

83118

BULANIK KONTROL ve UYGULAMALARI

YÜKSEK LİSANS TEZİ

Elk.Müh. Saime ŞAKA

83118

Tezin Enstitüye Verildiği Tarih : 11 Ocak 1999

Tezin Savunulduğu Tarih : 8 Şubat 1999

Tez Danışmanı : Doç.Dr. Metin GÖKAŞAN,

Diğer Jüri Üyeleri : Doç.Dr. İbrahim EKSİN

Doç.Dr. Hakan TEMELTAS

Metin Gökaşan 4.3.99

İbrahim Eksin 4.03.1999

Hakan Temeltas 06.03.99

ŞUBAT 1999

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

ÖNSÖZ

Tezimin hazırlanma aşamasında, bana yol gösteren Sayın Doç. Dr. İbrahim EKSİN'e, değerli yardımlarını ve ilgisini esirgemeyen Hocam, Sayın Doç. Dr. Metin GÖKAŞAN'a teşekkürlerimi sunmayı bir borç bilirim.

OCAK 1999

Saime ŞAKA



İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	ii
SEMBOL LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖZET	viii
SUMMARY	ix
BÖLÜM 1 GİRİŞ	1
1.1 Bulanık Sistem Nedir?	1
BÖLÜM 2 BULANIK MANTIK	4
2.1 Bulanık Mantık Nedir?	4
2.2 Bulanık Sistemler Nerede ve Nasıl :Kullanılıyor?	5
2.2.1 Bulanık Kontrollü Çamaşır Makineleri	5
2.2.2 Dijital Görüntü Düzeltme	6
2.2.3 Arabalarda Kullanılan Bulanık Sistemler	6
2.3 Bulanık Kümeler ve Bulanık Kümeler Üzerinde Temel İşlemler	7
2.4 Bulanık Kümelerle İlgili Temel Kavramlar	9
2.5 Bulanık Kümelerde İşlemler	10
2.5.1 Bulanık Tümleme	10
2.5.2 Bulanık Birleşme Özelliği S-Normu	10
2.5.3 Bulanık Kesişme Özelliği T-Normu	11
2.6 Dilsel Değişkenler ve Bulanık EĞER-İSE Kuralları	11
2.7 Bulanık EĞER-İSE Kurallarının Yorumlanması	13
BÖLÜM 3 BULANIK KONTROLÖRLER	15
3.1 Giriş	15
3.2 Bulanık Kural Tabanı	16
3.2.1 Bulanık Kural Tabanının Yapısı	16
3.2.2 Bulanık Kural Tabanının Özellikleri	17
3.3 Bulanık Çıkarım	18
3.3.1 Birleşik Tabanlı Çıkarım	18
3.3.2 Tek Kural Tabanlı Çıkarım	20

3.3.3 En Çok Kullanılan Çıkarım Motorları	20
3.4 Bulandırıcı	22
3.4.1 Tek Noktalı Bulandırıcı	23
3.4.2 Gaussian Bulandırıcı	23
3.4.3 Üçgen Bulandırıcı	24
3.5 Durulayıcı	24
3.5.1 Ağırlık Merkezli Durulayıcı	24
3.5.2 Ortalama Merkezli Durulayıcı	25
3.5.3 Maksimum Durulayıcı	26
BÖLÜM 4 BULANIK KONTROLÖR TASARLANMASI	27
4.1 Giriş	27
4.2 Bulanık Kontrolör Yapıları	28
4.2.1 Bulanık PD Kontrolör	28
4.2.2 Bulanık PI Kontrolör	28
4.2.3 Bulanık PID Kontrolör	29
4.2.4 Bulanık Karma Kontrolörler	29
4.3 Giriş-Çıkış Bilgilerini Kullanarak Bulanık Kontrolör Tasarımı	31
4.3.1 Look-Up Tablosu Kullanarak Bulanık Kontrolör Tasarımı	31
BÖLÜM 5 SİMÜLASYONLAR	34
5.1 Dijital PID Kontrolör Tasarımı ve Simülasyonu	34
5.1.1 Ziegler-Nichols Metodu	34
5.1.2 Analitik Olarak Dijital PID Kontrolör Tasarımı ve Simülasyonu	37
5.2 Look-Up Kural Tablosunu Kullanarak Bulanık Kontrolör Simülasyonu	40
5.3 Bulanık PI Kontrolör Simülasyonu	46
5.4 Bulanık Kontrolör Tasarımı ve Simülasyonu	53
SONUÇLAR ve ÖNERİLER	58
KAYNAKLAR	59
EKLER	60
ÖZGEÇMİŞ	66

SEMBOL LİSTESİ

$\mu_A(x)$: A Bulanık Kümesinin Üyelik Fonksiyonu
$A \cap B$: A ve B Kümelerinin Kesişimi
$A \cup B$: A ve B Kümelerinin Birleşimi
t	: Bulanık Kesişme t Normu
s	: Bulanık Birleşme s Normu
X	: Dilsel Değişkenin Adı
T	: Dilsel Değişken X'in Alabileceği Değerler
M	: Bulanık Kurallar Kümesi
Q	: Bulanık Bağıntı
$hgt(B')^-$: B' nün Üyelik Fonksiyonunun Maksimum Değerlerini Aldığı Noktalar Kümesi
e	: Hata
de, e	: Hatanın Değişimi
u	: Kontrolör Tarafından Üretilen Kontrol İşareti
r	: Resetleme Parametresi
ξ	: Söndürme Oranı

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 Bulanık kontrolörün yapısı	2
Şekil 2.1 Bulanık mantığın uygulama sahaları	5
Şekil 2.2a Sıfıra yakın sayılar için üyelik fonksiyonu	8
Şekil 2.2b Sıfıra yakın sayılar için diğer bir üyelik fonksiyonu	8
Şekil 2.3 Sayısal değişkenler ve dilsel değişkenler	12
Şekil 2.4 Bir arabanın hız tanımı için bulanık üyelik fonksiyonu	13
Şekil 3.1 Ağırlık merkezli durulayıcı	24
Şekil 3.2 Ortalama merkezli durulayıcı	25
Şekil 3.3 Maksimum durulayıcı	26
Şekil 4.1 Bulanık PD kontrolör yapısı	28
Şekil 4.2 Genelleştirilmiş bulanık PI kontrolör yapısı	28
Şekil 4.3 Bulanık PD+PI kontrolör yapısı	29
Şekil 4.4 Bulanık PD+Bulanık I kontrolör yapısı	29
Şekil 4.5 Karma kontrolör yapısı	30
Şekil 4.6a İki seviyeli bir bulanık kontrol sisteminin yapısı	30
Şekil 4.6b İki seviyeli bir bulanık kontrol sisteminin yapısı	31
Şekil 4.7 İki farklı giriş durumu için giriş-çıkış çiftleri ve üyelik fonksiyonları	33
Şekil 4.8 Bulanık look-up kural tablosu	34
Şekil 5.1 $T_s=0.01$ sn için Ziegler-Nichols yöntemine göre hesaplanan kontrolör parametrelerine göre birim basamak sistem cevabı	37
Şekil 5.2 PD kontrolörlü sistemin z-düzlemi köklerin yer eğrisi	38

Şekil 5.3 $K_p=123.92$ ve $K_D=33.17$ için sistem birim basamak cevabı	39
Şekil 5.4a Bulanık kontrolör yapısı	40
Şekil 5.4b Look-up tablosuyla oluşturulan bulanık kontrolörlü sistem modeli	40
Şekil 5.5 Bulanık kontrolör kaba kontrol look-up tablosu	41
Şekil 5.6 Bulanık kontrolör hassas kontrol look-up tablosu	42
Şekil 5.7 Look-up kural tablosu kullanarak elde edilen sistem cevabı	46
Şekil 5.8 Genelleştirilmiş bulanık PI kontrolör modeli	47
Şekil 5.9 Resetleme işleminde p'nin etkisi	47
Şekil 5.10 FLC1-du'yu hesaplamak için bulanık kural tablosu	48
Şekil 5.11 e, de ve du üyelik fonksiyonları	48
Şekil 5.12 FLC2-r'yi hesaplamak için bulanık kural tablosu	48
Şekil 5.13 e, de ve r üyelik fonksiyonları	49
Şekil 5.14 FLC1-du kontrol işareti değişimi giriş-çıkış arayüzeyi	50
Şekil 5.15 FLC2-r resetleme parametresi bulanık kontrolör giriş-çıkış arayüzeyi	50
Şekil 5.16a $S_e=1$ ve $S_{du}=1$ için S_{de} 'ye göre sistem cevabı	51
Şekil 5.16b $S_{de}=1$ ve $S_{du}=1$ için S_e 'ye göre sistem cevabı	51
Şekil 5.17 Resetleme parametrelili genelleştirilmiş bulanık PI kontrolör Simülasyonu	52
Şekil 5.18 Faz yüzeyini kullanarak kural tablosu düzenleme	53
Şekil 5.19 Bulanık kontrolör kural tablosu	53
Şekil 5.20a Hata için belirlenen üyelik fonksiyonları	54
Şekil 5.20b Hatanın değişimi için belirlenen üyelik fonksiyonları	54
Şekil 5.21 Çıkışlar için belirlenen üyelik fonksiyonları	55
Şekil 5.22 Bulanık kontrolör giriş-çıkış arayüzeyi	55
Şekil 5.23 $G(s)=1/(s+3.6)$ için klasik bulanık PI kontrolörle sistem birim basamak cevabıHata için belirlenen üyelik fonksiyonları	56

ÖZET

Bu çalışma son yıllarda çok hızlı bir gelişme gösteren bulanık sistemler ve bulanık kontrol konusundaki gelişmeleri tanıtmaktadır.

Gerçek hayatta karşımıza çıkan kontrol problemlerinin birçoğu en iyi dilsel tanımlamalarla açıklanabilir ve ayrıca kontrolünde çok hassas olunması gerekmez. Bu nedenle bulanık kontrolün en iyi uygulama alanları lineer olmayan, iyi tanımlanmamış, zamanla değişen sistemlerdir. Sistem karmaşıktıkça bulanık kontrol klasik kontrol karşısında ekonomik bir alternatif olmaya başlamaktadır.

Aslında günlük hayatımızın hemen her anında farkında olmadan bulanık kontrol uygulanmaktadır. Bir çok problem çeşitli dilsel ifadeler kullanarak farkında olmadan çok iyi bir şekilde çözülebilmektedir. Örneğin yıllardır araba kullanan bir insan bir arabanın nasıl parkedileceğini dilsel değişkenlerle ifade ederek çok doğru bir şekilde açıklayabilmektedir. Bulanık kontrolün de amacı; bu uzman bilgisini ve tecrübesini çeşitli dilsel değişkenlerle formüle ederek süreç kontrolünde kullanmaktır.

FUZZY CONTROL and APPLICATIONS

SUMMARY

The field of fuzzy systems and control has been making rapid progress in recent years. With the practical success of fuzzy control in consumer products and industrial process control, there has been an increasing work on theoretical studies of fuzzy systems and fuzzy control. As a result of these studies, fuzzy systems and fuzzy control is becoming clearer. In literature, there are two kinds of justification for fuzzy systems theory.

1. The real world is too complicated and obtaining mathematical model is difficult or impossible. Therefore approximation must be done to obtain a reasonable model.
2. In the complicated systems, human knowledge becomes increasingly important. Then, there must be a theory to formulate human knowledge systematically, to use together with other information like mathematical models and sensory measurements.

What Are Fuzzy Systems?

Fuzzy systems are knowledge-based or rule-based systems. The IF-THEN rules are some words to introduce the fuzzy system and these rules gives the fuzzy membership functions. For example, the following is a fuzzy IF-THEN rule.

IF the speed of a car is high, THEN apply less force to the accelerator

A fuzzy system is constructed from a collections of fuzzy IF-THEN rules. In order to use fuzzy systems in engineering systems, a simple method is to add a fuzzifier, which transforms a real valued variable into a fuzzy set and a defuzzifier, which transforms a fuzzy set into a real valued variable, to the output (Figure 1).

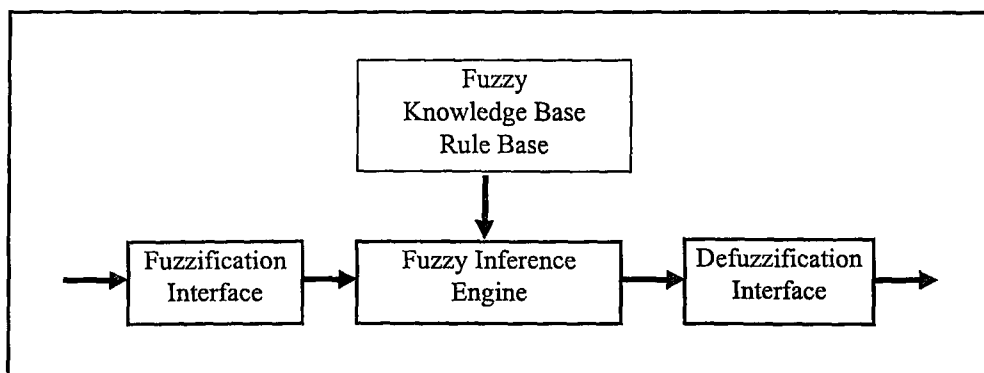


Figure 1. Basic Configuration of Fuzzy Controller

Where Are Fuzzy Systems Used and How?

Fuzzy systems have been applied to a wide variety of fields ranging from control, signal processing, communications, integrated circuit manufacturing and expert systems to business, medicine, psychology, etc. However the most significant applications have concentrated on control problems.

Fuzzy systems can be used as open-loop controllers or closed-loop controllers. When it is used as an open-loop controller, the fuzzy system usually sets up some control parameters and then the system operates according to these control parameters. When it is used as a closed-loop controller, the fuzzy system measures the outputs of the process and takes control actions on the process continuously. Applications of fuzzy systems in industrial processes belong to this category.

Fuzzy Logic Controller (FLC) :

FLC is composed of four components as shown in Figure 1, which is fuzzification interface, a knowledge or rule base, fuzzy inference engine, and a defuzzification interface.

The fuzzification interface involves the following functions:

1. Measures the real values of the input variables.
2. Performs a scale mapping that transfers the range of values of input variables into corresponding universes of discourse.
3. Performs the function of fuzzification that converts input data into suitable linguistic values which may be viewed as labels of fuzzy set.

The knowledge base comprises a knowledge of the application domain and the control goals. It consists of a data base and a linguistic(fuzzy) control rule base.

1. The data base provides necessary definitions which are used to define linguistic control rules and fuzzy data manipulation in a FLC.
2. The rule base characterizes the control goals and control policy of the domain experts by means of a set of linguistic variables.

The fuzzy inference engine is the kernel of an FLC it has capability of simulating human decision making based on fuzzy concepts and the rules of inference in fuzzy logic.

The defuzzification interface performs the following functions.

1. A scale mapping, which converts the range of values of output variables into corresponding universes of discourse.

2. Defuzzification which yields a nonfuzzy control action from an inferred fuzzy control action.

Fuzzy Controller Design

Conventional controller design starts with a mathematical model of the process, on the other hand fuzzy controller design starts with heuristics and human experience which is defined in terms of fuzzy IF-THEN rules. Experienced human experts can provide heuristics and rule –of-thumb that are very useful for controlling the process.

The design approaches for fuzzy controllers are classified into two categories. The first one is trial-and- error approach, the other one is the theoretical approach.

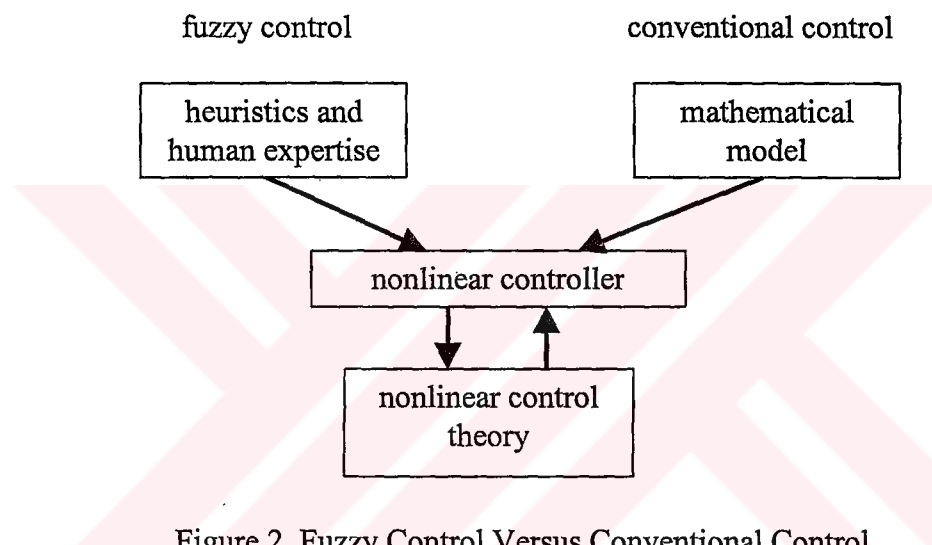


Figure 2. Fuzzy Control Versus Conventional Control

In the trial-and-error approach, a set of fuzzy IF-THEN rules collected from human experts, operating manuals or using experience based knowledge. Then fuzzy controllers are designed from these fuzzy IF-THEN rules, finally, the fuzzy controllers are tested in the real system and the performance of the system is not satisfactory,- the rules are fine-tuned or rules are redesigned . In the theoretical approach the structure and parameters of the fuzzy controller are designed in such a way that certain performance criteria are guaranteed.

Design of Fuzzy Systems from Input-Output Data

Fuzzy Systems are used to formulate human knowledge. Expert human knowledge about a particular engineering problem may be classified as conscious knowledge and subconscious knowledge. By conscious knowledge, fuzzy IF-THEN rule can be constructed easily. For subconscious knowledge, some demonstrations are required for the typical situations of the system. When the expert is demonstrating, the inputs and outputs are measured. These input-output pairs are a set. The fundamental problem is to construct fuzzy systems using these input-output pairs.

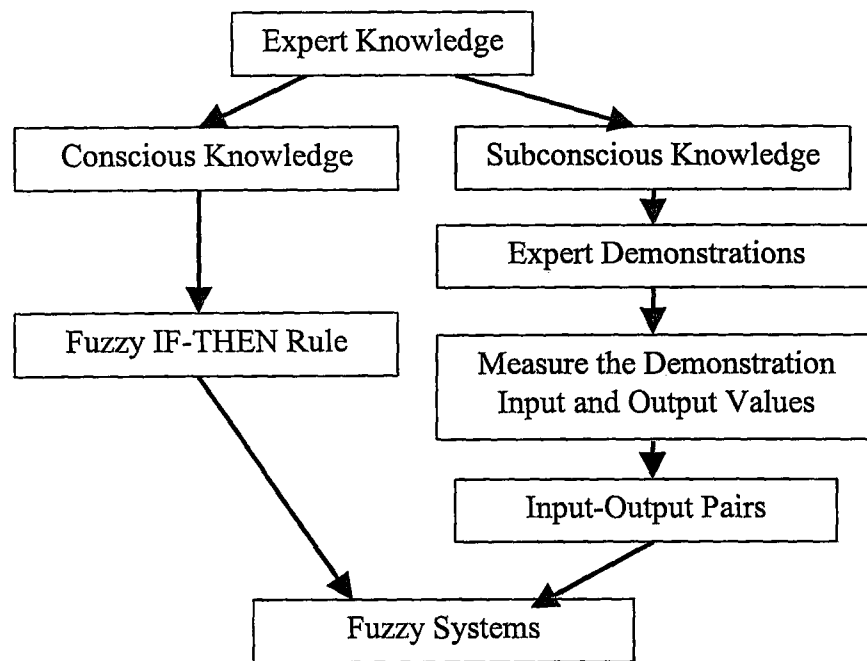


Figure 3. Converting expert knowledge into fuzzy systems

Fuzzy Controller Types

So far, mainly two types of fuzzy logic controller studied : one position-type fuzzy controller which generates control input(u) from error(e) and error rate (\dot{e}), and the other is velocity-type fuzzy logic controller which generates incremental input(Δu) from error and error rate. The former is called PD FLC and the latter is called PI FLC according to the information they process. Using PD FLC and PI FLC, PID FLC controller was developed. For complex systems the single loop control systems may not effectively achieve the control objectives, and hybrid controllers are used in this situation.

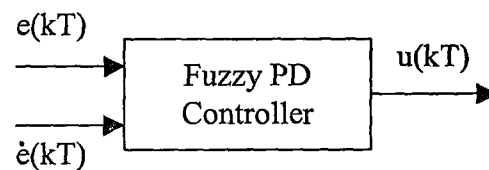


Figure 4. Fuzzy PD Controller

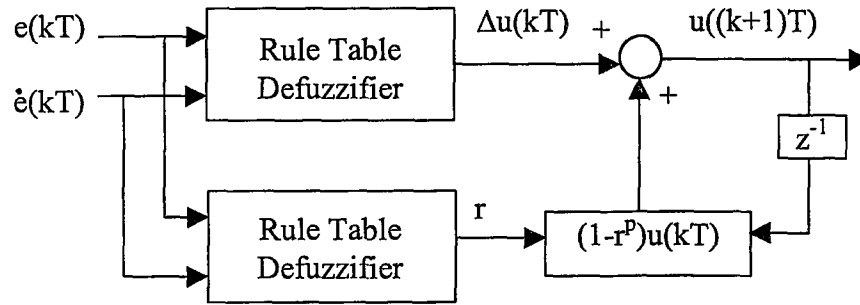


Figure 5. Fuzzy PI Controller

In the literature, some hybrid fuzzy controllers applications can be seen. The main advantage of two level control is that different controllers can be designed to different objectives, so that each controller is simpler and performance is improved.

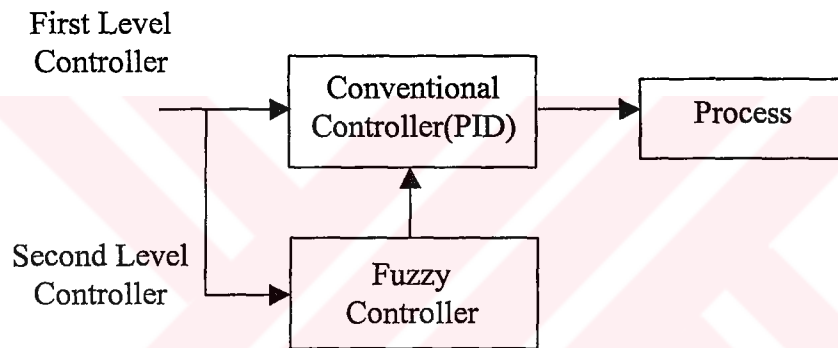


Figure 6.a Architecture of a two-level fuzzy control system

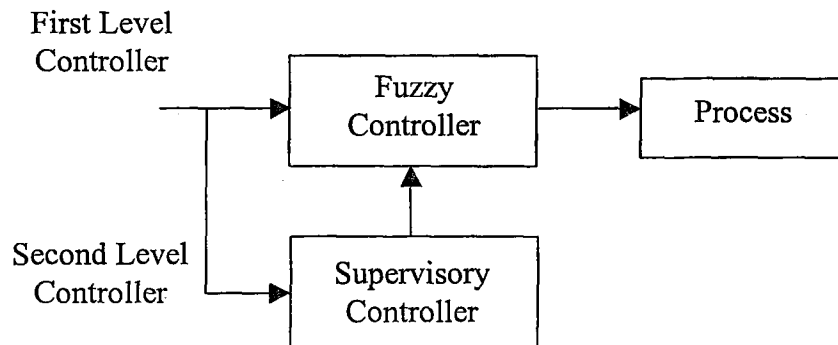


Figure 6.b Architecture of a two-level fuzzy control system

Conclusion

In this thesis, fuzzy logic and fuzzy controllers are investigated and some controller simulations are realized.

