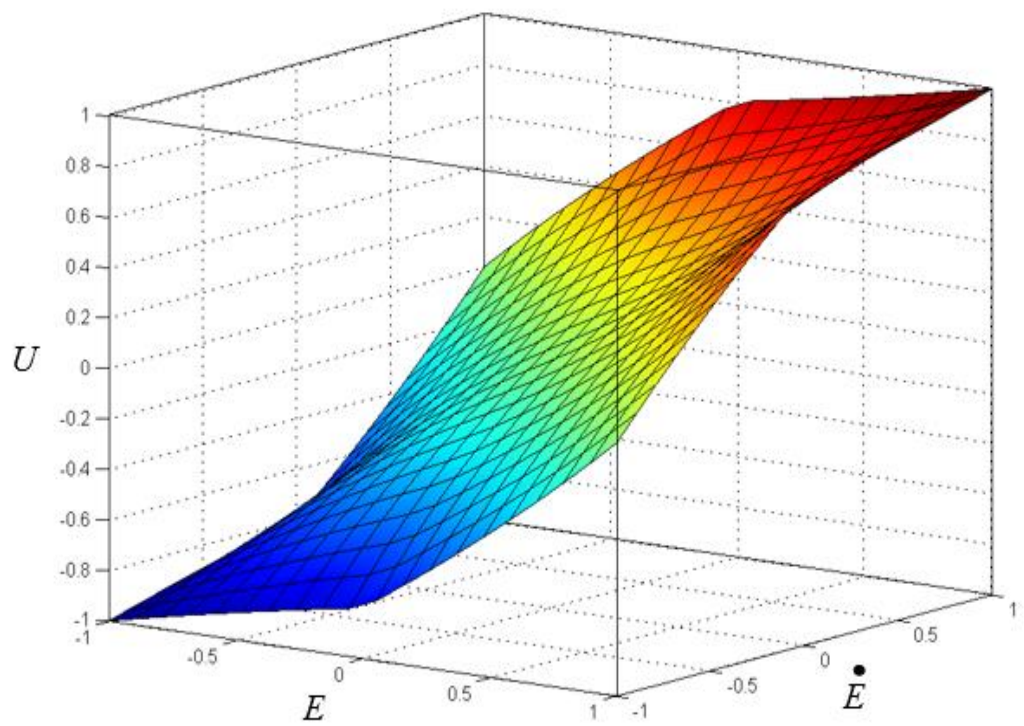


Figure 6.6 : Surface of the IT2-FPID controller with NT algorithm.

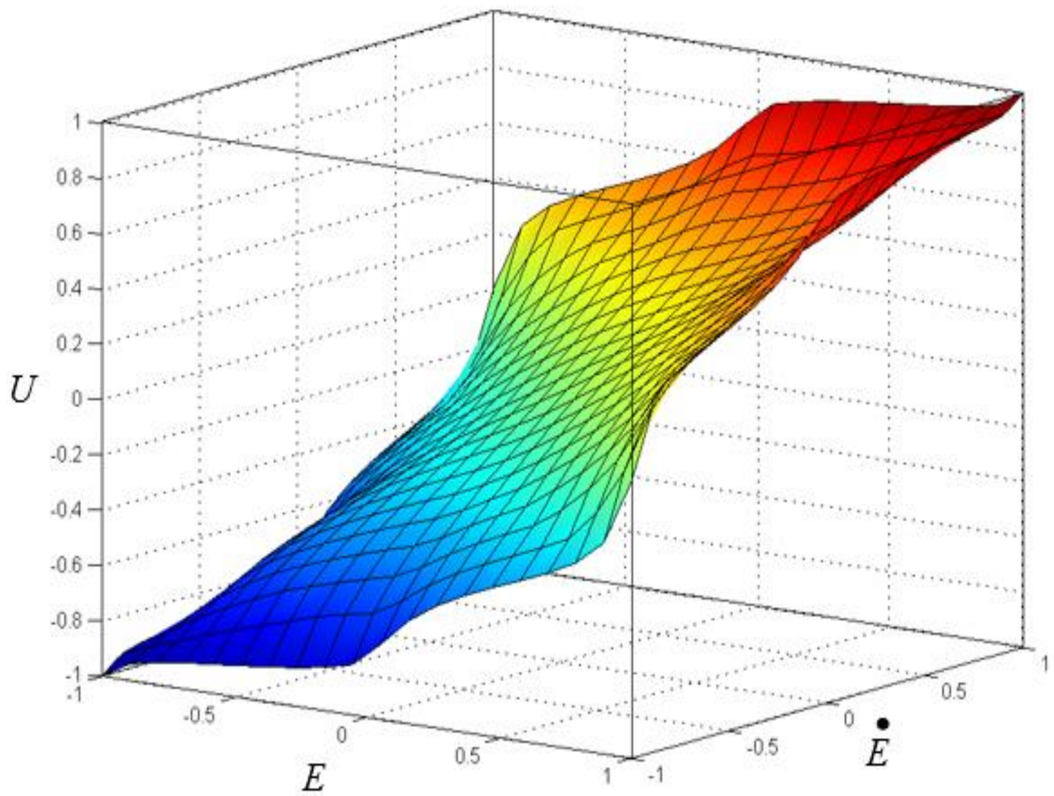


Figure 6.7 : Surface of the IT2-FPID controller with BMM algorithm.

As can be seen in the figures above, there are small differences in the surfaces as expected. The closed-form methods find approximation results and it is the reason why there are some differences between the same IT2-FPID controller with different TR methods.

As the next step, the process to be controlled is chosen as a First Order Plus Dead Time (FOPDT) system. The wide use of FOPDT is due both to its simplicity as well as its ability to capture the essential dynamics of several industrial processes. The transfer function of the FOPDT is as follows:

$$G(s) = \frac{K}{Ts+1}e^{-Ls} \qquad \textbf{(6.1)}$$

where K is the process gain, T is the time constant and finally L is the time delay. In the simulations, four different TR methods are used. In order to compare the performance of transient responses of the different TR methods, overshoot, settling time and these two following performance measures are considered:

i) Integral Absolute Error (IAE) which is defined as

$$\text{IAE} = \int_0^{\infty} |r(t) - y(t)| \, dt \tag{6.2}$$

ii) Integral Time Absolute Error (ITAE) which is defined as

$$\text{ITAE} = \int_0^{\infty} t|r(t) - y(t)| \, dt \tag{6.3}$$

The Simulink model that was created for the simulations is given in Figure 6.8.

