KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR DESTEKLİ BULANIK MANTIK DENETLEÇ TABANLI ISI DENETİMİ

YÜKSEK LİSANS TEZİ

Elo. ve Hab. Müh. Mustafa ÇAKIR

Anabilim Dalı: Elektronik ve Haberleşme Mühendisliği

Danışman: Yrd.Doç.Dr. Bekir ÇAKIR

HAZİRAN 1998

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR DESTEKLİ BULANIK MANTIK DENETLEÇ TABANLI ISI DENETİMİ

YÜKSEK LİSANS TEZİ

Mustafa ÇAKIR

Tezin Enstitüye Verildiği Tarih: 26.05.1998

Tezin Savunulduğu Tarih : 25.06.1998

Tez Danışmanı	Üye	Üye
Yrd.Doç.Dr Bekir ÇAKIR	Doç.Dr.İbrahim EKSİN	Yrd.Doç.Dr.Sıtkı ÖZTÜRK
(........................)	(İbrahim Eksin)	(........................)

HAZİRAN 1998

ÖNSÖZ ve TEŞEKKÜR

Bir çok araştırmacı, laboratuvarlarda ve üniversitelerde daha gelişmiş denetim kavramlarını araştırmak üzere yoğun bir çaba harcamaktadır. Bununla birlikte, keşfettikleri ilkeler, bunları zorunlu olarak uygulayacak kişilerle bir iletişim içinde olunmadığı takdirde gerçek değerlerini bulamayacaklardır.

Önceleri teorik bir araştırma konusu olan bulanık mantık, günümüzde pek çok uygulamada kullanılmaktadır. Kontrol düzenlerinde kullanılan klasik mantığın yetersiz kalması üzerine kontrol stratejisine yeni bir boyut kazandıran bulanık mantık teorisinde henüz tam bir sistematik geliştirilmemiştir ve bundan dolayıdır ki çoğu düzenleme özneldir. Bu tez çalışmasında simülasyonlar sonucu kazanılan operatör bilgisi doğrultusunda şekillenen bilgi tabanı ile bir bulanık kontrolör tasarlanıp örnek bir süreç üzerinde gerçekleştirilmiştir.

Bu çalışmamda yardımlarından dolayı sayın Doç.Dr. Bekir ÇAKIR'a teşekkür eder, bu çalışmanın bulanık mantık teorisi üzerinde çalışan tüm araştırmacılara yardımcı olmasını dilerim.

BİLGİSAYAR DESTEKLİ BULANIK MANTIK DENETLEÇ TABANLI ISI DENETİMİ

Mustafa ÇAKIR

Anahtar Kelimeler: Bulanık Mantık, Bulanık Denetleç, Üyelik Fonksiyonu

Özet: Bulanık denetleç PID gibi klasik denetleçlere alternatif olarak geliştirilmiştir. Bulanık denetimde amaç, matematiksel modeli tam olarak belirlenemeyen sistemlerin denetimi olduğundan sistem modeli ile ilgilenilmez. Klasik denetimde model vasıtası ile sağlanan kontrol bilgisi, bulanık denetlece deneyimli bir uzman tarafindan aktarılır. Bu aktarılan bilgiler ve sistemin giriş ile çıkış büyüklüklerinin belirlenmesi ile denetim stratejisi geliştirilir. Denetim stratejisine göre; bulanıklaştırma, bilgi tabanının üretilmesi ve bulanık denetim kuralları tanımlanır. Denetim sisteminin performansı da dikkate alınarak çıkartım ve durulaştırma metodu belirlenir.