BÖLÜM 1

GİRİŞ

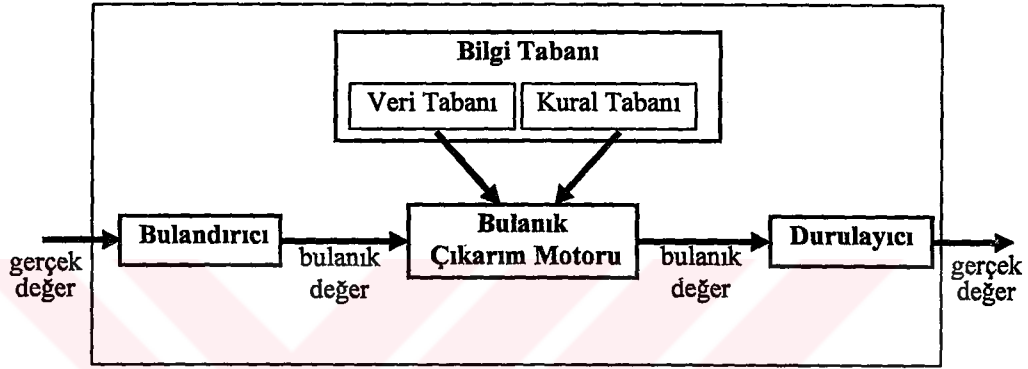
1.1 BULANIK SİSTEM NEDİR ?

Bulanık küme kavramı ilk olarak 1965 yılında California Üniversitesi öğretim üyelerinden aslen Azerbaycan'lı Prof. Lotfi Zadeh tarafından bir makaleyle ortaya atılmış ve hızla gelişerek birçok bilim adamının ilgisini çeken, araştırmaya açık yeni bir konu olmuştur. Bulanık küme teorisinin ortaya atılmasından sonra Zadeh 1973'te yayınladığı notlarında bulanık küme teorisinin en iyi yaklaşıklıkla insanın karar verme sistemini modelleyebilecek yapıda olduğu fikrini ileri sürmüştür. Geçen zaman içerisinde bulanık kontrolün dayandığı bulanık mantığın, insan düşünme yapısına ve dilsel değişkenlerine klasik mantıktan çok daha yakın olduğu kabul edilmiştir[1].

Bulanık mantık kavramıyla beraber araştırmacılar arasında bir tartışma ortamı doğdu. Bazı araştırmacılar bulanık mantığın temel bilimsel prensiplere aykırı olduğunu göstermeye çalıştılar. Özellikle olasılık ve istatistikle uğraşan matematikçiler olasılık teorisine belirsizlik problemlerinin çözülebileceğini savundular[1].

1965'te bulanık küme kavramı ve 1968'de bulanık algoritmalar, Zadeh ve 1970'te bulanık karar verme yöntemleri, Zadeh ve Bellman tarafından ortaya atıldı. 1973'te, Zadeh "Kompleks Sistemlerin ve Karar Verme İşlemlerinin Analizine Yeni Bir Yaklaşım" adıyla bulanık kontrolün temelini atan bir makale yayınladı. Zadeh bu makalesinde dilsel değişkenleri ve insan bilgisini formüle etmek için EĞER- İSE kurallarını kullanmayı önerdi[1].

1970'lerde gerçek sistemler için bulanık kontrolörler kullanılmaya başlandı. Bulanık mantıkla kontrol, ilk olarak 1975'de Mamdani ve arkadaşları tarafından bir buhar makinesinin denetimi için bulanık kural tabanlı bir kontrolör geliştirmeleriyle başladı [Şekil 1.1]. Bu uygulama sonucunda, nonlinear kontrol problemleri için bulanık kontrolörün klasik kontrolörlere göre çok daha kolay geliştirildiği ve oldukça iyi sonuçlar verdiği belirtildi. 1978'de de, ilk kez olarak bir çimento fırınına tüm endüstriyel süreç için bulanık bir kontrolör geliştirildi.



Şekil 1.1 Bulanık kontrolör yapısı

1980'lerde bulanık mantık teorik olarak çok yavaş gelişmiştir. Bu senelerde bulanık kontrolörler, matematiksel model eksikliği nedeniyle geleneksel kontrol yöntemlerinin uygulanamadığı sistemlerde uygulanmaya başlanmıştır. 1980'de Sugeno, Japonların ilk bulanık mantık uygulaması olan Fuji Electric su arıtma tesisinin kontrolünü gerçekleştirdi. 1983'te, komutlarla kendi kendine park eden bir araba olan bulanık kontrollü bir robot çalışmasına başlandı. Yine bu arada Hitachi'den Miyamoto, Sendai metrosu için bulanık kontrol sistemi geliştirmeye başladı. 1987'de bu proje bitirildi ve başarılı bir uygulama gerçekleştirildi. Sendai metrosunun devreye alınmasından sonra, temmuz 1987'de Tokyo'da ikinci uluslararası bulanık sistemler konferansı yapıldı. Bu konferansta katılımcılara Sendai metrosu ve ayrıca iki boyutlu masa tenisi oynayabilen bir bulanık robot kolu da tanıtıldı[1].

Sendai metrosu uygulamasından sonra bulanık kontrol konusundaki çalışmalar yeni bir ivme kazanmış ve endüstriyel uygulama alanları hızla artmıştır. Çalışmaların uluslararası alanda koordinasyonu amacı ile Japonya'da 1989 yılında LIFE (Laboratory for International Fuzzy Engineering) adlı bir laboratuvar kurulmuştur. Bu laboratuvar da yapılan araştırma çalışmalarına aralarında Hitachi, Toshiba, Omron, Matsushita gibi ünlü Japon firmalarının yanı sıra IBM, NCR ve Thomson gibi Japonya dışı firmalarında aralarında bulunduğu 51 firma katılmıştır. LIFE laboratuvarı üç ayrı laboratuvar dan oluşturuldu. İlk laboratuvar, bulanık kontrol konusunda çalışma yapmak amacıyla, ikinci laboratuvar bulanık akıllı bilgi işleme ve üçüncü laboratuvar ise bulanık bilgisayarlar konusunda araştırma yapmak amacıyla kuruldu[2]. Daha sonra şubat 1992'de San Diego'da ilk olarak bulanık sistemler üzerine IEEE uluslararası konferansı yapıldı. Bu konferans bulanık mantık teorisinin, en büyük mühendislik organizasyonu IEEE tarafından kabul edildiğini göstermektedir[1].

BÖLÜM 2

BULANIK MANTIK

2.1 BULANIK MANTIK NEDİR?

İlk olarak Prof. Zadeh tarafından kullanılan bulanık mantık, temelde çok değerli (multivalued) mantık, olasılık kuramı, yapay zeka ve yapay sinir ağları alanları üzerine oturtulmuş olup olayların oluşum olasılığından çok olabilirliğiyle ilgilenen bir kavramı tanımlamaktadır. Olasılık ve bulanıklık kavramları arasındaki en önemli farklılık bulanıklığın bir deterministik belirsizlik olmasıdır[3].

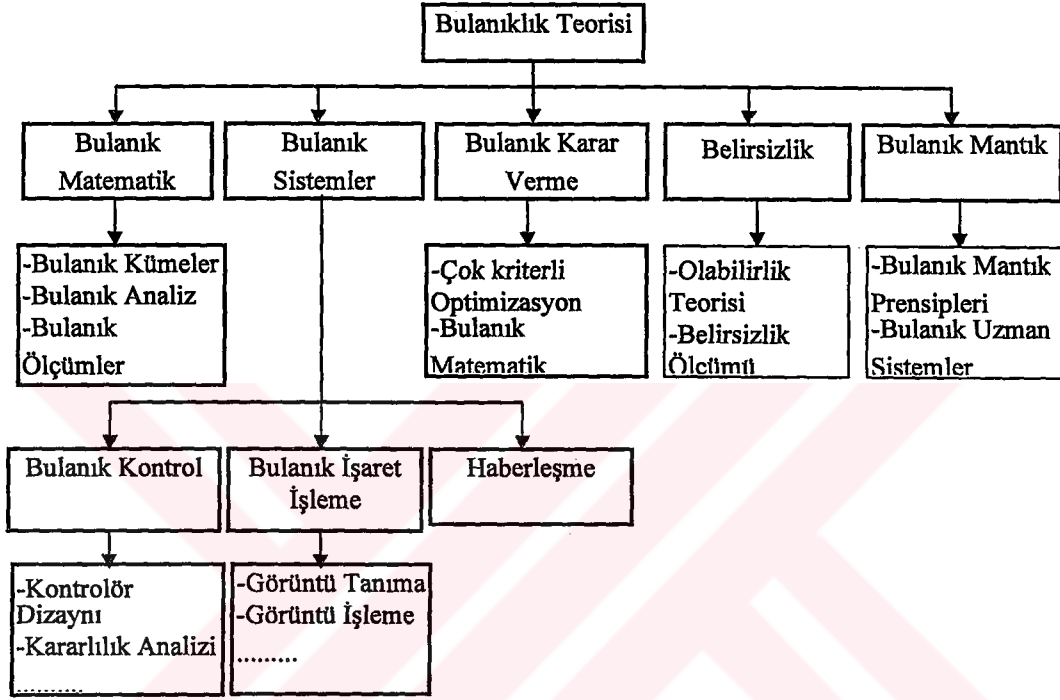
Bulanık sistemler bilgi tabanlı ve kural tabanlı sistemlerdir. Bulanık mantığın temeli EĞER-İSE kurallarından oluşmaktadır.

Literatürde bulanık mantık için iki şekilde açıklama yapılmaktadır[1]:

1. Gerçek dünya çok net ve açık tanımlamalar yapmak için fazlasıyla komplikedir. Bundan dolayı bir yaklaşıklık (bulanıklık) yapmak zorunluluğu vardır.
 2. Gerçek sistemler için önemli bilgiler iki kaynaktan elde edilir. Birinci kaynak; insan tecrübesi, diğeri ise sensör ölçümleri ve fiziksel kanunlardan çıkarılan matematiksel modellerdir. Hedef, bu iki bilgi kaynağını kullanarak sistem dizaynını yapmaktır. Bu kombinasyonu oluşturmak için insan tecrübesini ve bilgisini, matematiksel modele ve sensör ölçümlerine göre nasıl formüle edilebileceğini saptamak anahtar problemdir. Diğeri bir deyişle sorun insan bilgisinin ve tecrübesinin nasıl formüle edileceğidir.
-

2.2 BULANIK SİSTEMLER NEREDE VE NASIL KULLANILIYOR?

Bulanık sistemler kontrol, işaret işleme, haberleşme ve uzman sistem olarak tıpta, iş hayatında çok geniş alanda uygulanmaktadır. Bununla birlikte en önemli uygulamalar kontrol problemleri üzerinde yapılmaktadır[1].



Şekil 2.1 Bulanık Mantığın Uygulama Sahaları

2.2.1 Bulanık Kontrollü Çamaşır Makinaları

Bulanık kontrollü çamaşır makinaları bulanık kontrolün uygulandığı en önemli tüketici ürünlerinden biridir. 1990'da Matsushita Elektrik tarafından Japonya'da üretilmiştir. Bulanık sistem, çamaşır makinasındaki çamaşırların kirlilik oranına ve miktarına bağlı olarak yıkama süresini ve kullanılacak su miktarını ayarlamaktadır. Bu uygulamada kullanılan bulanık sistemde üç giriş ve iki çıkış kullanılmaktadır. Giriş olarak çamaşırların kirlilik miktarı, türü ve çamaşırın miktarı alınmaktadır. Çıkış olarak yıkama süresi ve kullanılan su miktarı ayarlanmaktadır. Optik sensörle kirliliğin yağlı veya çamurlu olup olmadığı ve bir yük sensörüyle de çamaşırın

miktarı anlaşılmaktadır. Sensörlerden alınan bilgilerle daha fazla çamaşır miktarı ve kirlilik oranı; daha fazla yıkama süresi ve daha fazla yıkama suyu kullanılması gerektiğini göstermektedir. Bu dilsel tanımlamalar EĞER-İSE kurallarıyla bulanık sistemi kurmak için kullanılmıştır.

2.2.2 Dijital Görüntü Düzeltme

Video kamera kullanırken sarsıntısız olarak çekim yapmak oldukça zordur. Japonya'da Matsushita el titrerken çekilen görüntüyü düzelten bulanık sistem mantığının kullanıldığı bir video kamera geliştirdiler. Bulanık sistem mantığında yine dilsel değişkenlerden yararlanıldı.

EĞER görüntüdeki bütün noktalar aynı yönde kayıyor İSE el titriyor

EĞER görüntüdeki sadece bazı noktalar kayıyor İSE el titremiyor

Görüntü düzeltici mevcut kareyi bir önceki kareyle karşılaştırarak düzeltme yapmaktadır. Eğer bütün noktalar kaymış gözüküyorsa, elin titrediği anlaşılmakta ve bulanık sistem karedeki kaymayı kompanze etmektedir.

2.2.3 Arabalarda Kullanılan Bulanık Sistemler

Bir otomobil birçok sistemin bir arada olduğu kompleks bir süreçtir ve bulanık mantık hemen hemen fren, vites, motor gibi bütün bölümlerde uygulanmıştır. Örneğin, Nissan yakıt kullanımını %12 ile 17 arasında azaltan bulanık otomatik güç aktarımı konusunda patent almıştır. Nissan ayrıca frende kilitlenmeyi önleyen bulanık bir sistem geliştirmiştir. Buradaki sorun frene çok fazla basıldığı takdirde kilitlenme olmaksızın frenlemeyi sağlamaktır. Burada dilsel tanım olarak;

EĞER araba çok hızlı yavaşlıyor İSE sistem frenlemeyi üzerine alır ve fren üzerindeki basıncı düşürür.

2.3 BULANIK KÜMELER VE BULANIK KÜMELER ÜZERİNDE TEMEL İŞLEMLER

Klasik küme teorisinde bir elemanı ele aldığımızda bu elemanı iki farklı kategoride değerlendirmeye alırız (örneğin iyi-kötü, tek-çift, siyah-beyaz gibi). Ancak gerçek hayatta bu iki değerli mantık çoğu uygulamada yeterli değildir. Genellikle bu iki değer arasındaki kesin sınır belirgin değildir.

Eski bir Yunanlı düşünür tarafından ortaya atılan bir problem bu sorunun varlığını kanıtlamaktadır.

“..... bir tek buğday bir yığın oluşturmaz, iki tane de, üç tane de ... Diğer yandan 100 milyon buğdayın bir yığın oluşturacağını herkes kabul eder. Peki kesin sınır nedir ? 325647 buğday bir yığın oluşturmazken 325648 buğdayın bir yığın oluşturabileceğini söyleyebilir miyiz?”

A kümesi U uzayında bir klasik küme olsun. Bu küme, bütün üye elemanları listeleyerek ya da üye elemanların sağlayacağı özelliklerin belirlenmesiyle (kural metodu) oluşturulabilir. Liste metodu sadece sonlu kümeler için kullanılır. Kural metodunda ise A kümesi şu şekilde ifade edilir.

$$A = \{ x \in U \mid x \text{ bazı koşulları sağlar} \} \quad (2.1)$$

A kümesini belirlemek için diğer bir metot ise 0-1 üyelik fonksiyonunu gösteren üyelik metodudur. $\mu_A(x)$ üyelik fonksiyonu olmak üzere

$$\mu_A(x) = \begin{cases} 1, & \text{eğer } x \in A \\ 0, & \text{eğer } x \notin A \end{cases} \quad (2.2)$$

Tanım : Bir bulanık küme, $[0,1]$ arasında değerler alan üyelik fonksiyonu $\mu_A(x)$ tarafından tanımlanır. Bir klasik kümenin üyelik fonksiyonu sadece iki değer yani 0

ile 1 değerlerini alabilirken, bir bulanık kümenin üyelik fonksiyonu 0 ile 1 arasında herhangi bir değer alabilmektedir.

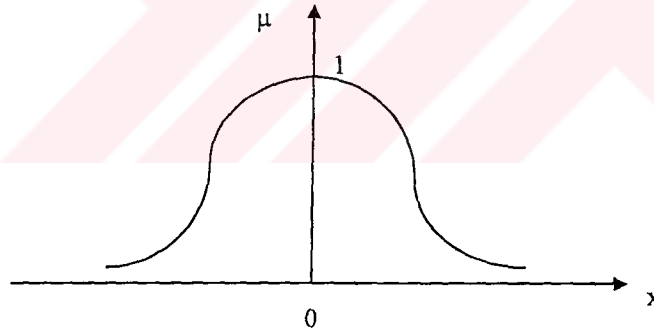
Örnek: Z sıfıra yakın sayıların kümesi olsun. Z için mümkün olan üyelik fonksiyonu

$$\mu_Z(x) = e^{-x^2}$$

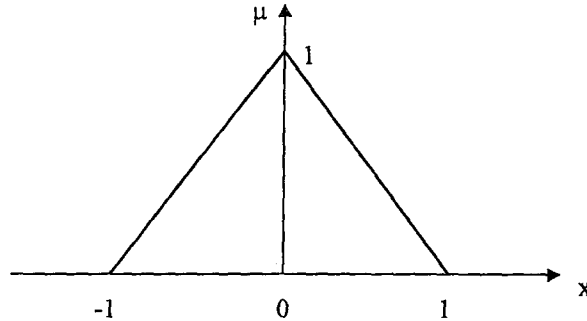
Bu Gaussian fonksiyon sıfıra yakınlık üyelik fonksiyonu $\mu_Z(0) = e^0 = 1$. Üyelik fonksiyonunu şu şekilde de tanımlayabiliriz.

$$\mu_Z(x) = \begin{cases} 0 & , \text{ eğer } x < -1 \\ x + 1 & , \text{ eğer } -1 \leq x < 0 \\ 1 - x & , \text{ eğer } 0 \leq x < 1 \\ 0 & , \text{ eğer } 1 \leq x \end{cases}$$

μ_Z üyelik fonksiyonuna göre -1 ve 1 arasındaki sayılar 0 ve 1 arasında üyelik derecesine sahiptir. Şekil 2.2a ve Şekil 2.2b'de sırasıyla seçilen üyelik fonksiyonlarının grafiksel ifadesi görülmektedir.



Şekil 2.2a Sıfıra yakın olan sayılar için gaussian üyelik fonksiyonu



Şekil 2.2b Sıfıra yakın olan sayılar için üçgen üyelik fonksiyonu

Örnekten üç önemli sonuç çıkmaktadır.

1. Örnekte sıfıra yakın olan sayıları tespit ederken kesin bir tanım yoktur. Bundan dolayı problemi çözerken farklı üyelik fonksiyonları tanımlanabilir. Bununla birlikte üyelik fonksiyonları tek başlarına kesin matematiksel fonksiyonlardır. Bir bulanık problem için üyelik fonksiyonu belirlendiği zaman bulanıklık durumu ortadan kalkar.

2. Bulanık bir problemi çözerken nasıl bir üyelik fonksiyonu tanımlamalıyız?

Bunun için iki yöntem söz konusudur. Birinci yöntem; üyelik fonksiyonlarını belirlerken insan tecrübesi ve bilgisinden yararlanmaktır. İkinci yöntem ise; çeşitli sensörler aracılığıyla toplanan bilgilerden yararlanarak üyelik fonksiyonlarını oluşturmaktır.

3. Bir bulanık küme üyelik fonksiyonuyla bire-bir bir ilişkiye sahiptir. Yani bir bulanık kümeyle ilgili sadece bir üyelik fonksiyonu olmalıdır.

2.4 BULANIK KÜMELERLE İLGİLİ TEMEL KAVRAMLAR

Burada bulanık kümelerle ilgili temel kavramları tanıtacağız.

Tanım : A ve B bulanık kümelerinin eşitlik, kapsama, tümleyen, birleşme ve kesişme özellikleri şu şekildedir.

- A ve B kümeleri ancak ve ancak her $x \in U$ için $\mu_A(x) = \mu_B(x)$ ise eşittir.
- Eğer her $x \in U$ için $\mu_A(x) \leq \mu_B(x)$ ise B kümesi A kümesini kapsar ve $A \subset B$ şeklinde gösterilir.
- Bir bulanık kümenin tümleyeni de bir bulanık kümedir ve üyelik fonksiyonu şu şekilde belirlenir.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.3)$$

- A ve B bulanık kümelerinin birleşimi $A \cup B$ ile gösterilir ve üyelik fonksiyonu ;

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (2.4)$$

- A ve B bulanık kümelerinin kesişimi $A \cap B$ ile gösterilir ve üyelik fonksiyonu ;

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad (2.5)$$

2.5 BULANIK KÜMELERDE İŞLEMLER

2.5.1 Bulanık Tümleme

$c: [0,1] \rightarrow [0,1]$ olmak üzere $c[\mu_A(x)] = \mu_A(x)$ sağlamakta olduğunu kabul edelim.

Aksiyom 1 : $c(0)=1$ ve $c(1)=0$ sınır koşulları

Aksiyom 2 : Tüm $a, b \in [0,1]$ için; a ve b $a=\mu_A(x)$ ve $b=\mu_B(x)$ bulanık kümelerinin üyelik fonksiyonları olsun. Eğer $a < b$ ise $c(a) \geq c(b)$ sağlanır.

Tanım : Herhangi bir fonksiyon $c: [0,1] \rightarrow [0,1]$ yukarıdaki aksiyomları sağlıyorsa bulanık tümlendirdir.

2.5.2 Bulanık Birleşme Özelliği - S Normu

$s: [0,1] \times [0,1] \rightarrow [0,1]$ bir fonksiyon olsun.

$$\begin{aligned} [\mu_A(x), \mu_B(x)] &= \mu_{A \cup B}(x) \text{ dolayısıyla} \\ [\mu_A(x), \mu_B(x)] &= \max[\mu_A(x), \mu_B(x)] \end{aligned} \quad (2.6)$$

s fonksiyonunun birleşme özelliğini sağlaması için aşağıdaki aksiyomları sağlamalıdır.

Aksiyom s1: $s(1,1)=1$, $s(0,a)=s(a,0)=a$, sınır koşulu

Aksiyom s2: $s(a,b)=s(b,a)$, yer değiştirme özelliği

Aksiyom s3: Eğer $a \leq a'$ ve $b \leq b'$ ise $s(a,b) \leq s(a',b')$,

Aksiyom s4: $s(s(a,b),c)=s(a,s(b,c))$ (birleşme özelliği)

Yukarıdaki s1-s4 aksiyomlarını sağlayan bir fonksiyon $s:[0,1] \times [0,1] \rightarrow [0,1]$ s normu olarak adlandırılır.

2.5.3 Bulanık Kesişme Özelliği - T Normu

$t: [0,1] \times [0,1] \rightarrow [0,1]$ bir fonksiyon olsun.

$t[\mu_A(x), \mu_B(x)] = \mu_{A \cap B}(x)$ dolayısıyla

$$t[\mu_A(x), \mu_B(x)] = \min[\mu_A(x), \mu_B(x)] \quad (2.7)$$

t fonksiyonunun kesişme özelliğini sağlaması için aşağıdaki aksiyomları sağlaması gerekir.