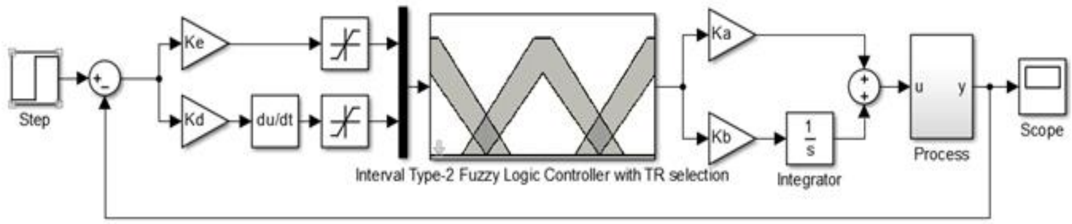


Figure 6.8 : Simulink model that is created for the simulations.

As the first step, the nominal process parameters are set to K=1, T=1 and L=0.2. The sampling time of the simulation is set to 0.05 s. Since the unit step changes are studied for the simulations K_e is set to 1, then the other scaling factors are determined as $K_d = 0.5141$, $K_a = 0.077$, $K_b = 7.336$ after optimization. The step responses of the closed loop control system with different TR methods are illustrated in Figure 6.9 and Figure 6.10.

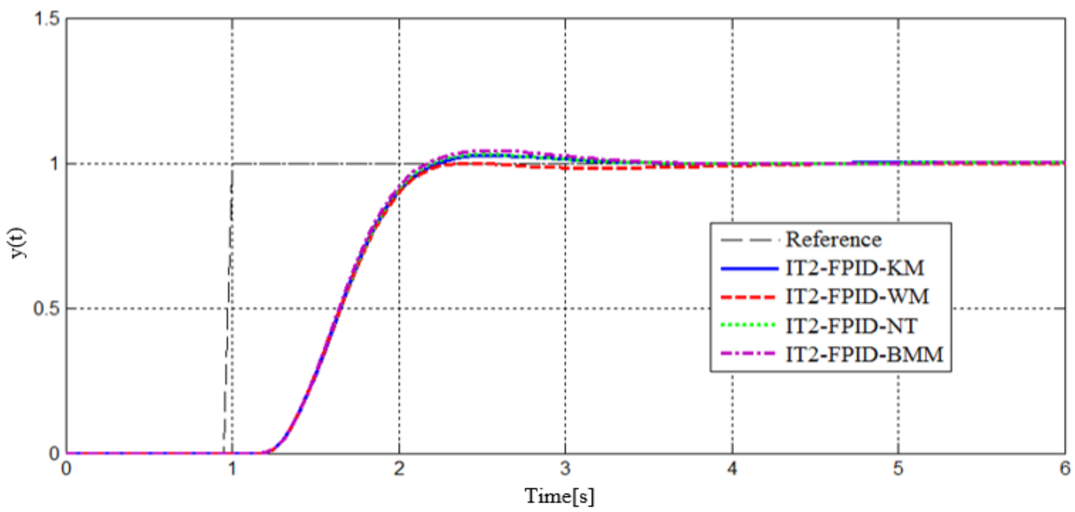


Figure 6.9 : The step responses of the closed loop system for various TR methods for the nominal process.

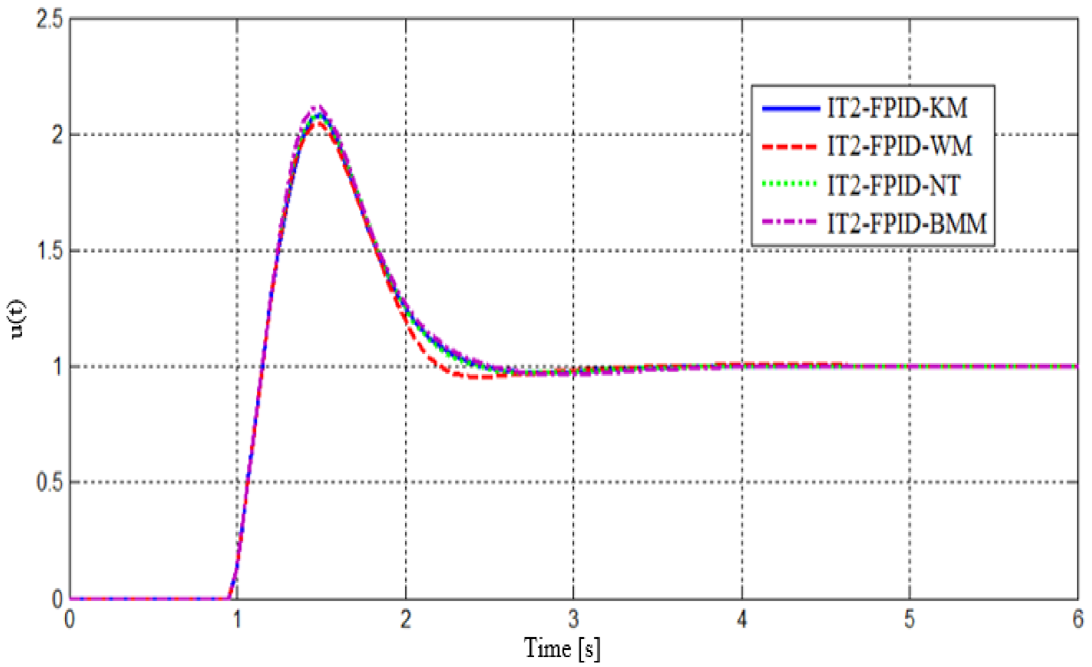


Figure 6.10 : The control signals of the IT2-FPID for different TR methods for the nominal process.