PC AIDED FUZZY LOGIC CONTROLLER BASED TEMPERATURE CONTROL

Mustafa ÇAKIR

Keywords: Fuzzy Logic, Fuzzy Logic Controller, Membership Function

Abstract: Fuzzy logic controller has been developed for an alternative method of classic controller as PID. The aim in designing fuzzy logic controller is control of a process which can not been solved in form of mathematical model so FLC doesn't interest in mathematical model. Control knowledge, in classic method, provided from mathematical equations is described by expert operator in fuzzy control systems. Within forminf control knowledge and system I / O values control strategy s developed. According to a control strategy, fuzzyfication, producing knowledge base and fuzzy control rules are defined. Implicatıon and defuzzyfication method is chosen according to the control system's performance

İÇİNDEKİLER

SİMGELER DİZİNİ ve KISALTMALAR

α	:tetikleme açısı
atm	:atmosfer
c_p	:özgül ısı
de	:hatanın değişimi
e	:hata
E_{eff}	:efektif gerilim
E_m	:maksimum gerilim
FLC	:bulanık mantık kontrolör
k	:ısıl geçirgenlik
k_d	:türevsel kazanç
k_i	:entegral kazancı
k_p	:oransal kazancı
L	:kalınlık
μf	:üyelik fonksiyonu
MIMO	:çok girişli çok çıkışlı
$\mu_K(x)$	:x kümesinde tanımlı K üyelik fonksiyonu
N	:doğal sayılar kümesi
P	:oransal
PI	:oransal ve entegral
PID	:oransal, entegral ve türevsel
q	:ısıl akı
S(F)	:F bulanık kümesinin desteği
T	:sıcaklık
T(F)	:F bulanık kümesinin toleransı
τ_s	:zaman sabiti
T_s	:örnekleme zamanı
u	:kontrolörün toplam çıkışı
u_d	:kontrolör çıkışı
u_i	:referans seviyesi

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

BÖLÜM 1. GİRİŞ

Günümüzde bilimin amacı daha akıllı, insan zekasına, daha benzer şekilde işleyen sistemler geliştirmektir. Bu yüzden denetim düzenlerinde kullanılan klasik mantık bu amaç için yetersiz kalmaktadır. Çünkü klasik mantıkta, önermeler için yalnızca iki doğruluk değerinin var olabileceği kabul edilir 1 (doğru / var) , 0 (yanlış / yok). Hava ya sıcaktır ya da soğuktur. Oysa insan aklının işleyişi böylesine katı kurallara bağlı değildir. Örneğin sıcak ve soğuk sıfatları, kişiden kişiye değişebilir, ve ya kişinin kimi zaman sıcak kimi zaman da soğuk kabul ettiği değerler birbirinden farklı olabilir.

Klasik mantıktaki kabulün gerçek dünyada, doğal dillerle karşılaşılan önermeleri değerlendirmedeki yetersizliğinin anlaşılması üzerine 1920'li yıllarda Polonyalı matematikçi J. Luksiewiecs tarafından "Çok Değerli Mantık" ortaya atılmıştır.

Çok değerli mantığa göre önermeler, ikiden fazla sayıda doğruluk değeri alabilmektedir. Dolayısıyla her şey bir "derecelendirme sorunudur". Bu ilkeyle yola çıkan bulanık mantıkçılar olayları 0 ve 1 gibi iki katı değer yerine 0 ve 1 arasında değişen bir çok esnek değerle açıklamışlardır. Kuantum fizikçisi Max Black "Philosopy of Science" ta 1937 yılında yayımlanan bir yazısında liste ya da nesnelerden oluşan kümelere çok değerli mantık kurallarını uygulayarak ilk bulanık küme eğrilerini çizmiş oldu.

Bundan aşağı yukarı 30 yıl sonra University of California of Berkeley' de elektrik mühendisliği bölüm başkanı olan Lotfi A. Zadeh, bu alana adını veren "Bulanık Kümeler" adlı çığır açıcı yazısını yayınladı. Zadeh bu kümenin tüm nesnelerine Luksiewiecs'in mantığını uygulayarak bulanık kümeler için eksiksiz bir cebir geliştirdi.

Önceleri sadece teorik bir araştırma konusu olan bulanık mantık, daha sonra pek çok uygulama alanı bulmuştur. Bilgisayar bilimleri, denetim, tıp, sosyal bilimler, yönetim bilimleri, uzman sistemler, yapay zeka konuları, bulanık mantığın uygulama alanları arasında en belirgin olanlarıdır.