Aksiyom t1: $t(0,0)=0$, $t(a,1)=t(1,a)=a$, sınır koşulu

Aksiyom t2: $t(a,b)=t(b,a)$, yer değiştirme özelliği

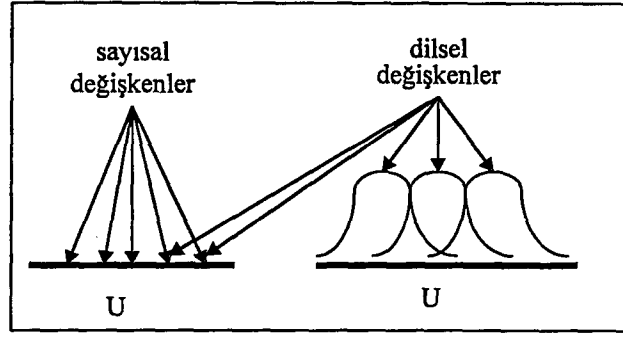
Aksiyom t3: Eğer $a \leq a'$ ve $b \leq b'$ ise $t(a,b) \leq t(a',b')$

Aksiyom t4: $t[t(a,b),c]=t[a,t(b,c)]$ (birleşme özelliği)

Yukarıdaki t1-t4 aksiyomlarını sağlayan bir fonksiyon $t:[0,1] \times [0,1] \rightarrow [0,1]$ t normu olarak adlandırılır.

2.6 DİLSEL DEĞİŞKENLER ve BULANIK EĞER-İSE KURALLARI :

Günlük hayatımızda kelimeler çoğunlukla değişkenleri ifade etmek için kullanılır. Örneğin, “Bugün hava çok sıcak !” veya “Bugün hava sıcaklığı yüksek!”. Burada hava sıcaklığı kelimesi yüksek değişkeniyle tanımlanmaktadır. Burada 19 °C veya 25 °C gibi rakamsal ifadeler kullanmakta mümkündür. Bir değişken rakamsal olarak ifade ediliyorsa matematiksel olarak formüle etmek daha kolay olacaktır. Fakat rakamsal değerlerin yerini kelimeler almaya başladığı zaman bu bilgiyi formüle etmek için klasik matematiksel kurallar yetersiz kalmaktadır.



Şekil 2.3 Sayısal değişkenler ve dilsel değişkenler

Tanım : Bir dilsel değişken (X,T,U,M) ile ifade edilir.

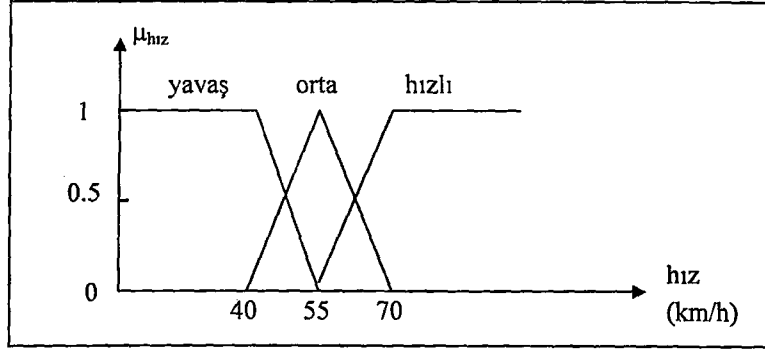
- X ; dilsel değişkenin ismi
- T ; dilsel değişken X 'in alabileceği değerler
- U ; X 'in değişken değerler alabileceği gerçek fiziksel domeni göstermektedir
- M ; bulanık küme U ile dilsel değişken değeri T 'yi birbirine bağlayan anlamsal kurallardır.

Örneğin bir arabanın hızını dilsel değişken olarak alalım. Dilsel değişkenin alabileceği değerler ;

$$T(\text{hız}) = \{\text{yavaş, orta, hızlı, çok yavaş, çok hızlı....}\}$$

olarak tanımlanabilmektedir.

$U=[0,100]$ arasında olmak üzere , dilsel değişken T için şu şekilde bir yorum yapabiliriz. 40 km/h 'e kadar olan hızlar için yavaş, 40 km/h ile 70 km/h arasındaki hızlar için orta hızlı, 70 km/h üzerindeki hızlar için hızlı tanımını yapabiliriz. Aşağıdaki şekilden görüldüğü üzere 40 km/h ile 70 km/h arasındaki hızlar için tam bir dilsel üyelik fonksiyonu yoktur. Şekil 2.4'te hız için tanımlanan bir üyelik fonksiyonu görülmektedir.



Şekil 2.4 Bir arabanın hız tanımını içeren bulanık üyelik fonksiyonu

Bir bulanık EĞER-İSE kuralı şu şekilde ifade edilir.

$$\text{EĞER } \langle \text{bulanık önerme} \rangle \text{ İSE } \langle \text{bulanık önerme} \rangle \quad (2.8)$$

Bulanık mantıkta basit ve birleşik önerme olmak üzere iki tip önerme vardır. Bir basit önerme “x A’dır.” şeklinde, x bir dilsel değişken ve A x’in bir dilsel değeri olmak üzere tek bir ifadedir. Bir birleşik önerme basit önermelerin çeşitli bağlaçlarla (ve, veya gibi) birleştirilmesiyle oluşturulan önermedir.

2.7 BULANIK EĞER-İSE KURALLARININ YORUMLANMASI

(2.8) ifadesini ‘EĞER <FP1> İSE <FP2>’ şeklinde ifade edelim. FP1, $U=U_1 \times U_2 \times \dots \times U_n$ ‘de ve FP2, $V=V_1 \times V_2 \times \dots \times V_m$ uzayında bir bulanık önerme olsun. x ve y U ve V’de dilsel değişkenler (vektörler) kabul edelim. Klasik mantıkta ‘EĞER p İSE q’ ifadesi $p \rightarrow q$ şeklinde gösterilir ve doğruluk tablosu, $\bar{p} \vee q$ ve $(p \wedge q) \vee \bar{p}$ ifadesinin doğruluk tablosuyla aynıdır. -, \vee ve \wedge operatörleri klasik ‘değil’, ‘ve’ ve ‘veya’ operatörleridir. p ve q yerine bulanık önermeler gelebileceği için bulanık EĞER-İSE kuralları -, \wedge ve \vee operatörleri yerine sırasıyla bulanık tümleme, bulanık birleşme ve bulanık kesişme konularak incelenmektedir. Bulanık tümlen, bulanık birleşme ve kesişme operatörleri farklı şekillerde gösterilebildiğinden dolayı literatürde çeşitli EĞER-İSE kural yorumları bulunmaktadır.

- **Dienes-Rescher Bağıntısı** : Eğer pVq ifadesinde lojik operatörler \neg ve V yerine bulanık tümleyen ve bulanık birleşme konulursa Dienes-Rescher bağıntısı olarak adlandırılır.

$$\mu_{Q_D}(x, y) = maks[1 - \mu_{FP_1}(x), \mu_{FP_2}(y)] \quad (2.9)$$

- **Lukasiewicz Bağıntısı** : Eğer V için s-normu kullanılırsa ve \neg için temel bulanık tümleyen kullanılırsa Lukasiewicz işlemi yapılır.

$$\mu_{Q_L}(x, y) = \min[1, 1 - \mu_{FP_1}(x) + \mu_{FP_2}(y)] \quad (2.10)$$

- **Zadeh Bağıntısı** : $(p \wedge q)Vp$ den yararlanarak yapılan işlemdir.

$$\mu_{Q_Z}(x, y) = maks[\min(\mu_{FP_1}(x), \mu_{FP_2}(y)), 1 - \mu_{FP_1}(x)] \quad (2.11)$$

- **Gödel Bağıntısı** : Gödel bağıntısı klasik lojikte en iyi bilinen bağıntıdır.

$$\mu_{Q_G}(x, y) = \begin{cases} 1 & , \text{eğer } \mu_{FP_1}(x) \leq \mu_{FP_2}(y) \\ \mu_{FP_2}(y) & , \text{aksi takdirde} \end{cases} \quad (2.12)$$

- **Mamdani Bağıntısı** : Bulanık sistemlerde ve bulanık kontrolde en çok kullanılan bağıntıdır. Mamdani EĞER-İSE kuralını Q_{MM} ve Q_{MP} bulanık bağıntı üyelik fonksiyonlarını şu şekilde ifade etmiştir.

$$\mu_{Q_{MM}}(x, y) = \min[\mu_{FP_1}(x), \mu_{FP_2}(y)] \quad (2.13)$$

veya

$$\mu_{Q_{MP}}(x, y) = \mu_{FP_1}(x)\mu_{FP_2}(y) \quad (2.14)$$

BÖLÜM 3

BULANIK KONTROLÖRLER

3.1 GİRİŞ

Bu bölümde bulanık kontrolörlerin genel yapısından bahsedilecek ve kontrolörü oluşturan birimler hakkında bilgi verilecektir.

Klasik kontrolörler matematiksel modeli bilinen süreçlere uygulanır ve kontrolör bu matematiksel modele göre tasarlanır. Bulanık kontrolörler ise matematiksel modeli tam olarak elde edilemeyen, çok parametrelili ve nonlineer süreçlere uygulanmaktadır. Son yıllardaki mikroişlemci teknolojisindeki gelişmeler bulanık kontrolörün artan bir ilgi ile birçok alanda kullanılmasına neden olmuştur. Bulanık kontrolörler uzman bir sistem olduğundan sürecin kontrolünde uzman bilgisine başvurulmaktadır. Uzman bilgisi dilsel EĞER-İSE kuralları haline getirilir ve bulanık kontrol algoritmasında kullanılır. Eğer uzman bilgisine ulaşılamıyorsa bulanık kural tabanını oluşturmak için; varsa matematiksel model, simülasyonlar, sürecin bulanık modeli ve çeşitli öğrenme algoritmaları kullanılır[1].

Daha önce de bahsedildiği gibi bir bulanık kontrolör dört ayrı kısımdan oluşmaktadır:

1. Bulanık Kural Tabanı :

Bulanık çıkarımda kullanılan dilsel EĞER-İSE kural tabanından oluşur.

2. Bulanık Çıkarım Motoru

Bulanık çıkarımda kurallar üzerinde bulanık mantık yürütülür ve bulanık kural tabanını kullanarak giriş ve çıkış uzayı arasında bir bağlantı kurar.

3. Bulandırıcı

Bu bölüm giriş değişkenlerini (gerçek değerleri) ölçer, onlar üzerinde bir ölçek değişikliği yapar ve bulanık kümelerle dönüştürür., yani onlara birer etiket vererek dilsel bir nitelik kazandırır.

4. Durulayıcı :

Çıkarım motorunun bulanık küme çıkışı (birden fazla çıkış olabilir) üzerinde gerekli ölçek değişikliklerini yapar ve bunları gerçek sayılara dönüştürür.

$$U = U_1 \times U_2 \times U_3 \times U_4 \times \dots \times U_n \subset R^n \text{ ve } V \subset R \quad (3.1)$$

Burada çok girişli ve tek çıkışlı bir sistem ele alınmaktadır. Çünkü çok girişli ve çok çıkışlı bir sistem daima çok girişli ve tek çıkışlı bir sisteme indirgenebilir.

3.2 BULANIK KURAL TABANI

3.2.1 Bulanık Kural Tabanının Yapısı

Bulanık kural tabanı, bulanık EĞER-İSE kurallarından oluşmaktadır. Bu kurallar bulanık sistemin temelini oluşturmaktadır.

$$\text{Kural}^l: \text{ EĞER } x_1=A_1^l \text{ ve } x_2=A_2^l \text{ ve } \dots \text{ ve } x_n=A_n^l \text{ İSE } y=B^l \quad (3.2)$$

A_i^l ve B^l bulanık kümeler ve $x=(x_1, x_2, \dots, x_n)^T \subset U$ ve $y \subset V$ giriş ve çıkış değişkenleridir. M , bulanık kural tabanındaki kural sayısı olsun. $l=1, 2, \dots, M$. Bu kurallar dizisi, farklı yapılardaki bulanık kural tiplerini ve bulanık önermeleri içerdiğinden *kanonik bulanık EĞER-İSE kuralları* olarak adlandırılır.

3.2.2 Bulanık Kural Tabanının Özellikleri

Bir bulanık kural tabanı, bir kurallar kümesi olduğundan dolayı kurallar arasındaki ilişkiler çeşitli soruları beraberinde getirmektedir. Örneğin, bulunan kurallar bulanık sistemin karşılaşılabileceği bütün durumları kapsayabilecek midir? Kurallar arasında herhangi bir çelişme durumu olabilir mi?

Tanım : Bulanık EĞER-İSE kurallar kümesi, eğer her $x \in U$ için kural tabanında en azından bir kural varsa bu küme tam kümedir.

$$\mu_{A_i}(x_i) \neq 0 \quad i=1,2,\dots,n \quad (3.3)$$

Sezgisel olarak, bir kurallar kümesinin tamlığının anlamı , giriş uzayındaki herhangi bir noktaya ait en azından bir kural vardır. Bunun anlamı kuralın EĞER kısmının üyelik değeri bu nokta için sıfırdan farklıdır.

Örnek : 2 girişli ve 1 çıkışlı bir sistemi ele alalım, $U = U_1 \times U_2 = [0,1] \times [0,1]$ ve $V=[0,1]$. U_1 'de S_1 , M_1 ve L_1 olmak üzere üç bulanık kümeyi ele alalım. S_2 ve L_2 'de U_2 'de bulanık kümeler olsun. Bu bulanık kural tabanının tam olması için S_1 , M_1 , L_1 'in S_2 , L_2 ile mümkün olan bütün kombinasyonlarını içeren altı kuralı içermesi gerekir.

$$\text{EĞER } x_1 = S_1 \text{ ve } x_2 = S_2 \text{ İSE } y = B^1$$

$$\text{EĞER } x_1 = S_1 \text{ ve } x_2 = L_2 \text{ İSE } y = B^2$$

$$\text{EĞER } x_1 = M_1 \text{ ve } x_2 = S_2 \text{ İSE } y = B^3$$

$$\text{EĞER } x_1 = M_1 \text{ ve } x_2 = L_2 \text{ İSE } y = B^4$$

$$\text{EĞER } x_1 = L_1 \text{ ve } x_2 = S_2 \text{ İSE } y = B^5$$

$$\text{EĞER } x_1 = L_1 \text{ ve } x_2 = L_2 \text{ İSE } y = B^6$$

Eğer bu grupta herhangi bir kural eksikse $x^* \in U$ noktası için bütün kalan kuralların EĞER kısmı sıfır üyelik değerine sahip olacaktır.

Tanım : Bir EĞER-İSE kurallar kümesinde, eğer aynı EĞER ifadesi için, İSE kısmı farklı değerler almıyorsa bu küme mevcuttur.

Bulanık olmayan kurallar için mevcut olma durumu gerek koşuldur. Bulanık kurallar için mevcut olma durumu o kadar önemli değildir. Bulanık çıkarım, bulanıklaştırma ve durulaştırma yöntemleriyle istenen sonuç üretilebilmektedir.

Tanım : Bir EĞER-İSE bulanık kurallar kümesi, İSE kısmına ait bulanık kümelerinin kesişimi boş küme olan birbirine çok yakın kurallar yoksa süreklidir. Yani bunun anlamı bir bulanık sistemin giriş-çıkış davranışı düzgün olmalıdır.

3.3 BULANIK ÇIKARIM

Bulanık çıkarımda; bulanık mantık işlemleriyle, bulanık kural tabanındaki EĞER-İSE kurallarını birleştirerek giriş ve çıkış bulanık uzayı arasında bir bağıntı oluşturulur.

Çoğu pratik sistemde bulanık kural tabanı birçok kuraldan oluşur. Bu kurallarla bir bulanık çıkarım yapmak için iki yöntem vardır. Bu yöntemler birleşik tabanlı çıkarım ve tek kural tabanlı çıkarım yöntemleridir.

3.3.1 Birleşik Tabanlı Çıkarım

Birleşik tabanlı çıkarımda, bulanık kural tabanındaki bütün kurallar tek bir bulanık bağıntıyla birleştirilerek $U \times V$ uzayındaki tek bir bulanık üyeliğe indirgenir. Bu birleştirme uygun mantıksal operatörler kullanarak yapılır. Bu operatörler birleşme ve kesişme operatörleri olabilir. Birbirinden bağımsız koşullar içeren kurallar varsa en iyi yöntem birleşme özelliğini kullanmaktır, ancak birbirine bağlı koşullar içeren kurallar varsa bu durumda kesişme özelliği kullanılmalıdır.

Ru^{ϕ} EĞER-İSE kuralını gösteren $U \times V$ uzayında $Ru^{\phi} = A_1^I \times A_2^I \times \dots \times A_n^I \rightarrow B^I$ bir bulanık bağıntı olsun. $A_1^I \times A_2^I \times \dots \times A_n^I \cup U = U_1 \times U_2 \times \dots \times U_n$ uzayında aşağıdaki ifade tarafından belirlenen bir bulanık bağıntıdır.

$$\mu_{A_1^I \times \dots \times A_n^I}(x_1, x_2, \dots, x_n) = \mu_{A_1^I}(x_1) * \dots * \mu_{A_n^I}(x_n) \quad (3.4)$$

Birbirinden bağımsız kurallar olması halinde M tane kuraldan $U \times V$ uzayında tek bir Q_M bulanık bağıntısı çıkarılır.

$$Q_M = \bigcup_{l=1}^M Ru^{(l)} \quad \text{ya da} \quad \mu_{Q_M}(x, y) = \mu_{Ru}^{(1)}(x_1) + \dots + \mu_{Ru}^{(M)}(x_n) \quad (3.5)$$

Bu kombinasyon *Mamdani kombinasyonu* olarak adlandırılır. $+$ sembolü s normu için kullanılmaktadır.

Birbirine bağlı koşulların bulunduğu kurallarda, Q_G , M tane bulanık EĞER-İSE kuralından $U \times V$ uzayında bir bulanık bağıntı olarak tanımlanırsa

$$Q_G = \bigcap_{l=1}^M Ru^{(l)} \quad \text{ya da} \quad \mu_{Q_G}(x, y) = \mu_{Ru}^{(1)}(x_1) * \dots * \mu_{Ru}^{(M)}(x_n) \quad (3.6)$$

Bu kombinasyon ise *Gödel kombinasyonu* olarak adlandırılır. $*$ operatörü t -normunu göstermektedir.

Q_M ve Q_G tek bir bulanık kural olmak üzere genelleştirilmiş modus ponensi kullanarak bulanık çıkarımın çıkışını Mamdani kombinasyonu ile şu şekilde elde ederiz.

$$\mu_B(y) = \sup t[\mu_{A^I}(x), \mu_{Q_M}(x, y)] \quad (3.7)$$

Gödel kombinasyonu ;

$$\mu_{B'}(y) = \sup t[\mu_{A'}(x), \mu_{QG}(x,y)] \quad (3.8)$$

3.3.2 Tek Kural Tabanlı Çıkarım

Tek kural tabanlı çıkarımda, kural tabanındaki her bir kural tek bir çıkış bulanık kümesi tanımlar ve tüm bulanık çıkarımın çıkışı M adet bulanık kümenin kombinasyonundan oluşur. Kombinasyon birleşme veya kesişme özellikleri kullanılarak alınabilir.

$$\mu_{B'}(y) = \sup t[\mu_{A'}(x), \mu_{Ru(l)}(x,y)] \quad (3.9)$$

$$\mu_{B'}(y) = \mu_{B1'}(y) + \mu_{B2'}(y) + \dots + \mu_{BM'}(y) \quad (3.10)$$

$$\mu_{B'}(y) = \mu_{B1'}(y) * \mu_{B2'}(y) * \dots * \mu_{BM'}(y) \quad (3.11)$$

3.3.3 En Çok Kullanılan Çıkarım Motorları

- **Çarpım Çıkarım Motoru :** Bu çıkarım motorunda kullanılan
 1. Birleşme kombinasyonu ile tek kural tabanlı çıkarım
 2. Mamdani çarpım bağıntısı
 3. Tüm t-normu operatörleri için cebirsel çarpım ve s-normu operatörleri için maksimum

$$\mu_{B'}(y) = \max_{l=1}^M [\sup (\mu_{A'}(x) \prod_{i=1}^n \mu_{A_i^l}(x_i) \mu_{B^l}(y))] \quad (3.12)$$

- **Minimum Çıkarım Motoru :**
 1. Birleşme kombinasyonu ile tek kural tabanlı çıkarım
 2. Mamdani minimum bağıntısı
 3. Tüm t-normu operatörleri için minimum ve s-normu için maksimum

$$\mu_{B'}(y) = \max_{l=1}^M [\sup \min(\mu_{A'}(x), \mu_{A_1^l}(x_1), \mu_{A_2^l}(x_2), \mu_{B^l}(y))] \quad (3.13)$$

Çarpım ve minimum çıkarım motorları bulanık sistemlerde ve kontrolde en çok kullanılan çıkarım motorlarıdır. Bu yöntemlerin en önemli avantajları hesaplama ve programlama basitlikleridir. Dezavantajı ise, eğer bazı $x \in U$ noktalarında $\mu_{A_i'}(x_i)$ çok küçükse hesaplanan $\mu_{B'}(y)$ çok küçük olacaktır.

- **Lukasiewicz Çıkarım Motoru :**

1. Kesişme kombinasyonu ile tek kural tabanlı çıkarım
2. Lukasiewicz bağıntısı
3. Tüm t-normu operatörleri için minimum

$$\begin{aligned} \mu_{B'}(y) &= \min_{l=1}^M [\sup \min(\mu_{A'}(x), \mu_{Ru^l}(x, y))] \\ \mu_{B'}(y) &= \min_{l=1}^M \{ \sup \min[\mu_{A'}(x), \min(1, 1 - \min_{i=1}^n (\mu_{A_i^l}(x_i)) + \mu_{B^l}(y))] \} \\ \mu_{B'}(y) &= \min_{l=1}^M \{ \sup \min[\mu_{A'}(x), 1 - \min_{i=1}^n (\mu_{A_i^l}(x_i)) + \mu_{B^l}(y)] \} \end{aligned} \quad (3.14)$$

- **Zadeh Çıkarım Motoru :**

Lukasiewicz çıkarım motorundan farkı Zadeh bağıntısını kullanmasıdır.

$$\begin{aligned} \mu_{B'}(y) &= \min_{l=1}^M \{ \sup \min[\mu_{A'}(x), \max(\min(\mu_{A_1^l}(x_1), \dots, \mu_{A_n^l}(x_n)), \mu_{B^l}(y)), \\ &\quad 1 - \min_{i=1}^n (\mu_{A_i^l}(x_i))] \} \end{aligned} \quad (3.15)$$

- **Dienes-Rescher Çıkarım Motoru :**

Lukasiewicz çıkarım motorundan farkı Dienes-Rescher bağıntısını kullanmasıdır.

$$\mu_{B'}(y) = \min_{l=1}^M \{ \sup \min [\mu_{A'}(x), \max(1 - \min_{i=1}^n (\mu_{A_i^l}(x_i), \mu_{B^l}(y)))] \} \quad (3.16)$$

3.4 BULANDIRICI

Bulandırıcı; $x^* \in U \subset R^n$ gerçek değerli noktadan U uzayındaki A' bulanık kümesine bir fonksiyondur. Bulandırıcıyı dizayn ederken şu kriterlere dikkat edilmelidir. İlk olarak, bulanık küme A' x^* değerindeki geçişler gözönüne alınmalıdır. İkinci olarak, bulanık kontrolöre alınan giriş gürültü nedeniyle bozulursa, bulandırıcı gürültüyü azaltacak yapıda olmalıdır. Üçüncü olarak; bulandırıcı, bulanık çıkarım motorunda yapılacak hesaplamaları basitleştirmeye yardımcı olmalıdır.

Burada üç farklı bulandırıcıyı ele alacağız

3.4.1 Tek Noktalı Bulandırıcı

Tek noktalı bulandırıcı üyelik değeri, x^* 'da 1 ve U uzayındaki diğer bütün değerlerde 0'dır.

$$\mu_{A'}(x) = \begin{cases} 1 & , x=x^* \text{ ise} \\ 0 & , x \neq x^* \text{ ise} \end{cases} \quad (3.17)$$

3.4.2 Gaussian Bulandırıcı

U uzayındaki A' bulanık kümesinin Gaussian üyelik fonksiyonu ;

$$\mu_{A'}(x) = e^{-\left(\frac{x_1 - x_1^*}{a_1}\right)^2} * \dots * e^{-\left(\frac{x_n - x_n^*}{a_n}\right)^2} \quad (3.18)$$

a_i pozitif parametreleri ve * t-normunu göstermektedir.