In order to examine the performance of the different TR methods, the system parameters are perturbed. The step responses of the closed loop control system with different TR methods for the first perturbed process, which has the parameters as K=1.3, T=1.9, L=0.4, is given in Figure 6.11 and Figure 6.12.

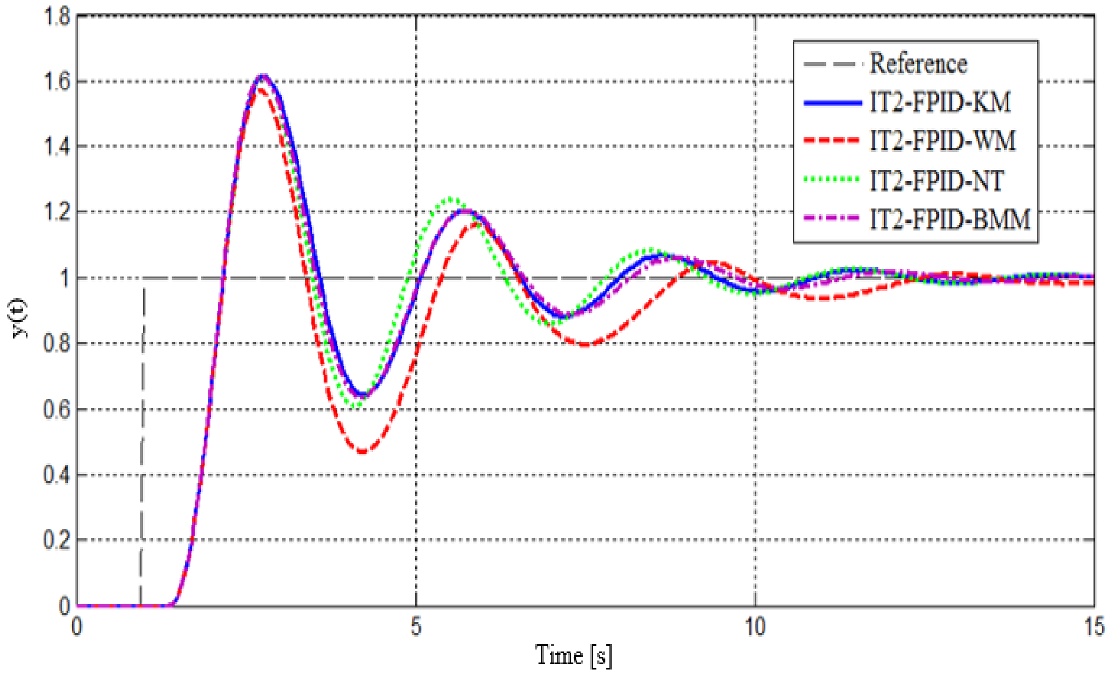


Figure 6.11 : The step responses of the closed loop system for various TR methods for the perturbed process (K=1.3, T=1.9, L=0.4).

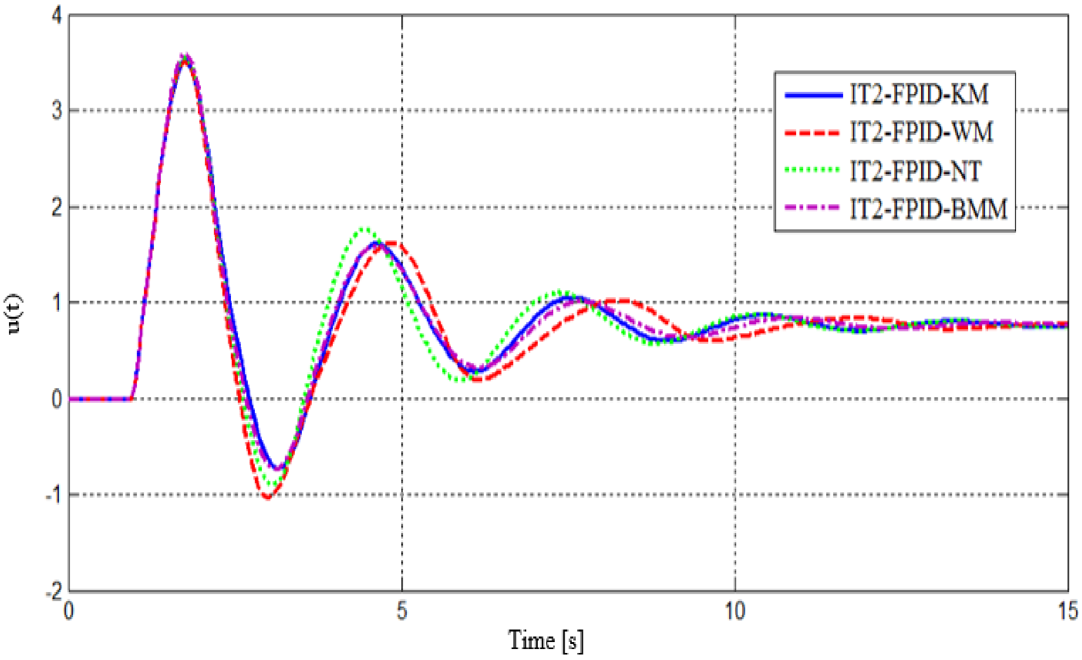


Figure 6.12 : The control signals of the IT2-FPID for different TR methods for the perturbed process (K=1.3, T=1.9, L=0.4).

Lastly, the step responses of the closed loop control system with different TR methods for the second perturbed process, which has the parameters as K=1.1, T=1.3, L=0.45, is given in Figure 6.8 and Figure 6.9.