1970'li yılların ortalarında Londra'daki Queen Mary College'den Prof. Ebrahim H. Mamdani ve arkadaşları, bulanık küme teorisinin denetim sistemlerine uygulamaları üzerine çalışmışlar ve bulanık denetleç yapısını kurmuşlardır.

Bulanık denetleçler de zaman içinde çeşitli gelişmeler göstermiş, pek çok değişik türü kullanıma sunulmuştur. Örneğin sadece hataya ve türevine bakan denetleçler olduğu gibi klasik PID denetleçler, hatanın entegralini çeşitli şekillerde işleyen yapılar da bulunmaktadır.

Bulanık mantık elektrik makineleri denetimine de bir çok olumlu özellik getirmiştir. Elektrik makinelerinin denetimi, çoğu kez karmaşık modellere dayanır ve parametrelerin değişiminden etkilenir. Oysa bulanık denetleçler kendi özelliklerinden dolayı dinamik model gerektirmezler ve parametre değişimlerinden etkilenmezler. Bulanık denetimin uygulanabilmesi için öncelikle sistemi gözetleyerek elde edilen bilgilerle, giriş ve çıkışların bir uzman tarafından bulanıklaştırılması gerekir. Bu arada girişler ve çıkışlar arasında bir kural tablosu kurmak gereklidir Çok parametreli denetleçlerde bazılarının bulanıklaştırılması yeterlidir.

Aşağıda bulanık mantığın geçirdiği önemli aşamalar kronolojik sıraya göre verilmiştir.

1965 Bulanık küme kuramı (Prof.. Lofi A. Zadeh, California)
1966 Bulanık mantık (Dr. Peter N. Marinos, Bell Laboratuvarı)
1972 Buhar türbini denetiminde bulanık mantık uygulaması (Prof. E. H. Mamdani Queen Mary College, University of London)
1980 Çimento sanayiinde uygulama (F.L. Smidth, Kopenghoy, Danimarka)
1987 Senabi metrosunda otomatik tren denetimi (Hitachi)
1988 Hisse senedi portföyü için uzman sistemler (Yamaichi Sequrity)

Bulanık mantık denetleyiciler konusunda teorik çalışmaların devam etmesi insanların günlük yaşamında söz sahibi olacak kadar girmesine engel olamamıştır. Özellikle

Japonya'da bulanık denetimin kullanıldığı ev aygıtları ve elektronik ürünler güncel yaşamın bir parçası oldular. Japonya Uluslararası Ticaret ve Sanayi Bakanlığı'nın bir tahminine göre Japonya 1992'de 2 milyar dolar değerinde bulanık mantık ürünü üretmiştir. Bu tutarın 2000 yılına kadar 13 milyar dolara çıkması beklenmektedir.

BÖLÜM 2. BULANIK MANTIK

Fuzzy anlam olarak dilimizde "bulanık", "hayal, meyal" kelimeleri ile ifade edilebilir. Bulanıklık, bir düşüncenin tanımında veya bir kelimenin anlamında bulunabilen belirsizliktir. Bu terim 1965'de Prof. Zadeh tarafindan ilk defa ortaya atıldığında olayların olma olasılığından çok oluşum derecesiyle ilgilenen ve yapay zeka, yapay sinir ağları gibi konuları içerisine alan bir konuya işaret etmekteydi.

Giriş kısmında da belirtildiği gibi bulanık mantık insanın düşünüş birimine oldukça yakındır. Geleneksel mantık ise biraz daha farklıdır, burada bir kümeyi oluşturan elemanlar kesin olup kümenin ya elemanıdır ya da değildir. Oysa bulanık mantıkta durum böyle değildir. Bulanık mantıkta klasik mantıktaki sıcak-soğuk, açık-kapalı, hızlı-yavaş gibi keskin üyelik değerleri az, biraz, çok gibi dilsel niteleyicilerle yumuşatılarak insan düşüncesine benzetilmesi sağlanmıştır.