3.4.3 Üçgen Bulandırıcı

$$\mu_{A^*}(x) = \begin{cases} \left(1 - \frac{|x_1 - x_1^*|}{b_1}\right) * \dots * \left(1 - \frac{|x_n - x_n^*|}{b_n}\right) & , \text{ eğer } |x_1 - x_1^*| \leq b_i, i = 1, 2, \dots, n \\ 0 & , \text{ aksi takdirde} \end{cases} \quad (3.19)$$

bi pozitif parametreleri ve * t-normunu göstermektedir.

3.5 DURULAYICI

Durulayıcı $V \subset \mathbb{R}$ uzayında B' bulanık kümesinin $y^* \in V$ geçiş noktasına dağılımıdır. Kavramsal olarak durulayıcının görevi bulanık küme B' 'yi V uzayında en iyi temsil eden noktayı belirlemektir. Bu noktayı belirlerken farklı yöntemler kullanılabilir. Aşağıdaki üç kriter durulayıcı seçimi yapılırken gözönünde bulundurulmalıdır.

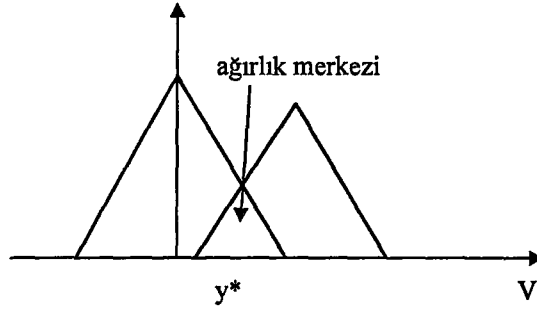
- **Uygun Olması (Plausibility):** y^* noktası sezgisel olarak B' bulanık kümesini temsil etmelidir.
- **Hesaplama Kolaylığı :** Bu kriter özellikle gerçek zamanda çalışan bulanık kontrolörler için önemlidir.
- **Süreklilik :** B' bulanık kümesindeki küçük bir değişiklik y^* 'da çok büyük bir değişikliğe neden olmamalıdır.

3.5.1 Ağırlık Merkezli Durulayıcı

Durulayıcının ağırlık merkezi, B' bulanık kümesinin üyelik fonksiyonu tarafından kapsanan alanın merkezi olarak y^* 'yü temsil etmektedir.

$$y^* = \frac{\int_V y \mu_{B'}(y) dy}{\int_V \mu_{B'}(y) dy} \quad (3.20)$$

Bu işlemi grafiksel olarak şekil 3.1 'de görmekteyiz.



Şekil 3.1 Ağırlık merkezli durulayıcı

Eğer $\mu_{B'}(y)$ rasgele değişkenin olabilirlik yoğunluk fonksiyonu ise durulayıcının ağırlık merkezi rasgele değişkenin ortalama değerini vermektedir. Bazen $y \in V$ için B' kümesindeki üyelik değerleri çok küçük ise bu indeksli durulayıcı ağırlık merkezini verir.

$$y^* = \frac{\int y \mu_{B'}(y) dy}{\int_{V\alpha} \mu_{B'}(y) dy} \quad V_\alpha = \{y \in V \mid \mu_{B'}(y) \geq \alpha\} \quad \alpha: \text{sabit} \quad (3.21)$$

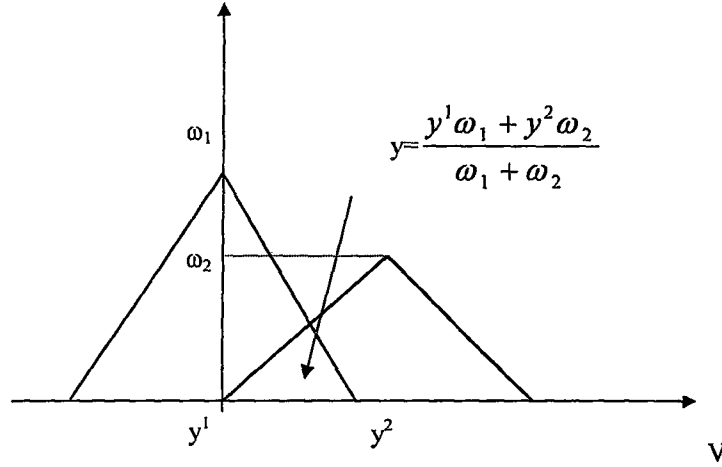
Ağırlık merkezli durulayıcının avantajı uygunluğudur. Dezavantajı ise hesaplama açısından zor olmasıdır. Çoğunlukla $\mu_{B'}(y)$ üyelik fonksiyonu kolay hesaplanabilir yapıda olmadığından dolayı integrasyon hesaplamalarını yapmak oldukça zordur.

3.5.2 Ortalama Merkezli Durulayıcı

M tane bulanık küme için y^l l. bulanık kümenin merkezi ve ω_l kümenin yüksekliği olmak üzere ortalama merkezli durulayıcı y^* şu şekilde gösterilir.

$$y^* = \frac{\sum_{l=1}^M y^l \omega_l}{\sum_{l=1}^M \omega_l} \quad (3.22)$$

M=2 için şekil 3.2 'de grafiksel olarak bir örnek verilmektedir.



Şekil 3.2 Ortalama merkezli durulayıcı

Ortalama merkezli durulayıcı hesaplama kolaylığı açısından ve diğer açılardan bulanık sistemlerde ve kontrolde en çok kullanılan durulama yöntemidir.

3.5.3 Maksimum Durulayıcı

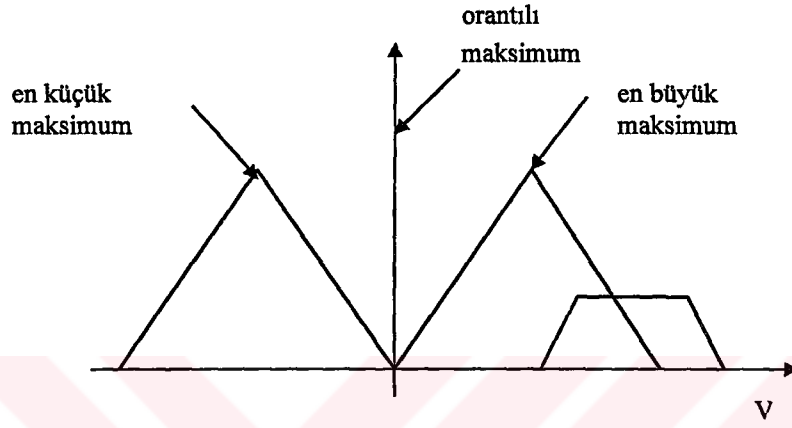
Maksimum durulayıcı kavramsal olarak, V uzayındaki y^* noktasını $\mu_{B'}(y)$ üyelik fonksiyonunun maksimum değerini aldığı nokta olarak seçilir. Maksimum durulayıcı hesaplama açısından basittir. Ancak en küçük bir değişimde y^* değeri büyük değişiklikler gösterebilir.

$$\text{hgt}(B') = \{y \in V \mid \mu_{B'}(y) = \sup \mu_{B'}(y)\} \quad (3.23)$$

$\text{hgt}(B')$ V uzayında $\mu_{B'}(y)$ 'nin maksimum değerlerini aldığı noktalar kümesidir. Maksimum durulayıcı $y^* \in \text{hgt}(B')$ kümesinde herhangi bir elemandır. Şekil 3.3.'de verildiği gibi en küçük maksimum durulayıcı, en büyük maksimum durulayıcı ve orantılı maksimum durulayıcı olmak üzere 3 farklı yöntemle durulama işlemi yapılabilir. Orantılı maksimum durulayıcı şu şekilde verilmektedir.

$$y^* = \frac{\int y dy}{\int dy} \quad (3.24)$$

Maksimum durulayıcı hesaplama açısından basit bir durulama yöntemidir.



Şekil 3.3 Maksimum durulayıcı

BÖLÜM 4

BULANIK KONTROLÖR TASARLANMASI

4.1 GİRİŞ

Bu bölümde bulanık kontrolör yapıları ve bulanık kontrolör tasarımı üzerinde durulacaktır.

Bulanık kontrolörler, klasik kontrolörlere göre çok daha yeni olmasına karşın giderek artan sayıda uygulamada kullanılmaktadır. Bulanık kontrolörler klasik kontrolörlerden esinlenerek bulanık PD, bulanık PI, bulanık PID ve karma kontrolörler şeklinde farklı yapılarda tasarlanmaktadır.

Bulanık kontrolörü ve klasik kontrolörü tasarlarken sistemin kararlılığı, performansı gibi aynı kontrol problemleri çözülmeye çalışılır. Burada temelde şu farklılık vardır. Klasik kontrolörler sürecin matematiksel modeline göre tasarlanır. Bulanık kontrolörler ise uzman insan bilgisinden yararlanarak elde edilen EĞER-İSE kuralları sentez edilerek tasarlanır[1].

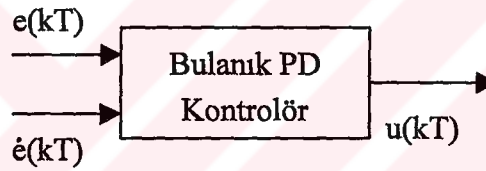
Bulanık kontrolör tasarlarken uygulanan yöntemler temelde iki kategoriye ayrılır. İlk yöntem deneme yanılma yöntemi, ikincisi ise teorik yöntemlerdir. Deneme yanılma ile veya eldeki uzman bilgilerini kullanarak sistem hakkında çeşitli EĞER-İSE kuralları oluşturulur ve sonra gerçek sistem test edilir. İstenen performans sağlanmıyorsa kurallar üzerinde değişiklikler yapılır, ince ayar kuralları oluşturulur ve sistem istenen performansı sağlayıncaya kadar test edilir. Teorik yaklaşımda ise bulanık kontrolör yapısı ve parametreleri istenen performansı garanti edecek şekilde

tasarlanır. Pratikte en iyi performansı sağlayan kontrolörü tasarlamak için iki yöntemi beraber kullanmak mümkündür[1].

4.2 BULANIK KONTROLÖR YAPILARI

4.2.1 Bulanık PD Kontrolör

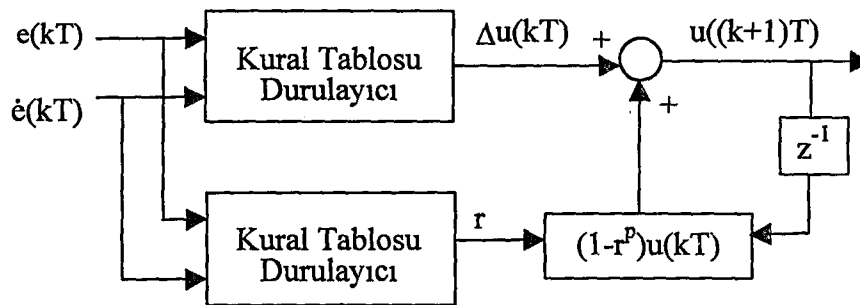
Bulanık PD kontrolörde; hata (e) ve hatanın türevine (\dot{e}) göre bir kontrol işareti (u) üretilir(Şekil 4.1). Burada bulanık önerme “*EĞER e pozitif büyük ve \dot{e} pozitif büyük İSE u pozitif büyüktür.*” şeklindedir. Bulanık PD kontrolör, sistem tipi 0’dan büyükse yani sistemin sürekli hal hatası sıfırsa uygundur. Bulanık PD kontrolör pozisyon tipi bulanık kontrolördür [4].



Şekil 4.1 Bulanık PD kontrolör yapısı

4.2.2 Genelleştirilmiş Bulanık PI Kontrolör

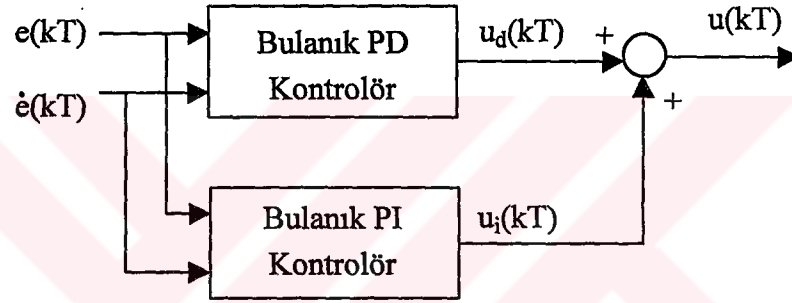
Bulanık PI kontrolörde hata (e) ve hatanın türevine (\dot{e}) göre bir kontrol işareti (Δu) üretilir(Şekil 4.2). Bulanık PI kontrolör, hız tipi bulanık kontrolördür [4].



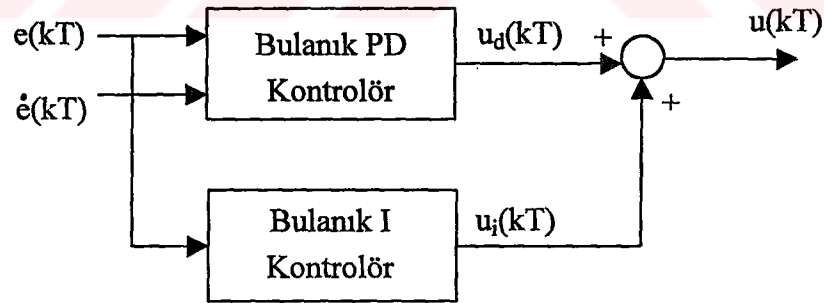
Şekil 4.2 Genelleştirilmiş bulanık PI kontrolör

4.2.3 Bulanık PID Kontrolör

PID kontrolör; modern kontrol teorisinin ortaya çıkmasından sonra en çok kullanılan kontrolör olmuştur. Parametre ayarı iyi yapılmış bir PID kontrolörden oldukça iyi performans elde edildiği bilinmektedir[1]. Bulanık PID kontrolör için farklı yapılar önerilmiştir. Bu yapılardan ilki bulanık PD ve bulanık PI kontrolörlerinin paralel çalıştırılmasıdır(Şekil 4.3). Her iki kontrolörün çıkışı toplanarak esas kontrol işareti bulunur. Diğer bir yapıda bulanık PD ve bulanık I kontrolörünün paralel kullanılmasıdır(Şekil 4.4).



Şekil 4.3 Bulanık PD+ Bulanık PI kontrolör yapısı

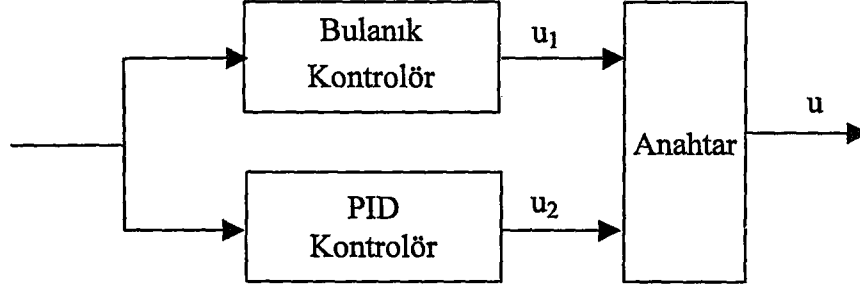


Şekil 4.4 Bulanık PD+ Bulanık I kontrolör yapısı

4.2.4 Karma Kontrolörler

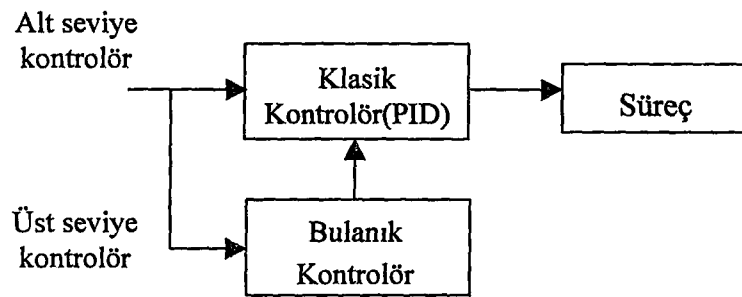
Kompleks yapıdaki gerçek sistemler için tek kontrolör kontrol hedeflerine ulaşmada yetersiz kalabilir ya da bunun için çok karmaşık bir kontrolör tasarımı gerekir. Bu nedenle pratikte iki seviyeli kontrolörler tasarlanır. Genellikle alt seviyeli

kontrolörler yüksek hızlı bölgede, yüksek seviyeli kontrolörler ise düşük hız bölgesinde etkin olur.

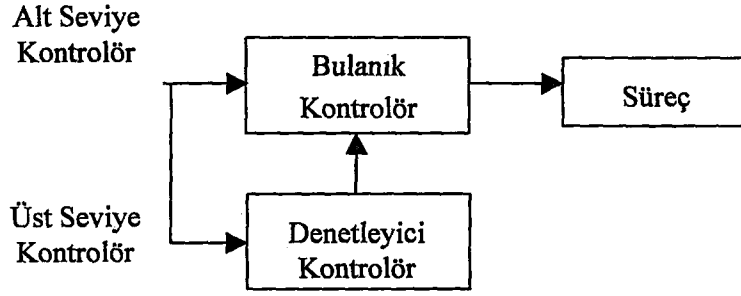


Şekil 4.5 Karma kontrolör yapısı

Şekil 4.5'te görülen karma kontrolörde PID ve bulanık kontrolör beraber kullanılmıştır. Bu yapıda bulanık algoritmanın oluşturulmasında daha az üyelik fonksiyonuna ve daha küçük bir kural tabanına ihtiyaç duyulur. Tek başına bulanık algoritma kullanan bir kontrolöre göre daha küçük bir hafıza ve daha basit işlemci kullanma avantajı vardır. Burada anahtar elemanın görevi u işaretini u_1 ve u_2 'den faydalanarak oluşturmaktır. Bulanık kontrolör nonlinear bir PD kontrolör gibi görev yaparken PID'nin integral terimi sürekli hal hatasını sıfırlar. Sistem referans değere yakinken PID kontrolörün, uzakken bulanık kontrolörün etkisindedir. Şekil 4.6a ve 4.6b'de iki seviyeli bulanık kontrolörler gösterilmiştir.



Şekil 4.6a İki seviyeli bir bulanık kontrol sisteminin yapısı



Şekil 4.6b İki seviyeli bir bulanık kontrol sisteminin yapısı

4.3 GİRİŞ-ÇIKIŞ BİLGİLERİNİ KULLANARAK BULANIK KONTROLÖR TASARIMI

Bulanık sistemler uzman insan bilgisini formüle etmek için kullanılmaktadırlar. Burada öncelikli olarak uzman insan bilgisinin nasıl elde edileceği bilinmelidir. Uzman insan sistem hakkındaki bilgisini EĞER-İSE kurallarıyla ifade edebilirse bu kurallar bulanık sistemin oluşturulmasında kullanılabilir. Ancak bu kurallar elde edilemiyorsa, sistemin tipik durumları için denemelerle çeşitli giriş çıkış çiftleri elde edilerek bulanık sistem dizayn edilir. Ancak birçok pratik uygulamada elde edilen giriş-çıkış çiftleri sınırlıdır ve bundan dolayı rasgele bir giriş için çıkış değeri elde edilemeyebilir. Bu durumda yapılması gereken eldeki giriş-çıkış çiftleriyle giriş-çıkış davranışını karakterize eden bir bulanık kontrolör tasarlamaktır.

4.3.1 Look-Up Tablosu Kullanarak Bulanık Kontrolör Tasarımı

$$(x_0^p; y_0^p), p=1,2,\dots,N \quad (4.1)$$

$$x_0^p \in U=[\alpha_1, \beta_1] \times \dots \times [\alpha_n, \beta_n] \subset \mathbb{R}^n \text{ ve } y_0^p \in V=[\alpha_y, \beta_y] \subset \mathbb{R} \quad (4.2)$$

1'den N'e kadar N adet giriş-çıkış çiftine sahip olduğumuzu kabul edelim Bu çiftleri kullanarak bulanık sistemi kurmak için look-up tablosu oluşturmak amacıyla bir yöntem sunulmaktadır[1].

1.Adım : Giriş ve çıkış uzayını kapsayacak bir bulanık küme tanımla

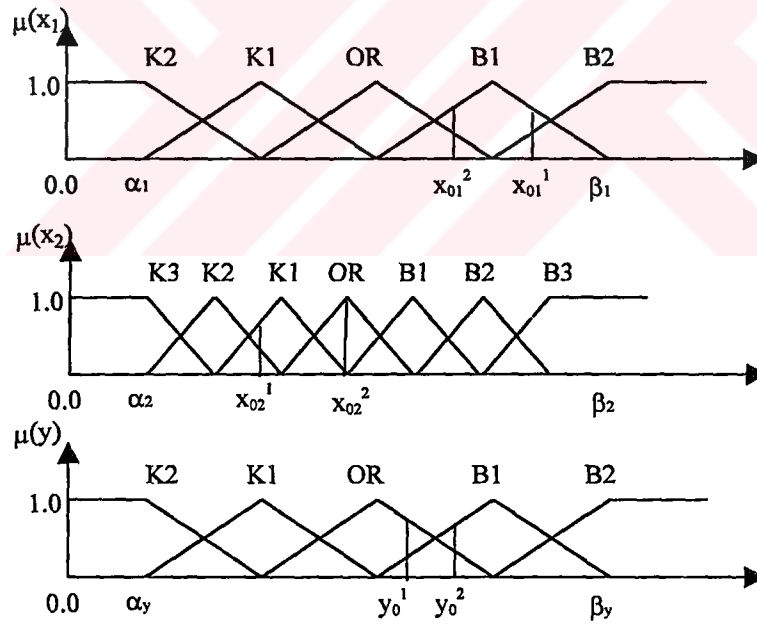
N_i adet A_i^j ($j=1,2,\dots,N_i$, $i=1,2,\dots,n$) bulanık küme tanımlansın. Her A_i^j için üyelik fonksiyonu $\mu_{A_i^j}(x_i) \neq 0$ olması gerekmektedir. Burada üyelik fonksiyonları üçgen, yamuk veya diğer şekillerde seçilebilir.

2.Adım : Her bir giriş ve çıkış çiftinden bir kural oluştur

Üyelik fonksiyonlarını oluşturduktan sonra her bir giriş-çıkış çiftinin en büyük üyelik değerine sahip olduğu bulanık küme belirlenir. Şekildeki gibi iki adet giriş-çıkış çiftine sahip bir örnek için şu kuralları oluşturabiliriz.

Eğer x_1 B1 ve x_2 K1 ise y OR

Eğer x_1 B1 ve x_2 OR ise y B1



Şekil 4.7 İki farklı giriş durumu için giriş-çıkış çiftleri ve üyelik fonksiyonları

3.Adım : Belirlenen her kural için bir derece belirle

Gerçekte giriş-çıkış çifti sayısı oldukça fazla olduğundan çok fazla sayıda kural elde edilmektedir. Çok sayıda kural elde edilmesi halinde çelişen yani aynı EĞER

kısmına fakat farklı İSE kısmına sahip kurallar bulunması olasılığı doğmaktadır. Bu çelişmeyi engellemek için üretilen her kurala bir derece atanmaktadır.