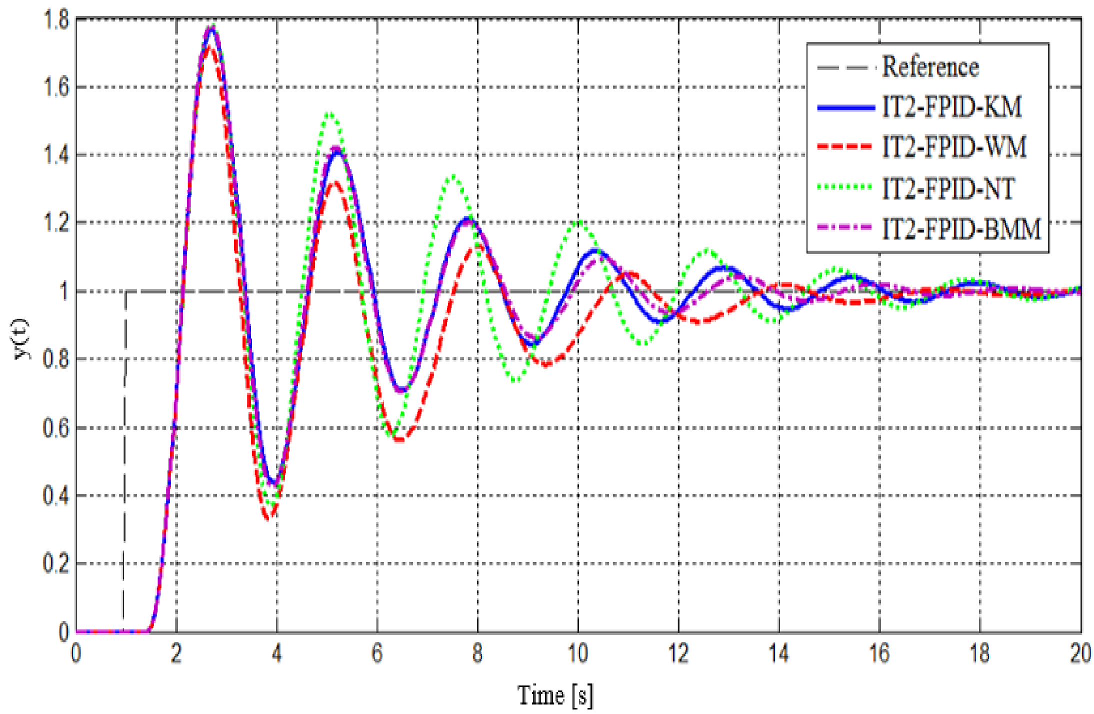


Figure 6.13 : The step responses of the closed loop system for various TR methods for the perturbed process (K=1.1, T=1.3, L=0.45).

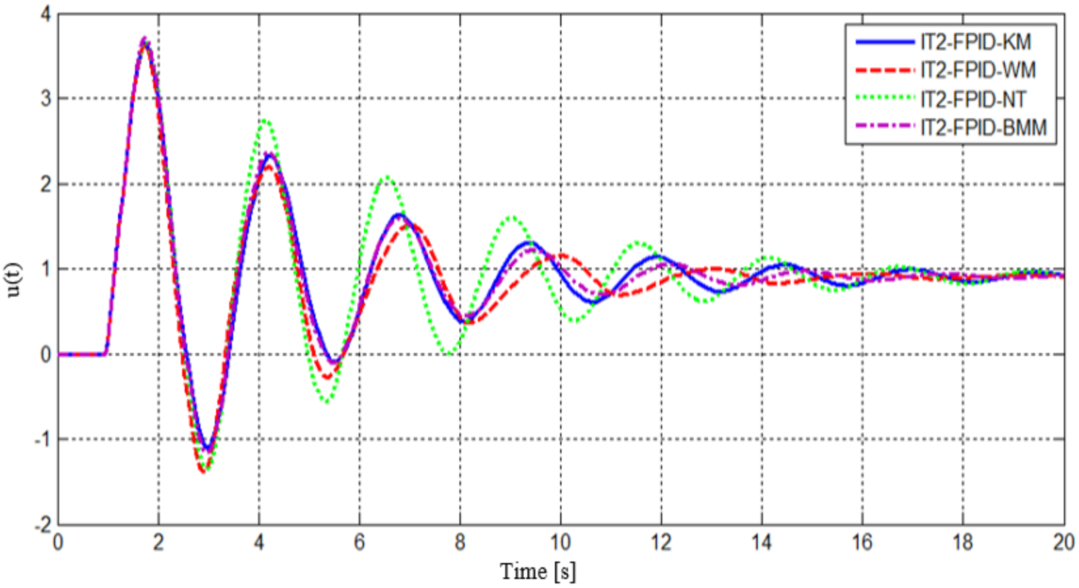


Figure 6.14 : The control signals of the IT2-FPID for different TR methods for the second perturbed process (K=1.1, T=1.3, L=0.45).

The performance results of the closed loop control system with different TR methods are tabulated in Table 6.1. The settling time, overshoot, IAE and ITAE values for each processes are given in this table.

Table 6.1 : Simulation comparison using different performance measures.

Process	TR METHOD	SETTLING TIME (s)	OVERSHOOT (%)	IAE	ITAE
Nominal Process K=1, T=1, L=0.2	KM	1.15	2.600	0.697	13.938
	WM	1.15	0.000	0.703	14.059
	NT	1.15	2.901	0.692	13.840
	BMM	1.15	4.294	0.708	14.035
First Perturbed Process K=1.3, T=1.9, L=0.4	KM	8.05	61.155	2.237	44.737
	WM	10.5	57.027	2.594	51.875
	NT	7.95	61.470	2.279	45.578
	BMM	8.20	61.676	2.219	44.379
Second Perturbed Process K=1.1, T=1.3, L=0.45	KM	13.30	75.909	3.168	63.358
	WM	12.10	71.448	3.280	65.594
	NT	14.45	76.314	3.750	74.995
	BMM	11.15	79.241	3.069	61.389

As seen in Table 6.1, for the nominal process, settling time of the each TR method is almost the same, the WM method has showed the best performance for the overshoot. For the first perturbed process, settling time of the NT method is the shortest, but the best performance for the overshoot has been shown by the WM method. Considering the IAE and ITAR criteria, BMM method has showed the most successfully performance. For the last perturbed system, KM algorithm is the worst method when considering the settling time and overshoot, BMM method has showed the best performance considering the IAE, ITAE and settling time, however, this method is the worst one for the overshoot. In addition to these analyses, the disturbance-rejection performance are analysed for the different TR methods. For this purpose, a disturbance is given to the IT2-FPID output at the 40th second with 0.3 amplitude. The disturbance rejection performance, and control signals for the different TR methods are illustrated in Figure 6.15, Figure 6.16, Figure 6.17, Figure 6.18, Figure 6.19 and Figure 6.20.