1920'lerdeki mantıkçıların "Her şey bir derecelendirme sorunudur" fikri bulanık mantığın temelini oluşturmaktadır. Diğer bir değişle bulanık mantık; olayların oluşum olasılığından çok oluşum derecesiyle ilgilenir. Bundan dolayı bazı bilim adamları bulanık mantığı olasılığın bir devamı olarak düşünmüşlerdir. Fakat olasılık ve bulanıklık birbirinden oldukça farklı kavramlardır. Olasılık, bir şeyin olup olmayacağını ölçer. Rasgelelik fikri ile sembolize edilen doğa olaylarına bağlı bir belirsizliği vardır. Bulanıklık ise bir olayın ne dereceye kadar olduğunu bir koşulun ne dereceye kadar var olduğunu ölçer.

Bulanık denetimin esasını önermelere konu olan kavramın, temsil ettiği özellikteki derecesi teşkil etmektedir. Bu önermeler, o sistemi daha önce kullanmış insanın deney ve tecrübelerine dayanmaktadır. Bir insanın bir aracı veya bir sistemi kullanması için o sistemin matematik modelini bilmesi gerekmez. Sistem, klasik denetim yöntemleri ile çok zor denetlenebilen bir sistem olabilir. Bu durumda onu başarıyla kullanan bir insanın tecrübelerinden yararlanmak kaçınılmaz olur.

Ancak bu kurallar da kişiden kişiye göre değişebilir. Bu nedenle tam denetim kuralları oluşturulurken birden fazla insanın deneyimleri beraber ele alınabilir.

Ayrıca bir sistemin bulanık denetimini yapabilmek için elde yeter sayıda mantık önermeleri bulunmalıdır. Bu önermelerin sayısı bazı hallerde onlarla ifade edilebilir. Bir diğer önemli zorluk, denetlenecek sistemin bir uzman gözüyle değerlendirme gerekliliğidir. Ancak yine de bütün bu zorluklarına rağmen bir kere bu önermeler oluşturulabiliyorsa, ondan sonra denetim için istemin yapısını bilme zorunluluğundan kurtulma önemli bir avantajdır.

2.1. Avantajlar ve Dezavantajlar

Bulanık mantık teorisinin, doğa (hava, doğal biçimler, okyanuslar gibi) veya insan yapısı (ekonomi, borsa veya seçimler gibi) sistemlerin modellenmesi ve denetimi için en iyi olduğu bilim adamlarınca savunulmaktadır.

Peki bulanık mantığın kullanılmasının doğru olmayacağı alanlar var mıdır? Şu şekilde bir ayrım yapılabilir. Geleneksel optimal denetim denklemleri elde edilmiş veya tamamı ile yeterli olan sistemlerde bulanık bir yaklaşım kullanmak pek tavsiye edilmez. Fakat istenirse karşılaştırma yapmak için kullanılabilir.

Bulanık mantığın

1- Geleneksel yöntemlerle yeterli doğrulukta modellenemeyen çok karmaşık sistemlerde,
2- Önemli ölçüde lineer olmayan sistemlerde,
3- Başlangıç koşullarında, girişlerinde veya tanımlarında belirsizlik olan sistemlerde kullanılması en iyi çözümü verir.

Bu nitelikleri taşıyan uygulamalar için gerekli yöntemleri içerecek şekilde bulanık denetim sistemleri geliştirilmiştir. Bu modelleme, değerlendirme, optimizasyon, karar verme, denetim, teşhis ve bilgi gibi uygulamalar olabilir.

Bulanık denetim, yapay zeka ve yönetim gibi çok çeşitli alanlarda denenmiştir. Ayrıca bulanık mantık teorisi fikrini geniş alanlara yaymak için projeler vardır. Bulanık sistem teorisini belirsiz düşünce ve karar süreçlerinin gelişen modellerine ait başlama noktası kabul ettiğimiz için aşağıdaki uygulama alanları geliştirilebilir.