$(x_0^p; y_0^p)$ giriş-çıkış çiftinin üyelik derecelerinden

$$D(\text{kural}) = \prod_{i=1}^n \mu_{A_i^j}(x_{0i}^p) \mu_{B^l}(y_0^p) \quad (4.3)$$

Şekil 4.7’de verilen; $x_{01}^1, x_{02}^1, y_0^1$ için $D(\text{kural1}) = \mu_{B1}(x_0^1) \mu_{K1}(x_0^1) \mu_{OR}(x_0^1)$
 $x_{01}^2, x_{02}^2, y_0^2$ için $D(\text{kural2}) = \mu_{B1}(x_0^1) \mu_{OR}(x_0^1) \mu_{B1}(x_0^1)$

4.Adım : Bulanık kural tabanını oluştur

Bulanık kural tabanı şu üç kurallar kümesini içermektedir.

- 2. adımda üretilen ve birbiriyle çelişmeyen kurallar
- Birbiriyle çelişen kurallardan en yüksek üyelik derecesine sahip kurallar
- Uzmanın dilsel olarak belirlediği kurallar

Elde edilen kuralları ve giriş-çıkış için belirlenen bulanık kümeleri kullanarak bir look-up tablosu oluşturulur.

X2 \ X1	K2	K1	OR	B1	B2
K3					
K2					
K1					
OR					
B1					
B2					
B3					

Şekil 4.8 Bulanık look-up kural tablosu

5. Adım: Bulanık kural tabanına göre bulanık sistem tasarlama

Bulanık kural tabanını oluşturduktan sonra uygun bulanık çıkarım motoru, bulandırıcı ve durulayıcıyı seçerek bulanık sistem oluşturulur.

BÖLÜM 5

SİMÜLASYONLAR

5.1 DİJİTAL PID KONTROLÖR TASARIMI VE SİMÜLASYONU

$G_s(s)=1/s(s+3.6)$ transfer fonksiyonuna sahip servomotor kontrolü için dijital PID kontrolör tasarlanacaktır. En çok kullanılan ayırık PID modeli T_s örnekleme periyodu olmak üzere şu şekildedir.

$$D(z) = K_P + K_D \frac{(z - 1)}{z T_s} + K_I \frac{z T_s}{(z - 1)} \quad (5.1)$$

PID kontrolörde; sistemin sürekli hal cevabını iyileştiren integral kontrolör sistem tipini yükseltmektedir. Türev kontrolörü ise sönüm oranını yükseltmekte ve sistemin kararlılığını artırmaktadır.

$G_s(s)=1/s(s+3.6)$ transfer fonksiyonuna sahip sistemin ayırık modeli sıfırıncı dereceden tutucu kullanılarak, $T_s=0.01$ saniye için

$$G_{ho}G_s(z)=10^{-4} \frac{0.4941 z + 0.4882}{z - 1) (z - 0.9646)} \quad \text{olarak bulunur.} \quad (5.2)$$

5.1.1 Ziegler Nichols Metodu

Ziegler-Nichols metodu sürekli sistemler için kullanılabileceği gibi ayırık sistemler içinde kullanılmaktadır. Ayırık sistemlerde örnekleme zamanı kontrolör formülasyonu içine girmektedir[5].

Ziegler-Nichols yönteminde, yapısında integratör bulunan sistem için ilk olarak $K_D=K_I=0$ olarak alınır ve oransal kazanç değeri artırılarak sistemin osilasyon yapmaya başladığı kazanç değeri saptanır. Bu kazanç değeri; s düzleminde köklerin yer eğrisinin imajiner eksenini kestiği kritik K_m değeridir. z-düzleminde ise birim daireyi kesen kapalı çevrim kutuplarından yararlanarak bulunur. Belirlenen bu kritik kazanç değeri K_m ve osilasyon frekansı ω_m olmak üzere (5.3)'de verilen eşitliklerden yararlanarak PID kontrolör parametreleri bulunur[6].

$$K_p=0.6 K_m \quad K_D=\frac{K_p \pi}{4\omega_m} \quad K_I=\frac{K_p \omega_m}{\pi} \quad (5.3)$$

Ele alınan sistem için s düzleminde köklerin yer eğrisi imajiner eksenini kesmediğinden dolayı kritik bir K_m kazancı bulunamaz ve dolayısıyla Ziegler-Nichols metodu uygulanamaz. z-domenine geçildiğinde ise birim daireyi kesen kritik bir kazanç değeri saptanabilmektedir..

K_m ve ω_m değerlerini saptamak için MatLab'de yazılan aşağıdaki programdan yararlanılmıştır.

```
% MatLab Program
% z Domeninde Zeigler-Nichols Metoduyla PID Parametrelerini Belirleme
ng=[1]; dg=[1 3.6 0];
% örnekleme periyodu
Ts=0.01;
[ng_z,dg_z]=c2dm(ng,dg,Ts)
axis('square'), zgrid('new'), rlocus(ng_z,dg_z)
[Km,pole]=rlocfind(ng_z,dg_z)
wm=angle(pole(1))/Ts
Kp=0.6*Km
Kd=Kp*pi/(4*wm)
Ki=Kp*wm/pi
% PID kontrolör transfer fonksiyonu
```



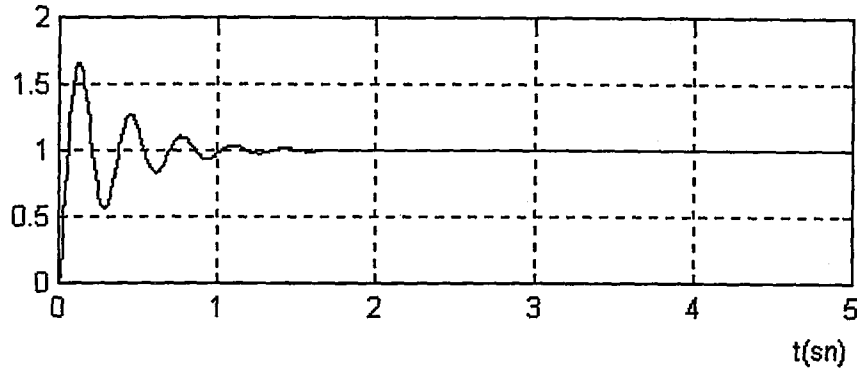
```

if Ki==0
    nk_PID=[Kp*Ts+Kd -Kd]; dk_PID=[Ts 0]
else
    nk_PID=[Kp*Ts+Kd+Ki*Ts*Ts -Kp*Ts-2*Kd Kd]; dk_PID=[Ts -Ts 0]
end
[ngk,dgk]=series(nk_PID,dk_PID,ng_z,dg_z);
[nt,dt]=cloop(ngk,dgk);
% kapalı çevrim kutuplarını, söndürme oranını ve açısal frekansları bul
ddamp(dt,Ts)
subplot(2,1,1); axis('square'), zgrid('new'), rlocus(ngk,dgk);
subplot(2,1,2);
x=[1 ones(1,20)]; v=[0 20 0 2]; axis(v);
i=[0:100]; time=i*Ts; g_zoh=dstep(nt,dt,101);
% birim basamak giriş cevabı
stairs(time,g_zoh);
end.

```

$K_m=787.5$ ve $\omega_m=27.88$ olarak bulunur. (5.2)'den yararlanarak $K_p=472.5$, $K_D=13.3$ ve $K_I=4.19 \times 10^3$ olarak hesaplanır. Bulunan PID parametreleri, örnekleme periyodu küçük alındığından dolayı çok yüksek değerler olarak bulunmuştur. Şekil 5.1'de belirlenen parametrelere göre sistem cevabı görülmektedir. Bu sistem cevabı gerilim sınırlamasız olarak elde edilmiştir. Sistem cevabında aşım %60'ın üstünde olduğu için uygulanan yöntemin iyi sonuç vermediğine karar verebiliriz.

Kutuplar	ω	ξ
$0.95+j0.21$	21.9	0.14
$0.95-j0.21$	21.9	0.14
0.9	10.3	1.00
0.08	257.0	1.00



Şekil 5.1 $T_s=0.01$ sn için Ziegler-Nichols yöntemine göre hesaplanan kontrolör parametrelerine göre birim basamak sistem cevabı

5.1.2 Analitik Olarak Dijital PID Kontrolör Tasarımı ve Simülasyonu

Endüstride PID parametrelerini belirlemek için sıkça kullanılan Ziegler-Nichols yöntemi istenen kapalı çevrim performansını sağlamakta yetersiz kalmaktadır. Analitik yöntemle istenen kapalı çevrim hedeflerini sağlayan PID parametreleri belirlenebilmektedir. Analitik yöntemde istenen ξ ve ω_n değerlerini dolayısıyla aşım ve yükselme zamanını verecek bir kontrolör tasarımı yapılmaktadır. İlk olarak, istenen sürekli hal hatasını sağlayan bir K_I katsayısı belirlenir[9].

Birim basamak giriş için sürekli hal konum hatası (5.4)'de verilen eşitlikten bulunur.

$$e_{ss}^* = \frac{1}{1 + \lim_{z \rightarrow 1} GH(z)} \quad (5.4)$$

ξ ve ω_n değerlerinden kapalı çevrim kutbu belirlenir. $s_1 = -\xi\omega_n + j\omega_n(1-\xi^2)^{1/2}$ kapalı çevrim kutbunu veren K_P ve K_D parametreleri bulunur. Kapalı çevrim kutbu köklerin yer eğrisi üzerinde olduğundan dolayı karakteristik denklemi sağlamaktadır. s_1 kutbunun z domenindeki karşılığı z_1 olsun.

$$\left(K_P + K_D \frac{(z_1 - 1)}{z_1 T_s} + K_I \frac{z_1 T_s}{(z_1 - 1)}\right) G_{h0} G_s(z) = -1 \quad (5.5)$$

İncelediğimiz sistem birinci tip bir sistem olduğundan dolayı sürekli hal hatası zaten sıfırdır. Dolayısıyla integratörü kaldırarak PD kontrolörü kullanmak yeterli olacaktır.

$$D(z) = K_P + K_D \frac{(z - 1)}{z T_s} \quad (5.6)$$

PD kontrolörden 0'a bir kutup ve $K_D/(K_P T_s + K_D)$ noktasına bir sıfır gelecektir. PD kontrolörü 1'e yakın olan sistem kutbunu ortadan kaldırarak şekilde seçilirse

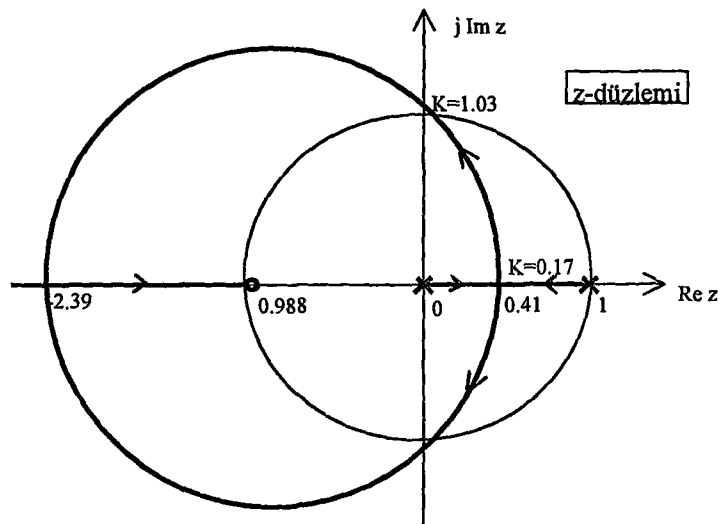
$$D(z)G_{h0}G_c(z) = K_c \frac{(z - 0.9646)}{z} K_s \frac{(z + 0.988)}{(z - 1)(z - 0.9646)} \quad (5.7)$$

K_c kontrolörden ve K_s sistemden gelen kazanç katsayıları olsun.

$$K_c = \frac{0.01K_p + K_D}{0.01}, K_s = 0.00004941 \text{ ve } K = K_c \cdot K_s \quad (5.8)$$

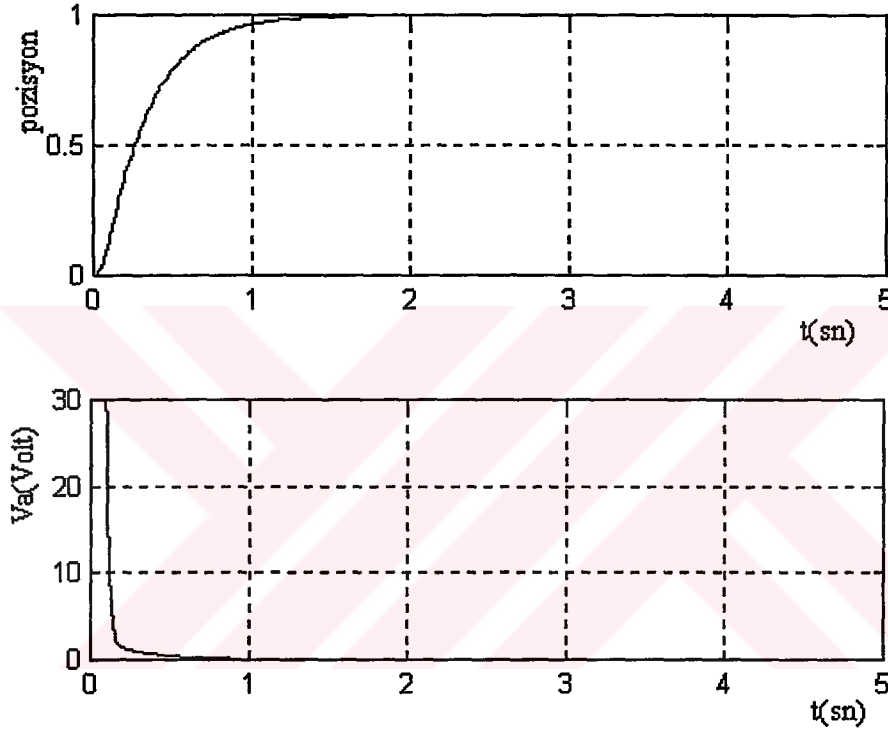
$$D(z)G_{h0}G_c(z) = K \frac{(z + 0.988)}{z(z - 1)} \quad (5.9)$$

K'ya göre köklerin yer eğrisi şekil 5.2'deki gibidir.



Şekil 5.2. PD kontrolörlü sistemin z-düzlemi köklerin yer eğrisi

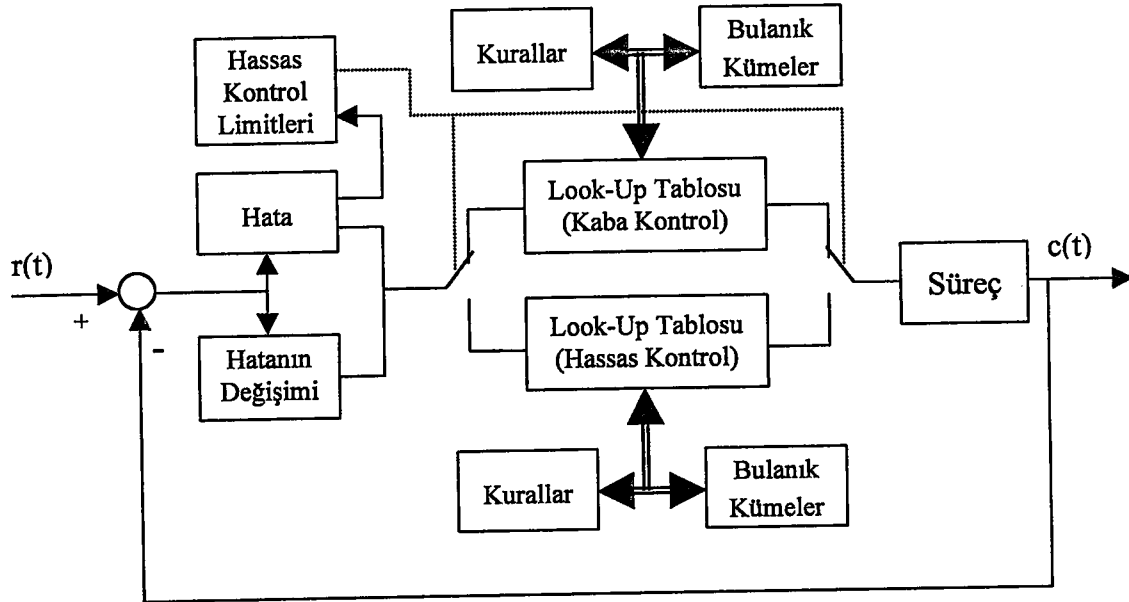
Kritik sönümlü bir sistem cevabı elde etmek için kapalı çevrim kutupları kopma noktasında olmalıdır. Karakteristik denklem kapalı çevrim kutuplarını sağlayacağından K kazancı bulunur. Karakteristik denklemden kopma noktası için K kazancı 0.17 olarak bulunur. Kazanç katsayısından ve yerleştirdiğimiz sıfırdan yararlanarak K_p 123.92 ve K_D 33.17 olarak bulunur.



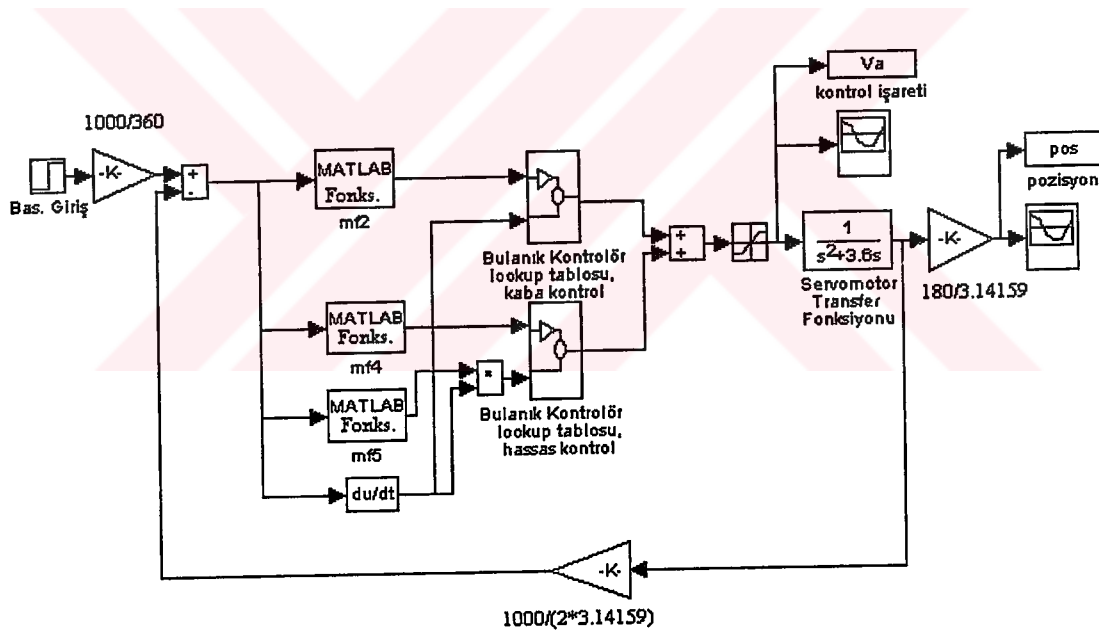
Şekil 5.3 $K_p=123.92$ ve $K_D=33.17$ için sistemin birim basamak cevabı

5.2 LOOK-UP KURAL TABLOSUNU KULLANARAK BULANIK KONTROLÖR SİMÜLASYONU

Transfer fonksiyonu $G(s)=K/s(s+3.6)$ şeklinde alınan servomotorun look-up tablosu bulanık kontrol algoritmasını kullanarak yapılan simulasyon MATLAB Simulink'te gerçekleştirilmiştir ve look-up tablosu değerleri [7] numaralı makaleden alınmıştır.