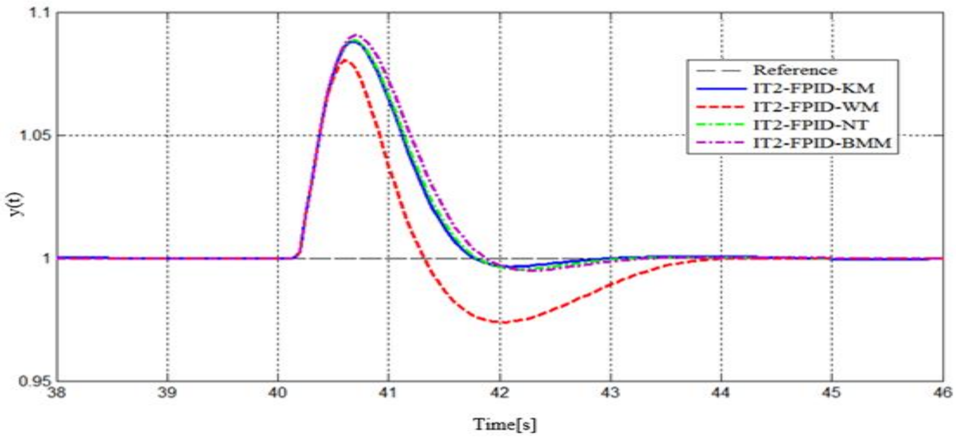


Figure 6.15 : Disturbance rejection performances of the IT2-FPID for different TR methods for the nominal process.

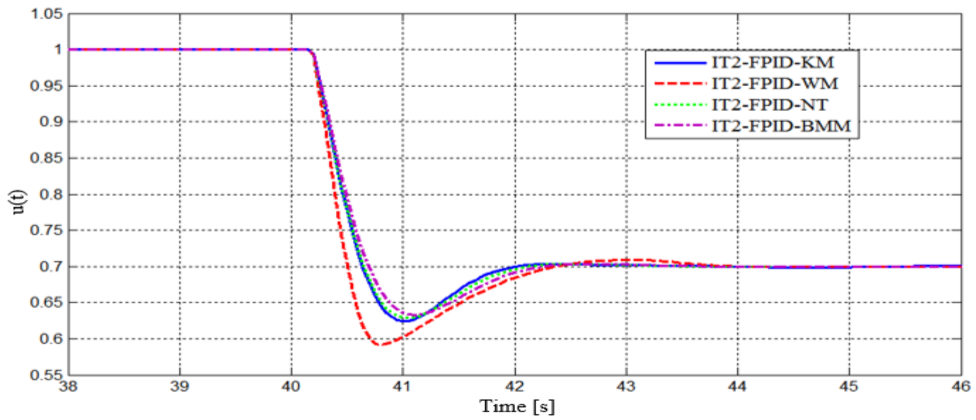


Figure 6.16 : The disturbance rejection control signals of the IT2-FPID for different TR methods for the nominal process.

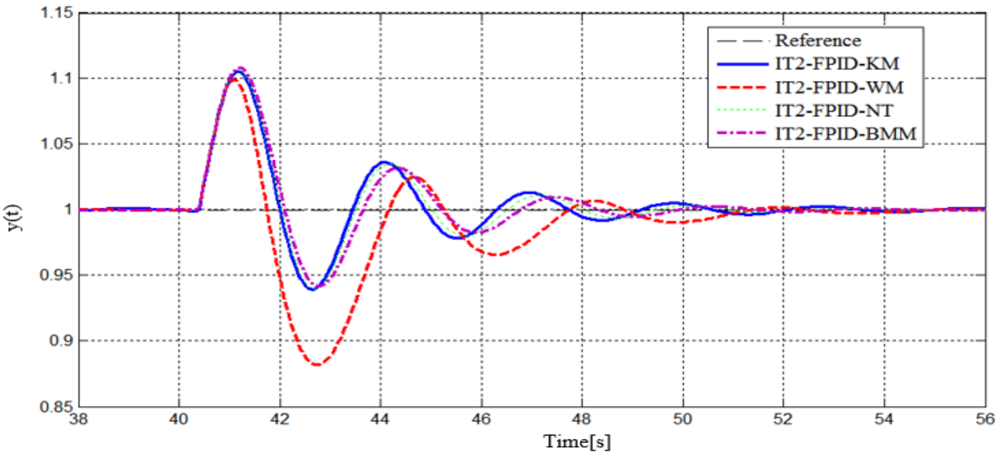


Figure 6.17 : Disturbance rejection performances of the IT2-FPID for different TR methods for the perturbed process (K=1.3, T=1.9, L=0.4).

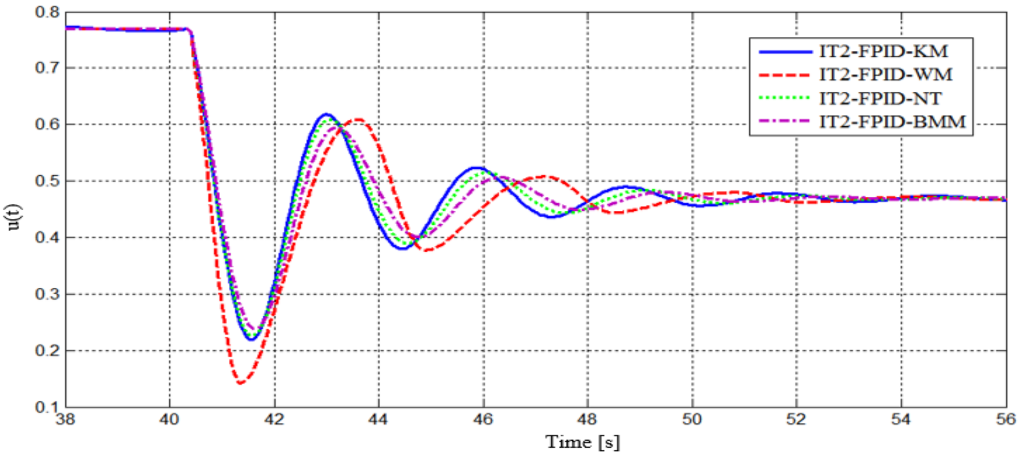


Figure 6.18 : The disturbance rejection control signals of the IT2-FPID for different TR methods for the perturbed process (K=1.3, T=1.9, L=0.4).

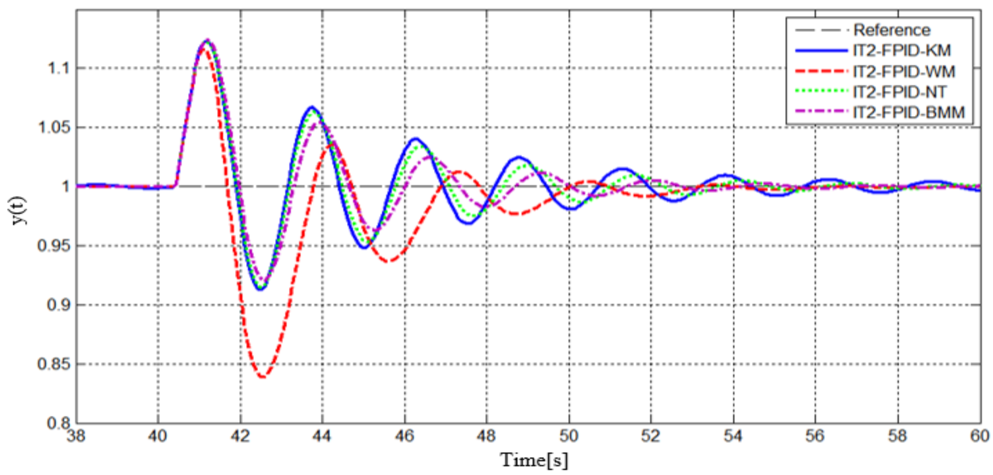


Figure 6.19 : Disturbance rejection performances of the IT2-FPID for different TR methods for the perturbed process (K=1.1, T=1.3, L=0.45).

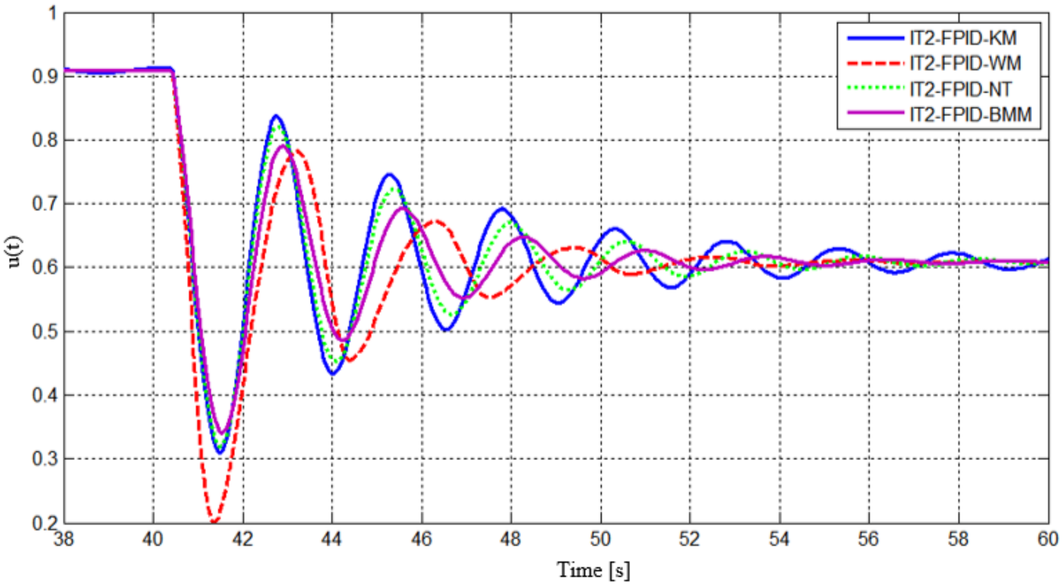


Figure 6.20 : The disturbance rejection control signals of the IT2-FPID for different TR methods for the perturbed process (K=1.1, T=1.3, L=0.45).

The other comparison has been made to compare the computational time of the different TR methods. For this purposes, the each function that has been created for the each TR method is run 2500 times and the maximum, minimum and average values have been measured and given in Table 6.2. As seen in this table, the KM algorithm takes the longest time to reduce the type-2 sets to type-1 sets. As expected, the enhancement methods of the KM algorithm are able to find the solution in shorter time. Also, the closed-form methods find the solution faster. It is expected, because these types of methods find the closed form solutions without need making any iteration.

Table 6.2 : Comparison of computational time of the different TR methods.

TR Method	MEAN (s)	MIN (s)	MAX (s)
KM	0.0058	0.0052	0.0116
EKM	0.0056	0.0053	0.0099
IASC	0.0055	0.0052	0.0101
EIASC	0.0055	0.0052	0.0113
EODS	0.0053	0.0051	0.0122
WM	0.0048	0.0045	0.0092
NT	0.0046	0.0044	0.0087
BMM	0.0049	0.0046	0.0096