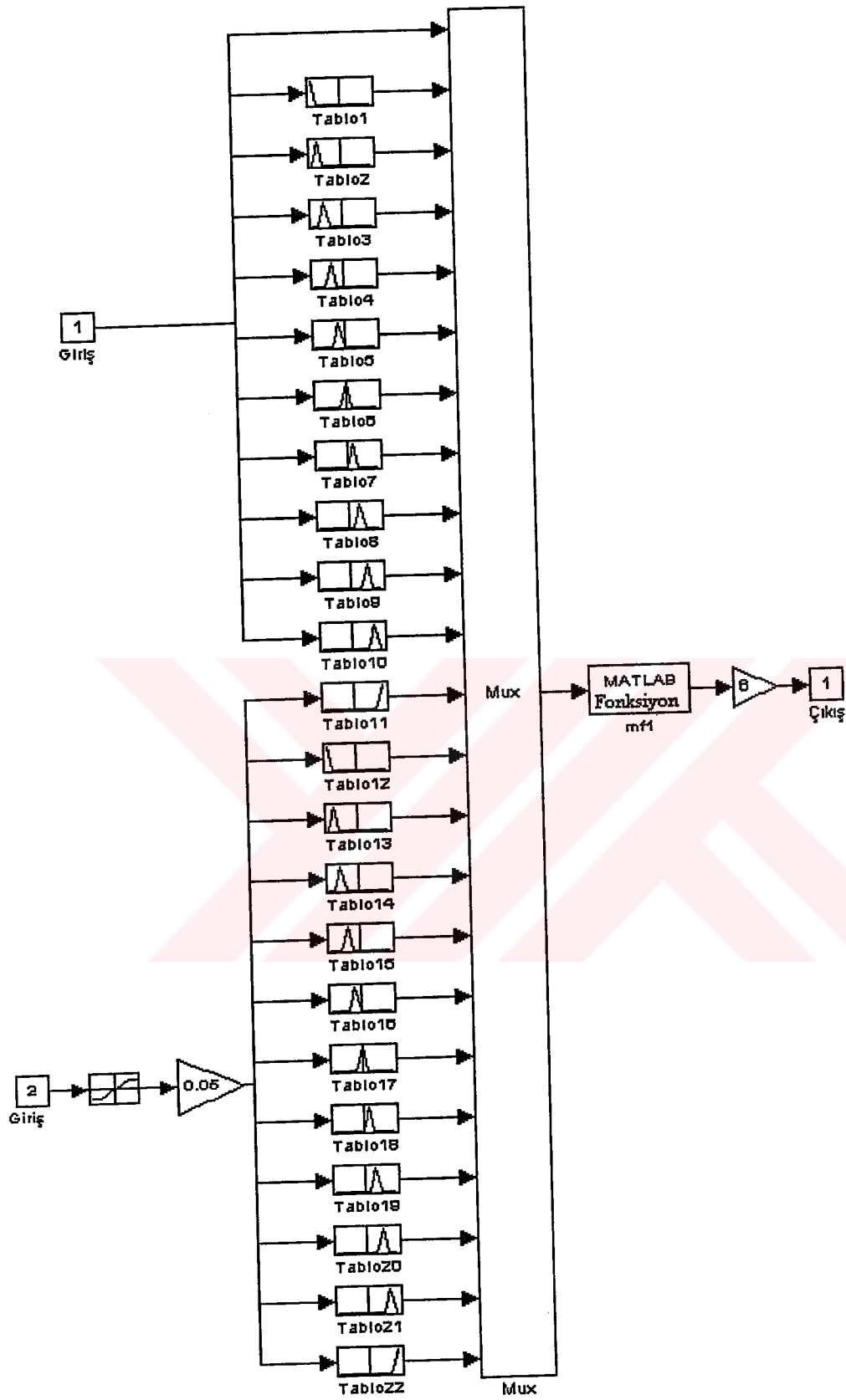


Şekil 5.4a Bulanık kontrolör yapısı

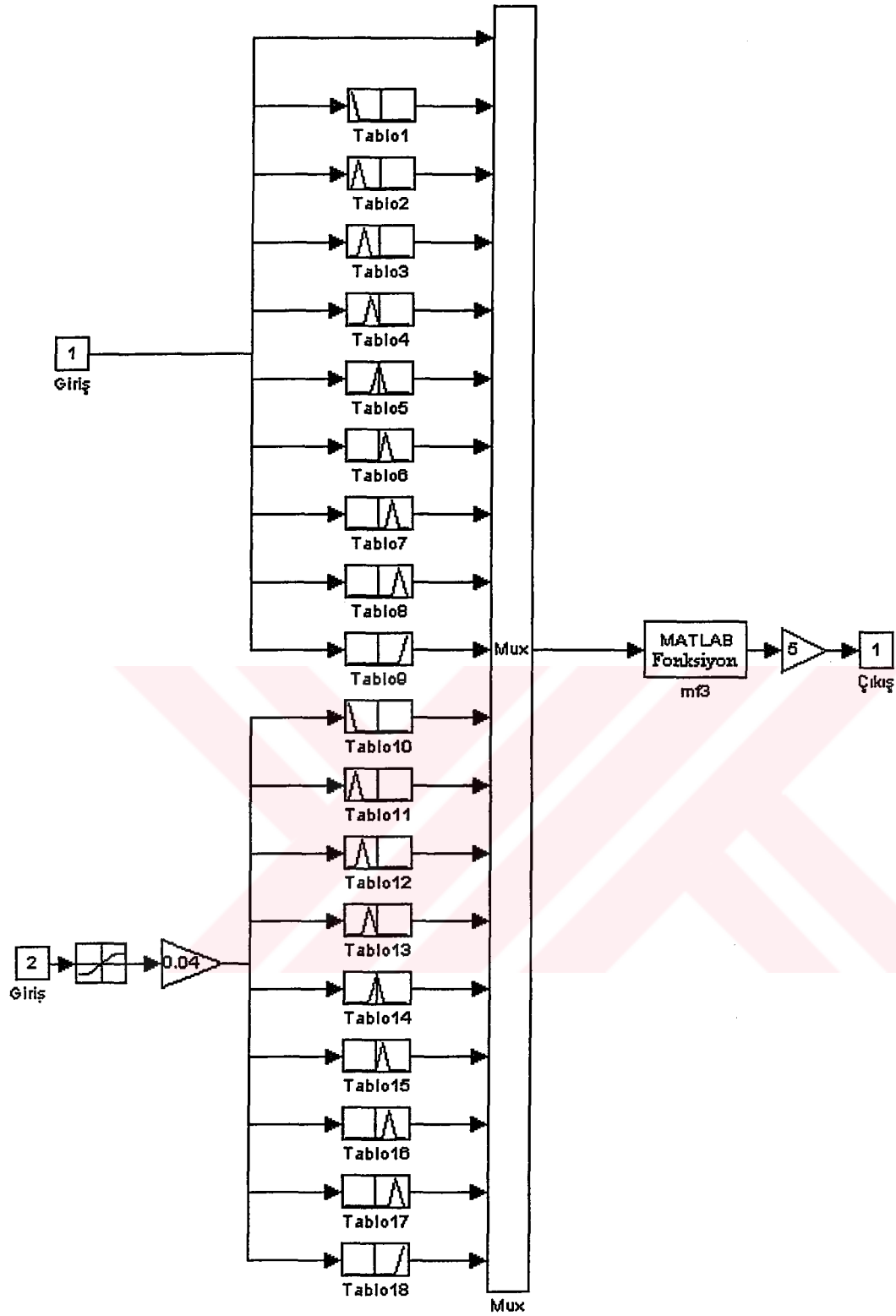


Şekil5.4b Look-Up tablosuyla oluşturulan bulanık kontrolörlü sistem modeli

Şekil 5.4a'da bulanık kontrolör yapısı, şekil 5.4b'de ise Simulink'te oluşturulan model verilmektedir. Literatürde ölçekleme parametreleri kullanarak sistem performansının iyileştirildiği görülmektedir[6]. Hata ve hatanın değişimi için ölçekleme katsayılarını 10 alırsak şekil 5.5'deki sistem cevabını elde etmekteyiz.



Şekil 5.5 Bulanık Kontrolör Kaba Kontrol Look-Up Tablosu



Şekil 5.6 Bulanık Kontrolör Hassas Kontrol Look-Up Tablosu

Aşağıda sırasıyla içeriği verilen mf1, mf2, mf3, mf4 ve mf5 MatLab fonksiyonlarını kullanarak Simulinkte gerçekleştirilen simülasyonda adımlar 0.01 sn olarak seçilmiştir. Eklerde verilen mf1 MatLab fonksiyonuyla kaba kontrol için oluşturulan bulanık kural tablosuna göre, mf3 MatLab fonksiyonuyla ise hassas kontrol için oluşturulan bulanık kural tablosuna göre kontrol işareti üretilmektedir. Hata -100 ile 100 arasındaysa hassas kontrol look-up tablosu kullanılmakta bunun dışında kaba kontrol look-up tablosu kullanılmaktadır. Şekil 5.5'te bulanık kontrolör kaba kontrol look-up tablosu, şekil 5.6'da ise hassas kontrol look-up tablosu görülmektedir.

```
function sonuc1=mf1(u);
```

```
% bulanık kontrolör kaba kontrol look-up tablosu
```

```
f1=[-5 -5 -4 -4 -3 -1 0 1 1 2 2;  
    -5 -5 -5 -4 -3 -1 0 1 1 2 2;  
    -5 -5 -5 -4 -3 -1 0 1 2 2 2;  
    -5 -5 -4 -4 -3 -1 1 2 3 3 3;  
    -5 -5 -4 -3 -2 0 1 2 3 3 4;  
    -5 -4 -4 -2 -2 0 1 3 4 4 4;  
    -4 -4 -3 -2 -1 0 2 3 4 4 4;  
    -4 -4 -3 -2 -1 1 2 3 5 4 5;  
    -3 -4 -2 -1 0 1 2 3 5 4 5;  
    -2 -3 -2 -1 0 1 3 3 5 5 5;  
    -2 -3 -1 -1 0 1 3 3 5 5 5];
```

```
if (u(1)>=5) sonuc=5;
```

```
elseif (u(1)<=-5) sonuc=-5;
```

```
elseif (u(1)==0) sonuc=0;
```

```
else % mamdani min çıkarımı
```

```
for i=1:11
```

```
for j=1:11
```

```
if (u(j+12)>u(i+1))
```

```
t(j,i)=u(i+1);
```

```
else
```

```
t(j,i)=u(j+12);
```

```
end
```

```
end
```



```

end
% ağırlık merkezli durulayıcı
pay=0; payda=0;
for i=1:11
    for j=1:11
        pay=pay+(t(i,j)*f1(i,j));
        payda=payda+t(i,j);
    end
end
sonuc1=pay/payda;
end;

```

```

function sonuc2=mf2(u); % kaba kontrol seçimi
if (abs(u)>100 )
    sonuc2=u*0.005;
else
    sonuc2=0;
end;

```

```

function sonuc3=mf3(u);
%bulanık kontrolör hassas kontrol look-up tablosu değerleri
f2=[ -4  -4  -3  -3  -2  -1  -1  0  1;
      -4  -3  -3  -2  -2  -1  0  1  1;
      -3  -2  -2  -1  -1  -1  0  1  2;
      -2  -2  -1  0  1  0  1  2  2;
      -2  -2  -1  0  0  0  1  2  3;
      -2  -2  0  1  1  1  2  3  3;
      -1  -1  0  1  1  1  2  3  4;
      0  0  1  2  2  2  3  3  4;
      0  0  1  2  3  3  3  4  4];
if (u(1)>=4) sonuc=4;
elseif (u(1)<=-4) sonuc=-4;
else

```

```

% mamdani min çıkarımı
for i=1:9
    for j=1:9
        if (u(j+10)>u(i+1))
            t(j,i)=u(i+1);
        else
            t(j,i)=u(j+10);
        end
    end
end
% ağırlık merkezli durulayıcı
pay=0; payda=0;
for i=1:9
    for j=1:9
        pay=pay+(t(i,j)*f2(i,j));
        payda=payda+t(i,j);
    end
end
sonuc=pay/payda;
end;

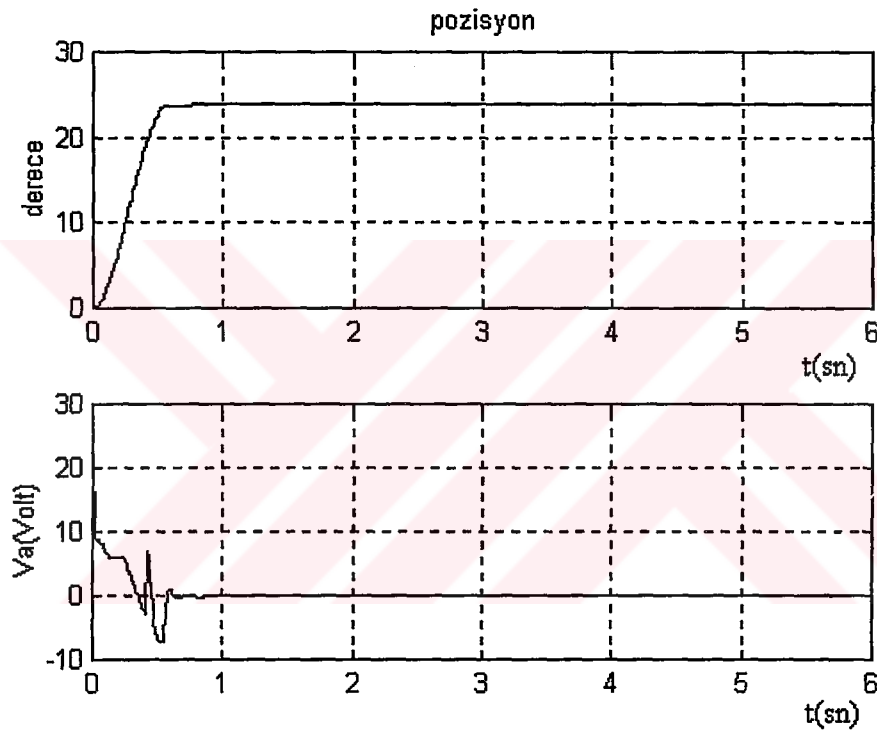
function sonuc=mf4(u); %hassas kontrol lookup tablosu
if (abs(u)<=100)
    sonuc=u*0.04;
else
    sonuc=0;
end;

```

```

function sonuc=mf5(u); % hassas kontrol seçimi
if (abs(u)<=100)
    sonuc=1;
else
    sonuc=0;
end;

```



Şekil 5.7 Look-Up tablosu kullanarak elde edilen sistem cevabı

5.3 BULANIK PI KONTROLÖR SİMÜLASYONU

Bulanık PI kontrolör için, kontrolör çıkışı

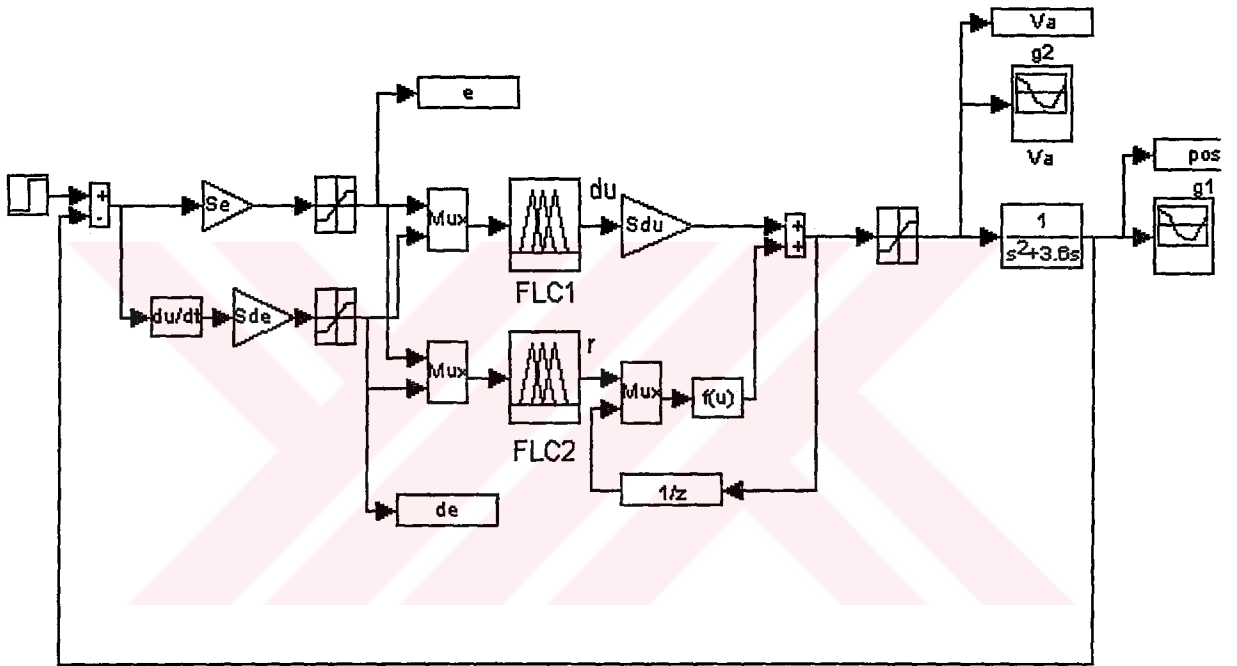
$$u(k+1)=u(k)+du(k) \quad (5.10)$$

$du(k)$; $e(k)$ ve $de(k)$ kullanılarak belirlenir. pı kontrolör bilindiği üzere yapısındaki integrasyon işlemi nedeniyle geçici hal cevabı iyi olmayan bir kontrolördür. Özellikle yüksek dereceden sistemler için bulanık PI kontrolörün geçici hal cevabını

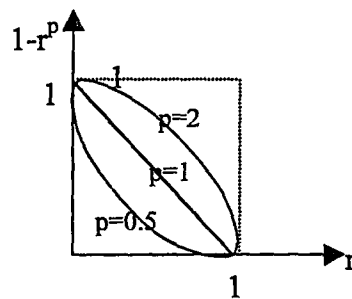
iyileştirmek kolay değildir[8]. Sistemin hızlı cevap vermesi için büyük bir kontrol işareti gereklidir, fakat aşım yapmasını engellemek için küçük kontrol işareti üretilmelidir.

$$u(k+1)=(1-(r(k))^p)u(k)+du(k) \quad f(u)=(1-(r(k))^p) \quad (5.11)$$

$r(k)$ resetleme oranıdır. Sabit p ise r 'nin resetleme işlemindeki etkisini belirler. Eğer $r(k)$ bir ise integrasyon işlemi kalkar, sıfır ise klasik bulanık PI kontrolör yapısı elde edilir.



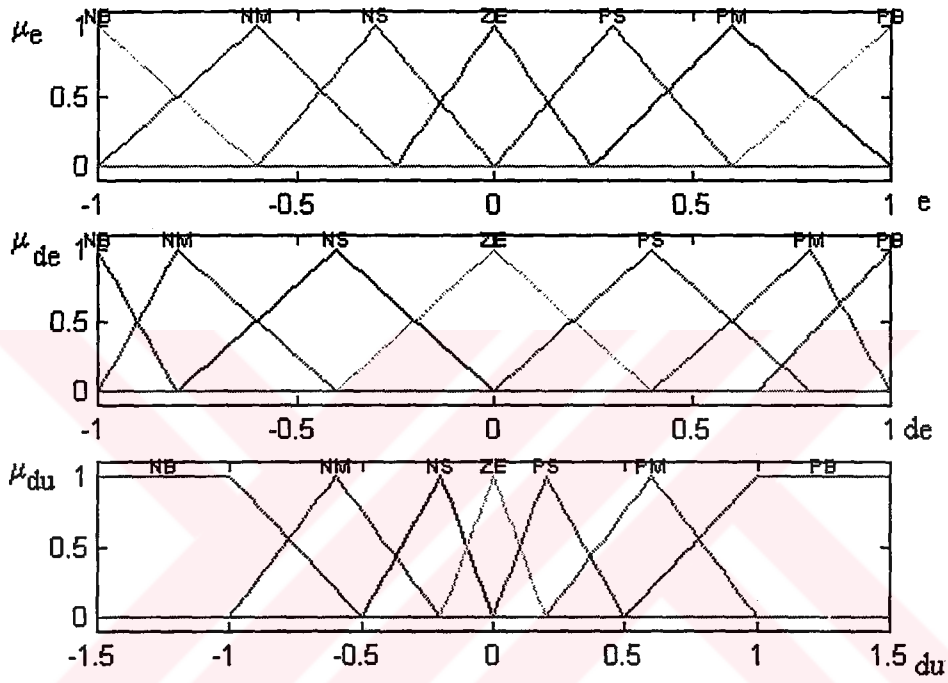
Şekil 5.8 Genelleştirilmiş Bulanık PI Kontrolör Modeli



Şekil 5.9 Resetleme İşleminde p 'nin Etkisi

de \ e	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NM	NS	NS	ZE
NM	NB	NM	NM	NM	NS	ZE	PS
NS	NB	NM	NS	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PM	PM	PM	PB
PB	ZE	PS	PS	PM	PB	PB	PB

Şekil 5.10 FLC1-du'yu Hesaplamak İçin Bulanık Kural Tablosu

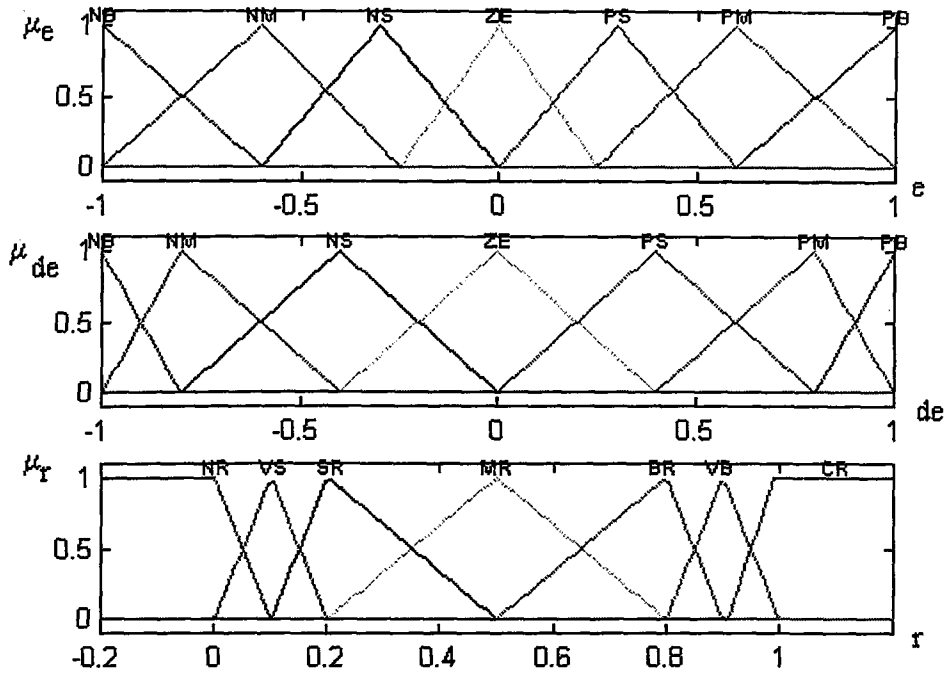


Şekil 5.11 e, de ve du Üyelik Fonksiyonları

FLC1 ve FLC2 bulanık kontrolörleri, MatLab Fuzzy Logic Toolbox kullanılarak oluşturulmuştur. Toolbox bulanık çıkarım sistemi olarak Mamdani çıkarımı kullanılmıştır. Oluşturulan kontrolörler Simulink'e aktarılmış ve simülasyonlar yapılmıştır[8].

de \ e	NB	NM	NS	ZE	PS	PM	PB
NB	NR	SR	BR	CR	BR	SR	NR
NM	NR	NR	MR	VB	MR	NR	NR
NS	NR	NR	VS	BR	VS	NR	NR
ZE	NR	NR	NR	NR	NR	NR	NR
PS	NR	NR	VS	BR	VS	NR	NR
PM	NR	NR	SR	VB	SR	NR	NR
PB	NR	SR	MR	CR	MR	SR	NR

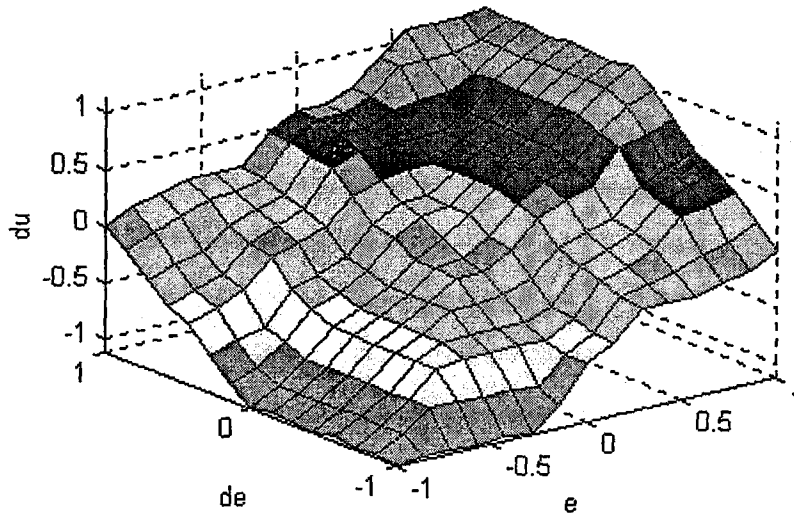
Şekil 5.12 FLC2-r'yi Hesaplamak İçin Bulanık Kural Tablosu



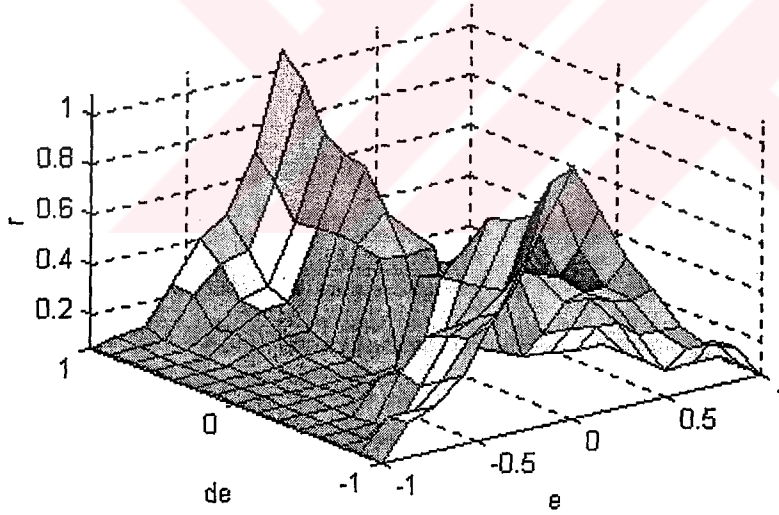
Şekil 5.13 e, de ve r Üyelik Fonksiyonları

Bulanık PI Kontrolör simülasyonu için önce klasik bulanık PI kontrolörlü sistemin cevabı görülmüştür. Şekil 5.16a'da ve 5.16b'de ikinci dereceden bir sistem için elde edilen sonuçlar görülmektedir. Hata ve hatanın değişimi ölçekleme parametrelerine göre sistem cevabı çok önemli değişiklikler göstermektedir. Sde küçültüldüğünde sistem hızlı cevap vermesine rağmen aşım yapmaya başlamakta ve osilasyona girmektedir. Se hata ölçekleme parametresi artırıldığında benzer durum yaşanmaktadır. Ölçekleme parametrelerinin küçük bir değişiminde sistem cevabı kötüleşmektedir.

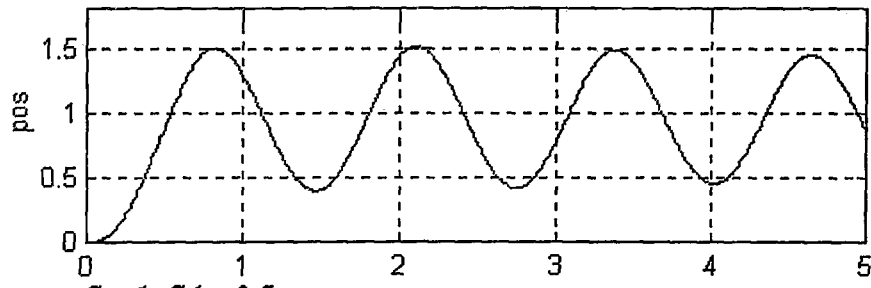
Resetleme parametresi r kullanılarak elde edilen geliştirilmiş bulanık PI kontrolör için üyelik fonksiyonları şekil 5.12'de ve kural tablosu şekil 5.13'de verilmiştir. Resetleme parametresi r için oluşturulan kurallar, integre edilmiş kontrol işaretini normal olmayan durum için sıfırlamaktadır. r büyük hatalar için küçüktür ve kontrol işaretini çok az etkilemektedir, ancak hata küçüldükçe r büyüyerek kontrol işaretinin aşırı artmasını engellemektedir. Şekil 5.17'de çeşitli ölçekleme parametreleri için elde edilen simülasyon sonuçları görülmektedir. Sde arttıkça sistem yerleşme zamanı büyümektedir, ancak sistem aşısız olarak oturmaktadır. Se büyüdükçe sistem hızlı oturmakta ve aşım yapmaya başlamaktadır. Ancak aşım miktarı büyük oranda azalmıştır. Sdu artırıldığı takdirde ise sistem daha hızlı olarak oturmaktadır.



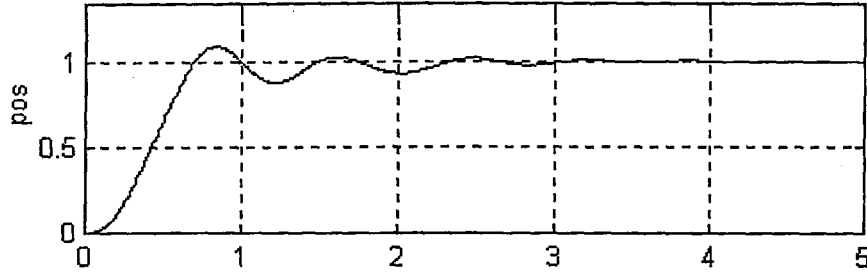
Şekil 5.14. FLC1- du kontrol işareti değişimi bulanık kontrolör giriş-çıkış arayüzeyi



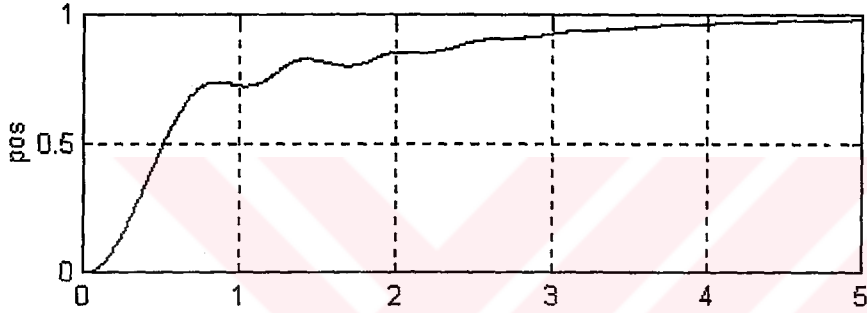
Şekil 5.15. FLC2- r resetleme parametresi bulanık kontrolörü giriş-çıkış ara yüzeyi



$Se=1, Sde=0.5$

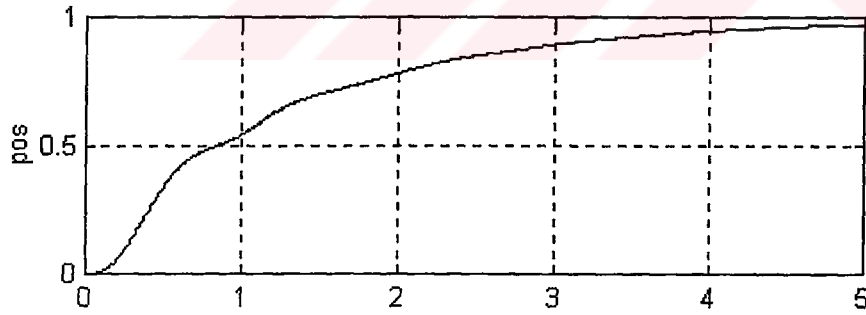


$Se=1, Sde=1$

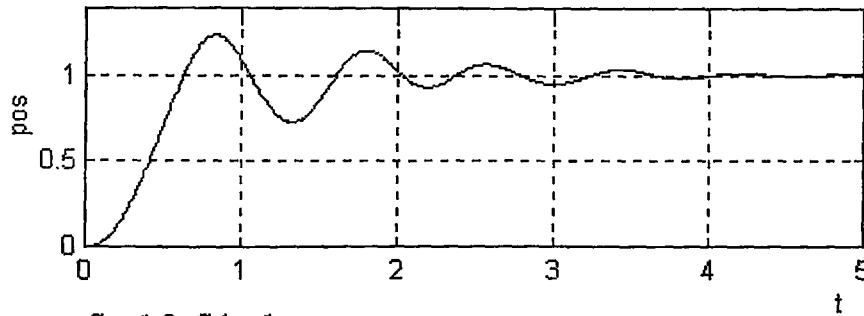


$Se=1, Sde=2$

Şekil 5.16a $Se=1$ ve $Sdu=1$ için Sde 'ye göre sistem cevabı

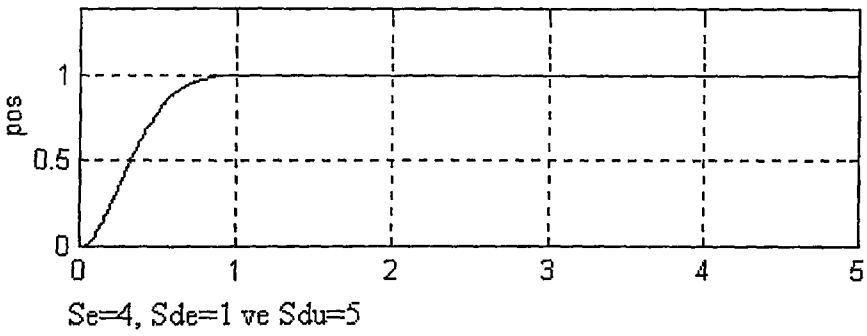
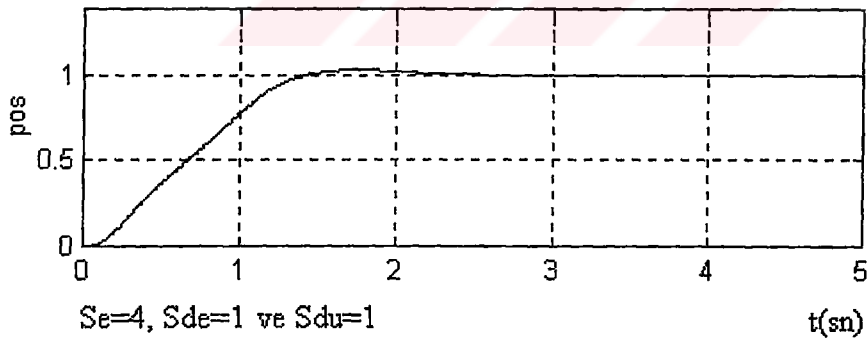
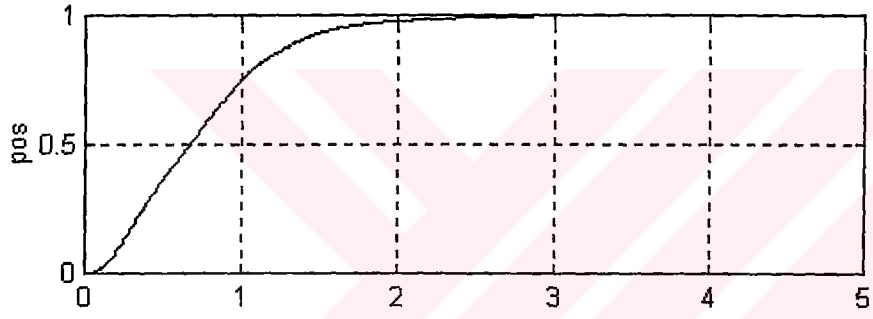
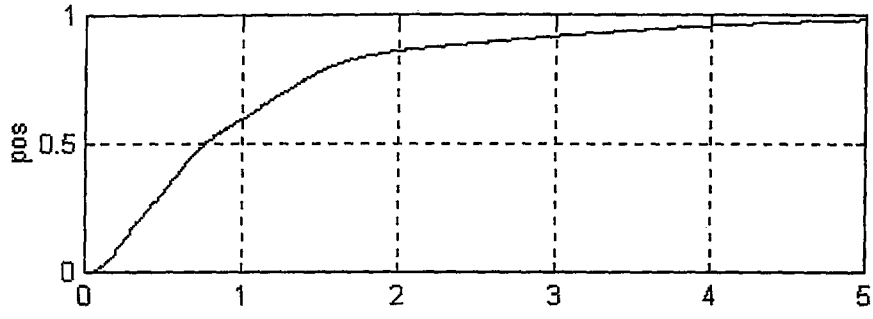
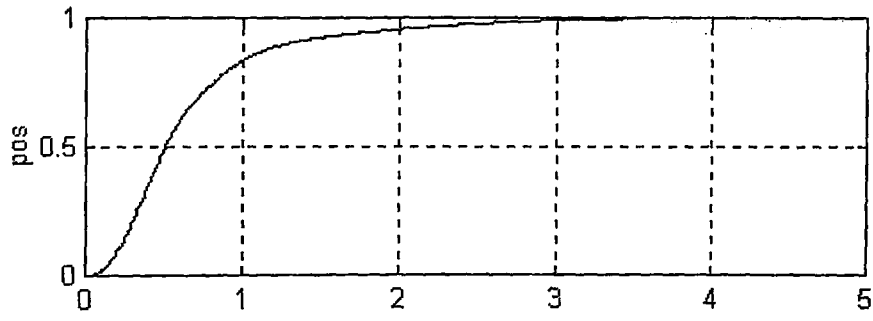


$Se=0.5, Sde=1$



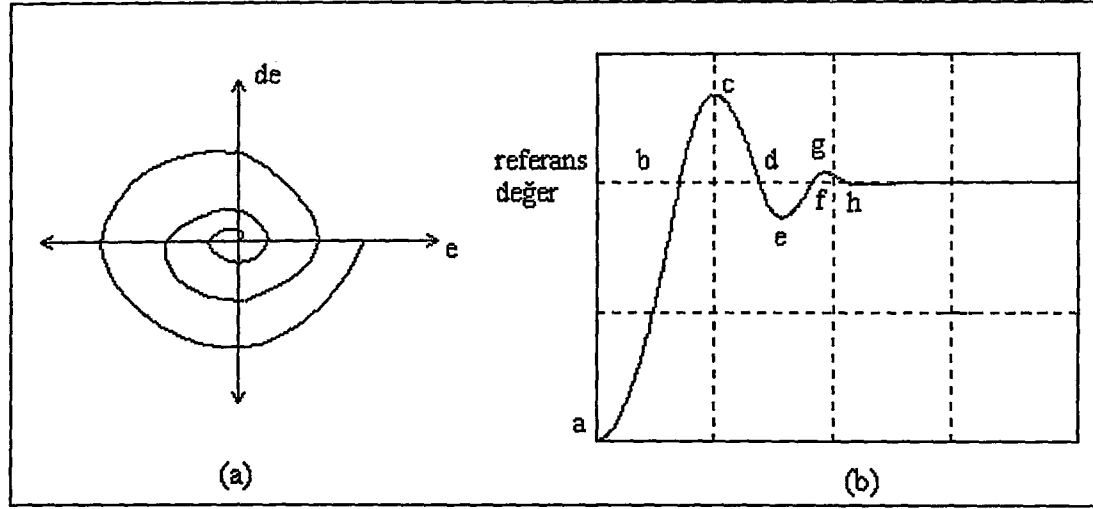
$Se=1.2, Sde=1$

Şekil 5.16b $Sde=1$ ve $Sdu=1$ için klasik bulanık PI kontrolör sistem cevabı



Şekil 5.17 Resetleme parametrelili genelleştirilmiş bulanık PI kontrolör simülasyonu

5.4 BULANIK KONTROLÖR TASARIMI ve SİMÜLASYONU



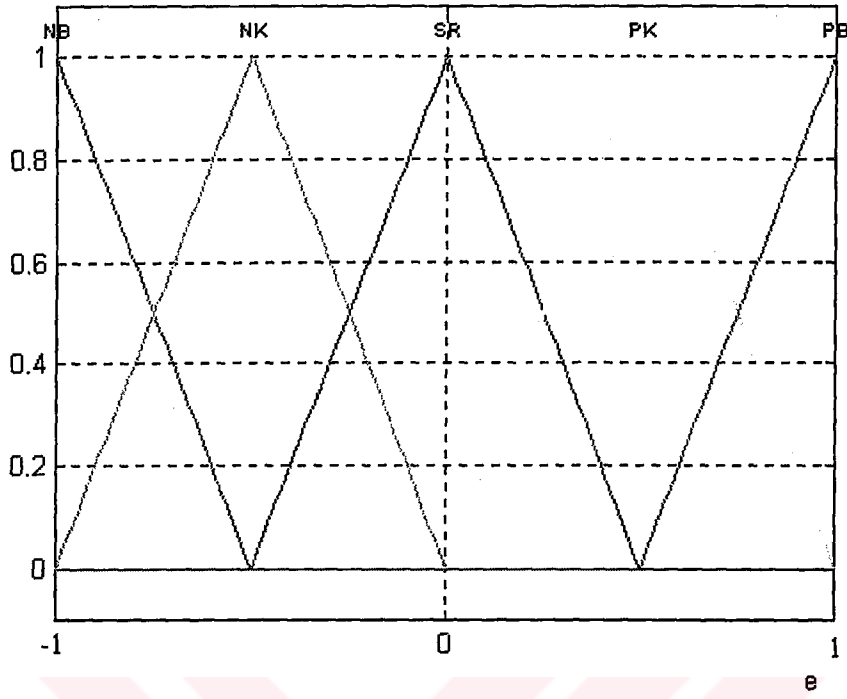
Şekil 5.18 Faz Yüzeyini Kullanarak Kural Tablosu Düzenleme

(a) Faz Yüzeyi Eğrisi (b) Sistem Basamak Cevabı

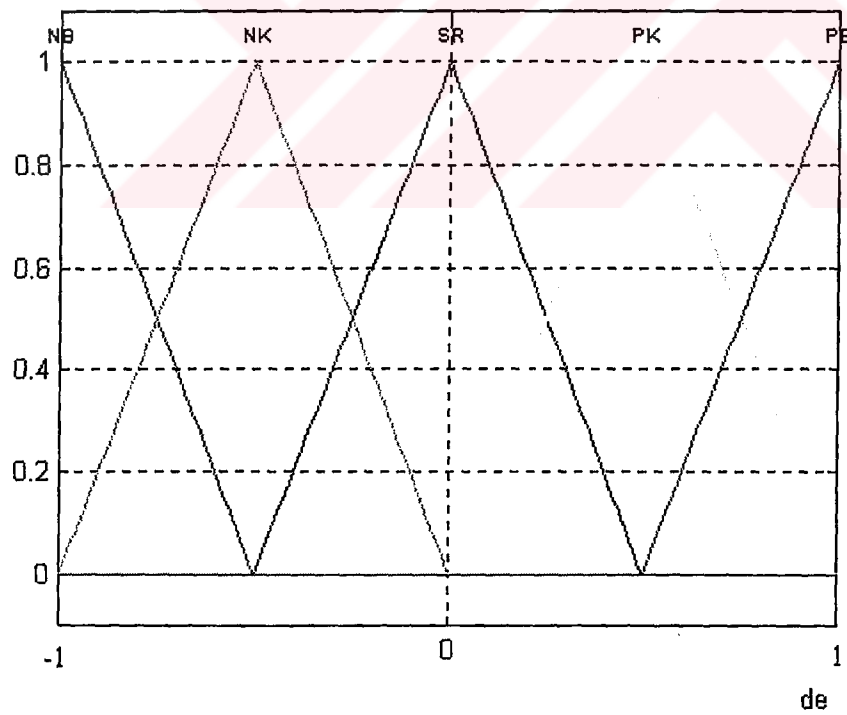
King ve Mamdani tarafından bulanık kural tabanı oluşturmak için sunulan metot, sistem cevabının faz yüzeyinde istenen eğriyi (dolayısıyla hedeflenen aşım ve yükselme zamanı) izlemesini sağlamayı hedefler[9,10]. Şekil 5.18'de verilen basamak cevabına sahip bir sistem için bulanık kontrolör tasarlayalım. Şekil 5.20a'da ve şekil 5.20b'de verilen üyelik fonksiyonlarından yararlanarak şekil 5.19'daki kural tablosunu oluşturalım.

$\begin{matrix} de \\ e \end{matrix}$	NB	NK	SR	PK	PB
NB	NB	NB	NB	NB	NB
NK	NB	NB	NB	NK	NK
SR	NK	NK	SR	PK	PK
PK	PK	PK	PB	PB	PB
PB	PB	PB	PB	PB	PB

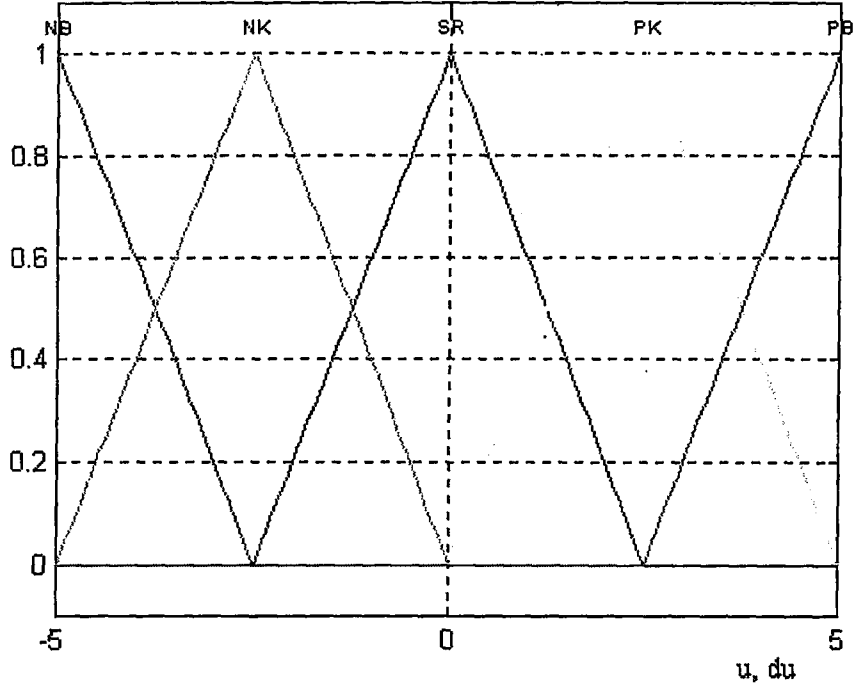
Şekil 5.19 Bulanık Kontrolör Kural Tablosu



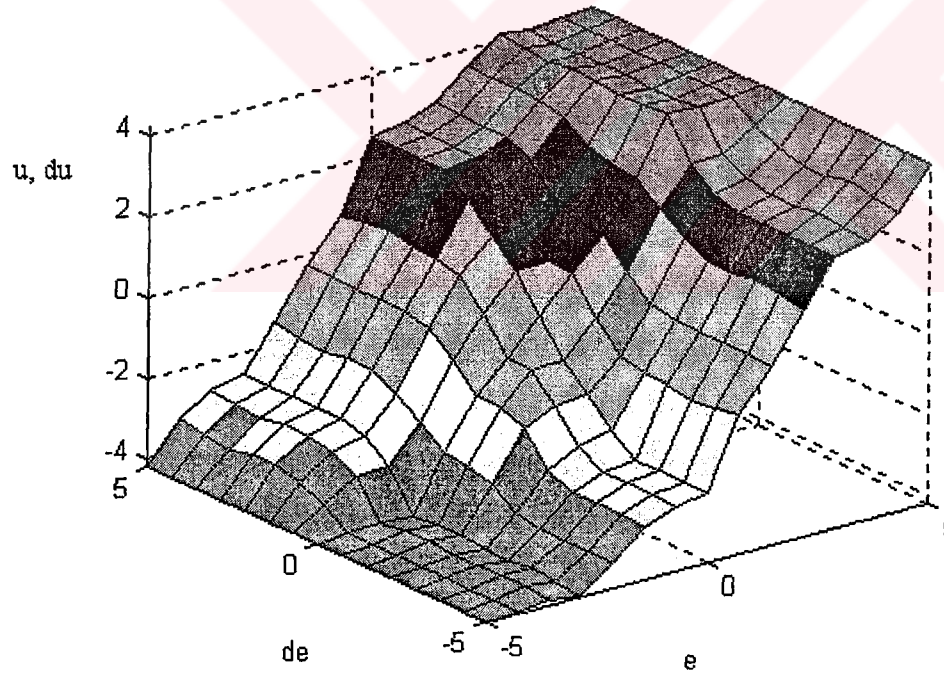
Şekil 5.20a Hata için belirlenen üyelik fonksiyonları



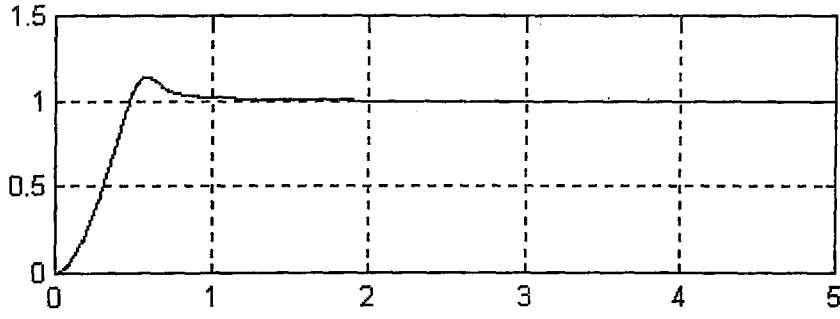
Şekil 5.20b Hatanın değişimi için belirlenen üyelik fonksiyonları



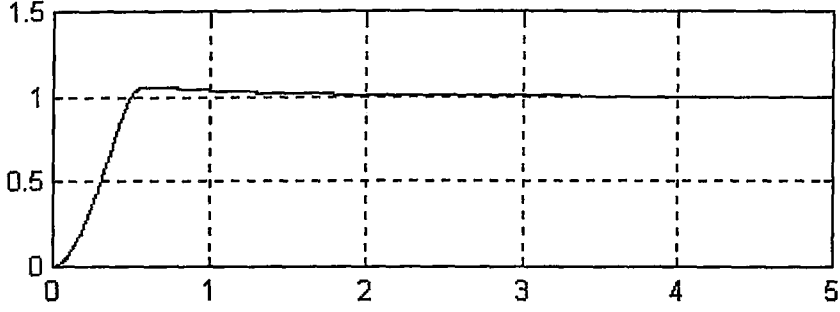
Şekil 5.21 Çıkışlar için belirlenen üyelik fonksiyonları



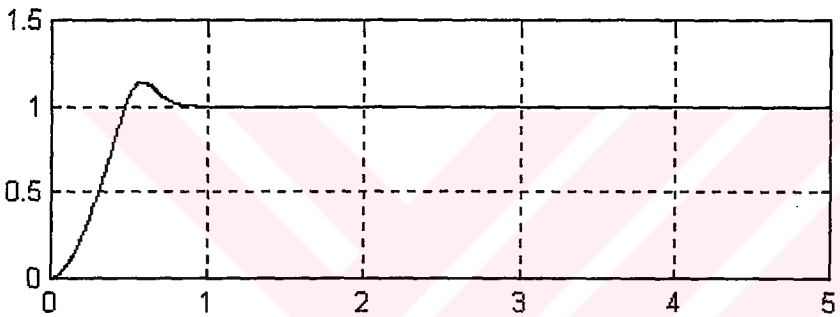
Şekil 5.22 Bulanık kontrolör giriş-çıkış arayüzeyi



$Se=1, Sde=1, Su=30$

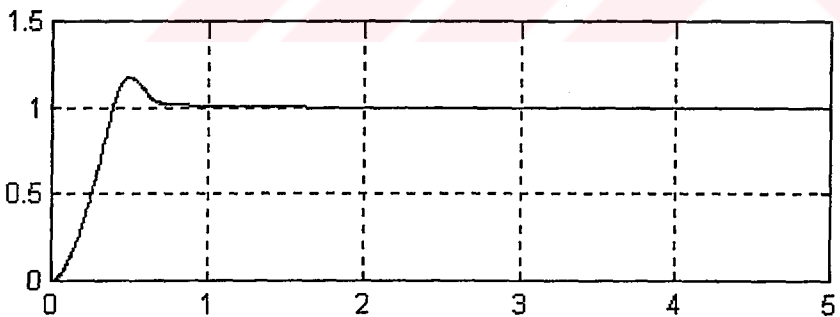


$Se=0.5, Sde=1, Su=30$

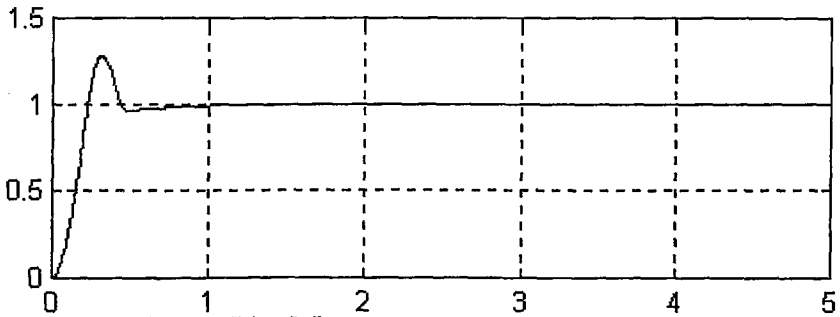


$Se=1, Sde=5, Su=30$

$G(s)=1/(s+3.6)$ için bulanık PD kontrolör için sistem birim basamak cevabı



$Se=1, Sde=1, Sdu=0.2$



$Se=1, Sde=1, Sdu=0.5$

Şekil 5.23 $G(s)=1/(s+3.6)$ için klasik bulanık PI kontrolörle sistem birim basamak cevabı

MatLab Fuzzy Logic Toolbox kullanılarak tasarlanan kontrolör, bulanık PD ve klasik bulanık PI kontrolör olarak uygulanmış ve simülasyon sonuçları şekil 5.23'deki gibi elde edilmiştir.