7. CONCLUSION

In this Master thesis, type-1 fuzzy sets and systems are explained first. After that, T2-FSs and T2-FLSs have been introduced and the advantages of the IT2-FSs are highlighted. The most known TR methods have been explained more in detail.
A Matlab/Simulink toolbox is designed and proposed for the IT2-FLSs. The design of the IT2-FLS toolbox has been started to creating a structure in the Matlab workspace to save the all information of the IT2-FLS. The TR methods, input and output variable types, the MF parameters, the rule information and the other information are saved to this structure after each action. In addition, defining MFs of the input and output variables can be implemented easily by using the user interfaces of the toolbox. To provide this, the all MFs of the Matlab fuzzy logic toolbox have been reused by adding an additional input to the end. The last input provides the opportunity to define the height of the LMFs. In addition, some functions have been created to define the MFs of the output variables in either crisp or interval. The other opportunity of the toolbox is possibility to view the surface of the current interval type-2 fuzzy logic design. The one of the other important feature of the interval type-2 fuzzy logic toolbox is possibility to work with Simulink. To work with Simulink, firstly a new Simulink library has been created for the interval type-2 fuzzy logic toolbox. The created Simulink library has two blocks. The first one in used to simulate the designed IT2-FLC and the second one gives the user an opportunity to choose desired TR method. In addition, it is possible to export the IT2-FLS design from the toolbox to Simulink automatically by clicking a button. Then, it creates a Simulink model with current design and the user can start the simulations easily. In the last section of this thesis, some comparisons have been made between the TR methods that are explained in this thesis. For this purpose, an IT2-FLC has been designed first, and then it has been exported to the Simulink. In Simulink, a closed loop control system has been created with the IT2-FLC and a FOPDT system. Then, the control performances of the different TR methods have been analyzed. In addition, the parameters of the FOPDT system has perturbed twice and the simulations repeated. The simulation results showed that the control performances have been changed for the same IT2-FLC with

different TR methods. In addition, computational time of the each TR methods has been measure. As expected, the KM algorithm reduce the type the longest period. The enhancement methods of the KM algorithm find the solution in shorter time. In addition, the closed-form TR methods find approximation result faster.

In summary, IT2-FLSs are quite complicated systems. There are many steps to implement an interval type-fuzzy logic system from the first description phase to the implementation phase. In this thesis, a Matlab/Simulink toolbox has been designed and introduced for the IT2-FLSs to cover the all phases from beginning to the end. In addition, the performance of the different TR methods have been compared by using the developed toolbox.

REFERENCES

Babuska, R., Oosterhoff, J., Oudshoorn, A., Bruijn, P. M. (2002). Fuzzy self-tuning PI control of pH in fermentation, *Engineering Applications of Artificial Intelligence,* vol. 15, pp. 3–15.

Begian, M. B., Melek, W., Mendel, J. (2008). Stability analysis of type-2 fuzzy systems, *in Proceedings IEEE International Conference on Fuzzy Systems*, pp. 947–953.

Begian, M. B., Melek, W., Mendel, J. (2010). On the stability of interval type-2 TSK fuzzy logic control systems, *IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics*, vol. 40, pp. 798-818.

Castillo, O., Huesca, G., Valdez, F. (2005). Evolutionary computing for optimizing type-2 fuzzy systems in intelligent control of non-linear dynamic plants, *In Proceeding of North American Fuzzy Information* Processing Society, pp. 247–251.

Castillo, O., and Melin, P. (2008). Type-2 Fuzzy Logic Theory and Applications, Berlin, Germany: Springer-Verlag.

Castro, J. R., Castillo, O., Melin, P. (2007). An interval type-2 fuzzy logic toolbox for control applications. *IEEE International Conference on Fuzzy Systems Conference*, pp. 1-6.

Coupland, S., and John, R. (2007). Geometric type-1 and type-2 fuzzy logic Systems, *IEEE Transactions on Fuzzy Systems*, vol. 15, pp. 3-15.

Duran, K., Bernal, H., Melgarejo, M. (2008). Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set. *IEEE Information Processing Society on Fuzzy Systems*, pp. 1-5.

Eksin, İ., Güzelkaya, M., Gürleyen, F. (2001). A new methodology for deriving the rule-base of a fuzzy logic controller with a new internal structure, *Engineering Applications of Artificial Intelligence*, vol. 14, pp.617-628.

Erol, O. K., and Eksin, I. (2006). A new optimization method: big bang big crunch, *Advances in Engineering Software*, vol. 37, pp. 106-111.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization and machine learning, Addison-Wesley Longman Publishing USA.

Hagras, H. (2004). A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, *IEEE Transactions on Fuzzy Systems*, vol.12, pp. 524-539.

Hu, H., Wang, Y., Cai, Y. (2012). Advantages of the Enhanced Opposite Direction Searching Algorithm for Computing the Centroid of An Interval Type-2 Fuzzy Set, *Asian Journal of Control*, vol. 14, pp. 1422-1430.

Huang, T. T., Chung, H. Y., Lin, J. J., (1999). A fuzzy PID controller being like parameter varying PID, *IEEE International Fuzzy Systems Conference Proceeding*, vol. 1, pp. 269–275.

Karasakal, O., Güzelkaya M., Eksin İ., Yeşil, E., Kumbasar T. (2013). Online tuning of fuzzy PID controllers via rule weighting based on normalized acceleration, *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 184-197.

Karnik, N. N., Mendel, J. M., Liang, Q. (1999). Type-2 fuzzy logic systems, *IEEE Transaction on Fuzzy Systems*, vol. 7, pp. 643–658.

Kumbasar, T., Yesil, E., Eksin, İ., Güzelkaya, M. (2008). Inverse fuzzy model control with online adaptation via big bang big crunch Optimization, *The 3rd International Symposium on Communications, Control and Signal Processing*.

Lam H. K., and Seneviratne, L. D. (2008). Stability analysis of interval type-2 fuzzy-model-based control systems, *IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics*, vol. 38, pp. 617-628.

Li, H. X., and Gatland, H. B., (1996). Conventional fuzzy control and its enhancement, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, pp. 791-796.

Li, H. X., Gatland H. B., Green, A.W. (1997). Fuzzy variable structure control, *IEEE Transactions on Systems, Man, and Cybernetics Part B*, vol. 27, pp. 306-312.

Liang, Q., and Mendel, J. (2000). Interval type-2 fuzzy logic systems: theory and design, *IEEE Transactions on Fuzzy Systems*, vol. 8, pp. 535–550.

Mamdani, E. H. (1974). Application of fuzzy logic algorithms for control of simple dynamic plant. *In Proceedings of the Institute of Electrical Engineering*, vol. 121, pp. 1585-1588.

Martinez, R., Castillo, O., Aguilar, L. T. (2009). Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, *Information Sciences*, vol. 179, pp. 2158-2174.

Mendel, J. (2000). Uncertain Rule-Based Fuzzy Logic: Introduction and New Directions, Prentice Hall, USA.

Mendel, J., John, R., Liu, F. (2006). Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, vol. 14, pp. 808-821.

Mendel, J. (2007). Type-2 fuzzy sets and systems: an overview, *IEEE Computational Intelligence Magazine*, vol. 2, pp. 20 –29.

Mizumoto, M. (1992). Realization of PID controls by fuzzy control methods, *Fuzzy Sets and Systems*, vol. 70, pp. 171–182.

Mudi, R. K., and Pal, N. R. (1999). A robust self-tuning scheme for PI- and PD-type fuzzy controllers, *IEEE Transactions of Fuzzy Systems*, vol. 7 pp. 2-16.

Nie, M., and Tan, W. W. (2008). Towards an efficient type-reduction method for interval type-2 fuzzy logic systems, *in Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 1425–1432.

Nie, M., and Tan, W. W. (2010). Analytical structure and characteristic of symmetric Karnik-Mendel type-reduced interval type-2 fuzzy PI and PD controllers, *IEEE Transactions on Fuzzy Systems*, vol. 20, pp. 416-430

Qiao, W. Z., and Mizumoto, M. (1996). PID type fuzzy controller and parameters adaptive method, *Fuzzy Sets and Systems*, vol. 78, pp. 23-35.

Ozek, M. B., and Akpolat, Z. H. (2008). A software tool: Type-2 fuzzy logic toolbox. *Computer Applications in Engineering Education*, vol. 16, pp. 137-146.

Precup, R. E., and Preitl, S. (2004). Optimization criteria in development of fuzzy controllers with dynamics, *Engineering Applications of Artificial Intelligence*, vol. 17, pp. 661-674.

Sugeno, M., and Kang, G. T. (1988). Structure identification of fuzzy model, *Fuzzy Sets and Systems*, vol. 28, pp. 15-33.

Takagi, T., and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modelling and control, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, pp. 116-132.

Wagner, C., and Hagras, H. (2010). Toward general type-2 fuzzy logic systems based on zSlices, *IEEE Transactions on Fuzzy Systems*, vol. 18, pp. 637-660.

Wu, D. (2010). A brief Tutorial on Interval type-2 fuzzy sets and systems, University of Southern California.

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

Wu, D. (2013). Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons. *IEEE Transactions on Fuzzy Systems*, vol. 21, pp. 80-99.

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

Wu, D., and Tan, W. W. (2005). Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller, *in Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 353–358.

Wu, D., and Tan, W. W. (2006a). Genetic learning and performance evaluation of internal type-2 fuzzy logic controllers. *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 829-841.

Wu, D., and Tan, W. W. (2006b). A simplified type-2 fuzzy logic controller for real-time control, *ISA Transactions*, vol. 45, pp. 503–516.

Wu, D., and Mendel, J. (2007). Enhanced Karnik-Mendel algorithms for interval type-2 fuzzy sets and systems, *in Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 184–189.

Wu, D., and Mendel, J. (2009). Enhanced Karnik-Mendel algorithms. *IEEE Transactions on Fuzzy Systems*, vol. 17, pp. 923-934.

Wu, D., and Mendel, J. (2011). On the continuity of type-1 and interval type-2 fuzzy logic systems, *IEEE Transactions on Fuzzy Systems*, vol. 19, pp. 179-192.

Wu, D., and Tan, W. W. (2010). Interval type-2 fuzzy PI controllers: why they are more robust, *IEEE International Conference on Granular Computing*, pp. 802–807.

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

Wu, H., and Mendel, J. (2002). Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems, *IEEE Transactions on Fuzzy Systems*, vol. 10, pp. 622–639.

Xu, J.X., Hang, C.C, Liu, C., (2000). Parallel structure and tuning of a fuzzy PID controller. *Automatica*, vol. 36, pp. 673–684.

Yeşil, E., Güzelkaya, M., Eksin, İ. (2004). Self-tuning fuzzy PID type load and frequency controller, *Energy Conversion and Management*, vol. 45, pp. 377-390.

Zadeh, L. A. (1965). Fuzzy sets, *Information and Control*, vol. 8, pp. 338-353.

Zadeh, L. A. (1975). The concept of a linguistic variable and it application to approximate reasoning-I, *Information Science*, vol. 19, pp. 829-841.

Zamani, M., Nejati, H., Jahromi, A. T., Partovi, A., Nobari, S. H., Shirazi, G. N. (2008). Toolbox for Interval Type-2 Fuzzy Logic Systems. *International Conference on Information Sciences*, Atlantis Press.

CURRICULUM VITAE

Name Surname: Ahmet Taşkın

Place and Date of Birth: Balıkesir, 02.12.1989

Address: Uğurmumcu Mahallesi Safahat Caddesi No:11 B Blok Daire: 70, Kartal / İstanbul

E-Mail: taskinah@itu.edu.tr

ahmet.taskin@live.com

EDUCATION:

B.Sc.: Istanbul Technical University, Control Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

2012 – AVL Research and Engineering Turkey

2011 – B/S/H/ Home Appliances Turkey

LIST OF PUBLICATIONS:

- Yesil E., **Taşkın A.,** Kumbasar T., 2012. Çevrimiçi Bulanık Ayar Mekanizması İçeren Yeni Bir Kaskat Kontrol Yapısının Tasarımı ve DCS Üzerinde Gerçeklemesi, *Otomatik Kontrol Ulusal Toplantısı (TOK 2012)*, Niğde, Türkiye.

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- Kumbasar T., **Taskin A.,** 2015. An Open Source Matlab/Simulink Toolbox for Interval Type-2 Fuzzy Logic Systems, IEEE Symposium Series on Computational Intelligence *(IEE-SSCI 2015)*, Cape Town, South Africa. (submitted)