SONUÇLAR ve ÖNERİLER

Bulanık kontrol; anlaşılması ve geliştirilmesi kolay bir kontrol yöntemi olduğundan dolayı endüstride çalışan mühendislerin tercih edebilecekleri bir kontrol yöntemidir.

Nitekim günümüzde görülmektedir ki birçok tüketici ürününde ve endüstriyel uygulamalarda bulanık kontrolörler kullanılmaktadır.

Bir sistem için, bulanık kontrolörün kural tablosunu ve parametrelerini dikkatli seçerek uygun bir bulanık kontrolör tasarlamak mümkündür. Oluşturulan kural tablosu aynı dinamik davranışa sahip diğer bir sisteme de rahatlıkla uygulanabilir.

Yapılan simülasyonlarda elde edilen sonuçlardan bulanık kontrolörle sistemin çok hızlı olarak referans değere ulaştığı ve PID kontrolörde ise daha yavaş olarak referansa yaklaşılmasına rağmen integratörün sürekli hal performansını iyileştirdiği söylenebilir.

Klasik bulanık PI kontrolör; ikinci ve üstü derecelerdeki sistemler için iyi sonuçlar vermemektedir. Simülasyonlarda incelemesi yapılan klasik bulanık PI kontrolöre resetleme parametresi eklenmesiyle aşımın çok büyük bir oranda düştüğü ve sistem geçici hal cevabının dikkate değer düzeyde iyileştiği görüldü.

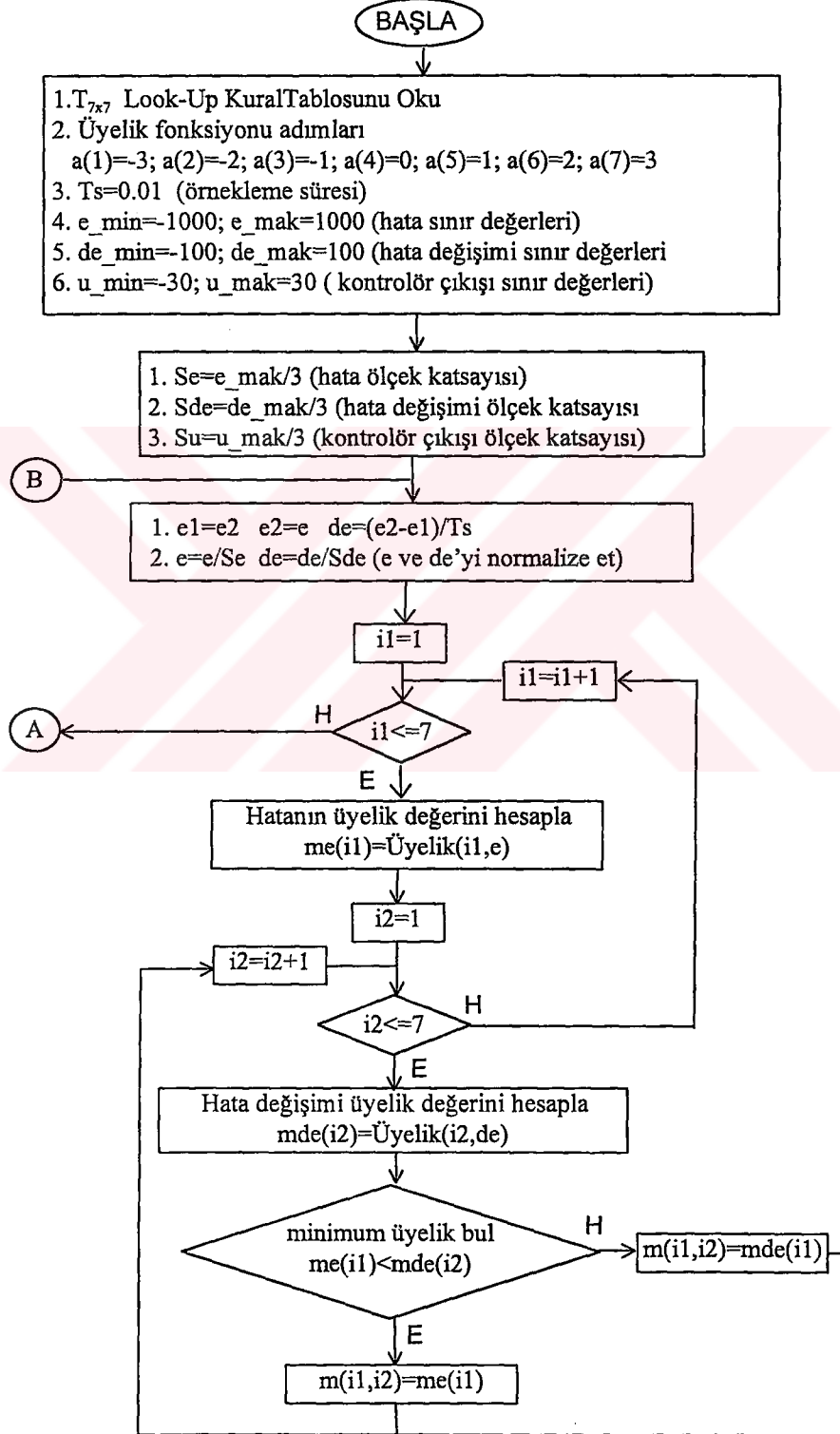
Bulanık kontrolörde ölçekleme parametrelerinin sistem performansına olan etkisi incelendi. Deneme-yanılma metoduyla ayarlanan ölçekleme parametreleriyle sistem cevabı arasında bir bağıntı olduğu saptanmıştır. Bu bağıntıdan yararlanarak bir ölçekleme parametresi ayarlama algoritması geliştirilebilir.

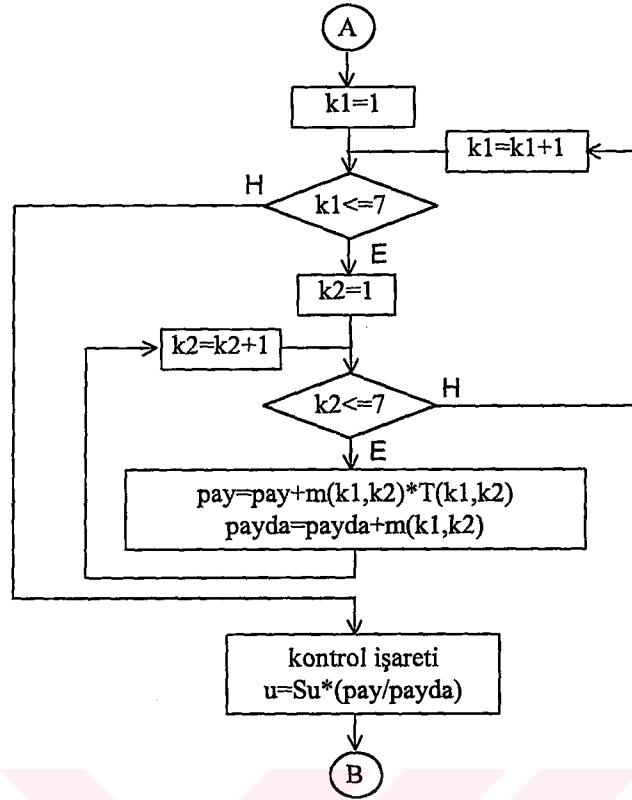
KAYNAKLAR

- [1] **WANG, Li-Xin**, A Course in Fuzzy Systems and Control,
Prentice Hall, 1997
 - [2] **YEN, John, LANGARI, Reza, ZADEH, Lotfi**, Industrial Applications of Fuzzy
Logic and Intelligent Systems, IEEE Press Marketing, 1995
 - [3] **KAYNAK, Oktay, ARMAĞAN Gülçin** , Süreç Denetiminde Yeni Bir Yaklaşım:
Bulanık Mantık, Sayısal Motor Kontrolü Semineri ,14 Eylül 1995
 - [4] **LEE, Jihong**, On Methods for Improving Performance of PI-Type Fuzzy Logic
Controllers, IEEE TRANSACTIONS ON FUZZY SYSTEMS Vol.1, NO.4,
November 1993
 - [5] **SHAHIAN, Bahram**, Control Sytem Design Using MATLAB,
Prentice Hall, 1993
 - [6] **OGATA, Katsuhiko**, Modern Control Engineering,
Prentice Hall, 1997
 - [7] **Li, Y.F., Lau, C.C.**, Development Of Fuzzy Algorithms for Servo Systems, IEEE
Control System Magazine, April 1989
 - [8] Fuzzy Logic Toolbox User's Guide , 1995
 - [9] **CHUEN, Chien Lee**, Fuzzy Logic in Control Systems: Fuzz Logic Controller-
Part1, IEEE Transactions On Systems, Man ad Cybernetics Vol:20, No:2
March/April 1990
 - [10] **CHUEN, Chien Lee**, Fuzzy Logic in Control Systems: Fuzz Logic Controller-
Part2, IEEE Transactions On Systems, Man ad Cybernetics Vol:20, No:2
March/April 1990
 - [11] **PEDRYCZ, Witold**, "Fuzzy Control and Fuzzzy Sytems",
Research Studies Press Ltd. England, 1989
-

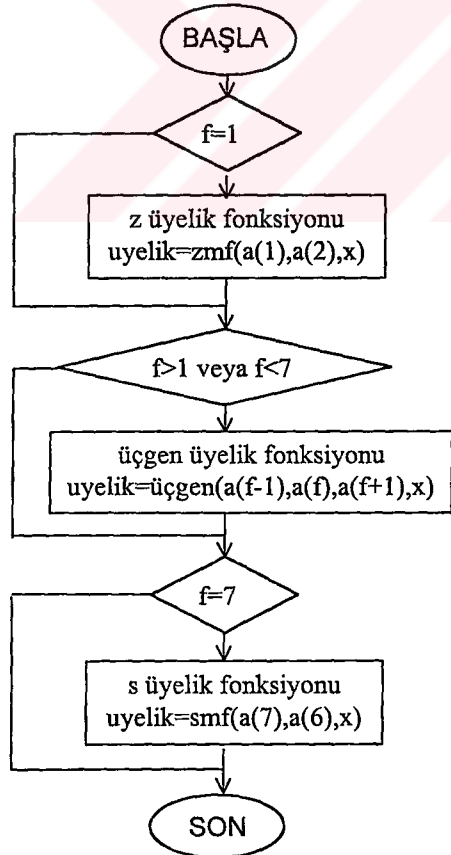
EKLER

BULANIK KONTROLÖR AKIŞ DİYAGRAMI

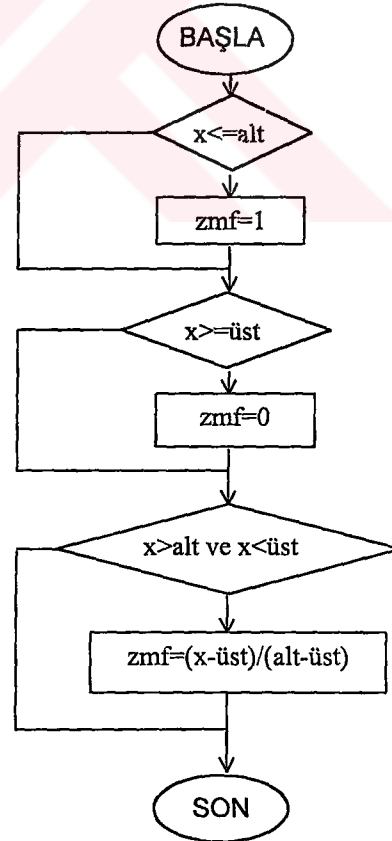




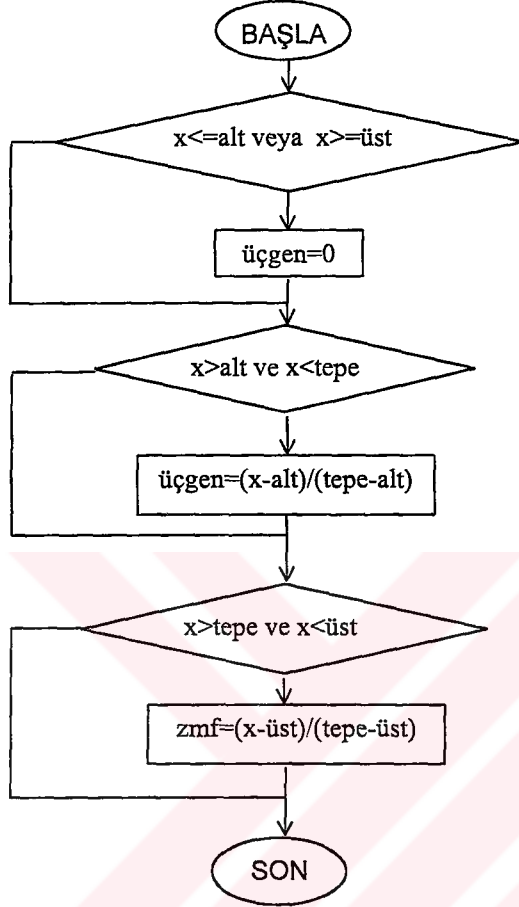
Üyelik Alt Programı
Üyelik(f,x)



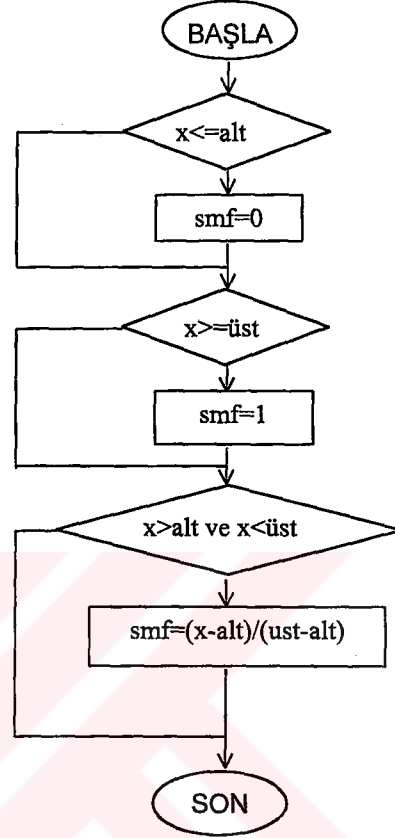
z Üyelik Fonksiyonu Alt Programı
zmf(alt,üst,x)



Üçgen Üyelik Fonksiyonu Alt Programı
üçgen(alt,tepe,üst,x)



s Üyelik Fonksiyonu Alt Programı
smf(alt,üst,x)



{*/ BULANIK KONTROLÖR */}

```

uses crt,dos;
var
  T,m      :array [1..7,1..7] of real;
  me,mde   :array [1..7] of real;
  Ts       :real;
  il,i2    :byte;
  e,e1,e2  :real;
  de,u     :real;
  Se,Sde,Su :real;
  n3,n2,n1 :real;
  sr       :real;
  p1,p2,p3 :real;
  pay,payda :real;
  son      :boolean;
  e_mak,e_min :longint;
  eh_mak,eh_min :longint;
  
```

```

de_mak,de_min      :longint;
u_mak,u_min        :longint;

```

```

{* z üyelik fonksiyonu *}
function zmf(alt:real;ust:real; x:real):real;
begin
  if x<alt then
    zmf:=1
  else if x>ust then
    zmf:=0
  else
    zmf:=(x-ust)/(alt-ust);
end;

```

```

{* üçgen üyelik fonksiyonu *}
function trimf(alt,tepe,ust,x:real):real;
begin
  if (x<alt) or (x>ust) then
    trimf:=0
  else if x<tepe then
    trimf:=(x-alt)/(tepe-alt)
  else
    trimf:=(x-ust)/(tepe-ust);
end;

```

```

{* s üyelik fonksiyonu *}
function smf(alt,ust,x:real):real;
begin
  if x<alt then
    smf:=0
  else if x>ust then
    smf:=1
  else
    smf:=(x-alt)/(ust-alt);
end;

```

```

{* üyelik değerini bul *}
function uye(f:real;x:real):real;
begin
  if f=1 then
    uye:=zmf(n3,n2,x)
  else if f=2 then
    uye:=trimf(n3,n2,n1,x)
  else if f=3 then
    uye:=trimf(n2,n1,sr,x)
  else if f=4 then
    uye:=trimf(n1,sr,p1,x)
  else if f=5 then
    uye:=trimf(sr,p1,p2,x)
  else if f=6 then

```

```

    uye:=trimf(p1,p2,p3,x)
else if f=7 then
    uye:=smf(p3,p2,x);
end;

{*/ ana döngü */}
begin

    {*/kural tablosunu yükle*}
    T[1,1]:= 0;T[1,2]:= 0;T[1,3]:=-2;T[1,4]:=-3;T[1,5]:=-3;T[1,6]:=-3;T[1,7]:=-3;
    T[2,1]:= 0;T[2,2]:= 0;T[2,3]:= 0;T[2,4]:=-2;T[2,5]:=-3;T[2,6]:=-2;T[2,7]:=-2;
    T[3,1]:= 0;T[3,2]:= 0;T[3,3]:=-1;T[3,4]:=-1;T[3,5]:=-1;T[3,6]:=-3;T[3,7]:=-2;
    T[4,1]:= 1;T[4,2]:= 3;T[4,3]:= 0;T[4,4]:= 0; T[4,5]:= 0;T[4,6]:=-3;T[4,7]:=-1;
    T[5,1]:= 2;T[5,2]:= 3;T[5,3]:= 1;T[5,4]:= 1; T[5,5]:= 1;T[5,6]:= 0;T[5,7]:= 0;
    T[6,1]:= 2;T[6,2]:= 2;T[6,3]:= 3;T[6,4]:= 2; T[6,5]:= 0;T[6,6]:= 0;T[6,7]:= 0;
    T[7,1]:= 3;T[7,2]:= 3;T[7,3]:= 3;T[7,4]:= 3; T[7,5]:= 2;T[7,6]:= 0;T[7,7]:= 0;

    Ts:=0.01; {*/ örnekleme periyodu */}
    {*/ üyelik fonksiyon aralıklarını oku */}
    n3:=-3;
    n2:=-2;
    n1:=-1;
    sr:= 0;
    p1:= 1;
    p2:= 2;
    p3:= 3;
    e_mak:=1000; e_min:=-1000;{*/ hatanın sınır değerleri */}
    eh_mak:=100; eh_min:=-100;{*/ hassas kontrol hata sınır değerleri */}
    de_mak:=-100; de_min:=100; {*/ hatanın değişiminin sınır değerleri */}
    u_mak:=30; u_min:=-30; {*/ kontrolör çıkışı sınır değerleri */}

    repeat
        e1:=e2;
        e2:=e;
        de:=(e2-e1)/Ts;

        {*/ hata sınırlarına göre üyelik fonksiyonlarını ölçeklendir. */}
        if (e<-100) or (e>100) then
            begin
                Se:=e_mak/3;
                Sde:=de_mak/3;
                Su:=u_mak/3;
            end
        else
            begin
                Se:=eh_mak/3;
                Sde:=de_mak/3;
                Su:=u_mak/3;
            end
        end;
    end;

```

```
{* bulanık kontrolörün giriş değerlerini ölçek değerlerine göre hesapla *}
e:=e/Se;
de:=de/Sde;
```

```
{* Mamdani minimum çıkarım metodu, en küçük üyelik değerini bul *}
for i1:=1 to 7 do
begin
me[i1]:=uye(i1,e);
for i2:=1 to 7 do
begin
mde[i2]:=uye(i2,de);
{*min üyelik değerini belirle*}
if me[i1]<mde[i2] then
m[i1,i2]:=me[i1]
else
m[i1,i2]:=mde[i2];
end;
end;
```

```
{* ağırlık merkezli durulama metodu *}
for i1:=1 to 7 do
begin
for i2:=1 to 7 do
begin
pay:=pay+m[i1,i2]*T[i1,i2];
payda:=payda+m[i1,i2];
end;
end;
u:=Su*(pay/payda);
if u<-30 then
u:=-30
else if u>30 then
u:=30;
until son
end.
```

ÖZGEÇMİŞ

02.05.1974 tarihinde Ordu'nun Fatsa ilçesinde doğdu. İlköğrenimini Fatsa Yassıtaş Köyü İlkokulunda yaptı. 1986 yılında Tekirdağ Namık Kemal Lisesi Ortaokulu, 1989 yılında Namık Kemal Lisesini bitirdi. Aynı sene İstanbul Teknik Üniversitesi Elektrik Mühendisliği Bölümüne başladı ve 1993 senesinde iyi dereceyle mezun oldu.



**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**