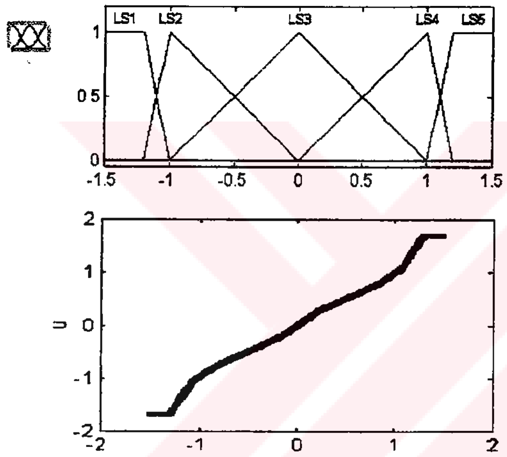
1- Yönetim ve sosyal problemler için kullanılabilen insan modellerinin yapılması,
2- Otomasyon ve bilgi sistemlerinde kullanım için yüksek derecede insan yeteneklerinin taklidi,
3- İnsan ve makineler arasındaki insan orijinli arabirimlerin oluşturulması,
4- Diğer sosyal ve yapay zeka uygulamaları (risk analizi ve tahmin, fonksiyonel aygıtların gelişimi) vs.

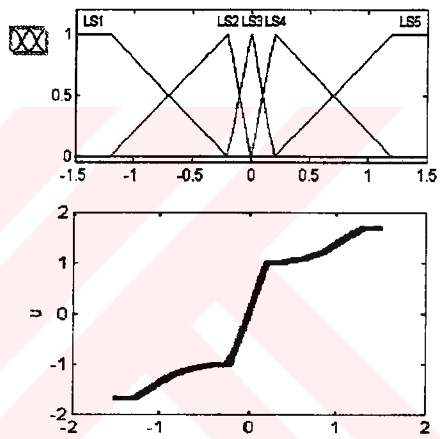
2.2. Bulanık Ölçeklendirme

Sistemden istediğimiz çıkış ile sistemin uyguladığımız denetim girişine verdiği cevap arasındaki farka hata denir ve yeni uygulanacak denetim girişleri bu hata ile hatadan türetilen başka işaretlere göre değişir. Bir hataya "büyük", hatanın değişim hızına "hızlı" diyebilmek için o sistemde bu gibi büyüklüklerin hangi değerler arasında yer aldığını bilmemiz gerekir. Bu alt ve üst sınırlar, çalışma aralığını oluşturur. Bulanık denetimde işlenecek bütün bilgilerin hangi aralıklarda değiştiğinin bilinmesi gerekir.

Bulanık denetimde işaretler "büyük","küçük" tanımları ile ayrımlanabileceği gibi "çok büyük" ve benzeri tanımlarla da kavramsal ifade sayısı arttırılabilir. Kavramsal seviye sayısının artması işaretleri izlemedeki hassasiyeti arttırırken buna karşılık işlem hızını düşürür. Bir sistem için uygun seviye, deneysel sonuçların sistemden istenen performans ile karşılaştırılması sonucu bulunur.

Ayrık seviyelendirme, basit olarak alt ve üst sınırların belirlediği çalışma aralığının kavramsal seviye sayısına bölümü ile yapılabilir. Buna doğrusal ayrık seviyelendirme denir. Hata ve diğer alt giriş değişkenleri çalışma aralığının her yerinde değerler alıyorsa, veya bu aralık için olasılık yoğunluk fonksiyonu her nokta için aynı ise "doğusal ayrık seviyelendirme" kullanılmalıdır. Ancak bazı özel durumlarda sistem hep belli bir çalışma noktası civarında çalışıyor ve hata çoğunlukla alt ve üst sınırların belirlediği aralıktan daha küçük bir aralık içinde kalıyor, bu aralığın dışına çok kısa anlar için çıkıyorsa o zaman doğrusal ayrık seviyelendirme yeterli olmamaktadır. Ayrıca çıkışta bir kavramın daha ağırlıklı olması isteniyorsa, kural tabanı dışında seviyelendirme düzeyleri de değiştirilebilir.





Şekil 2.1. Kavramsal Seviyelendirme ile Çıkış Arasındaki İlişki

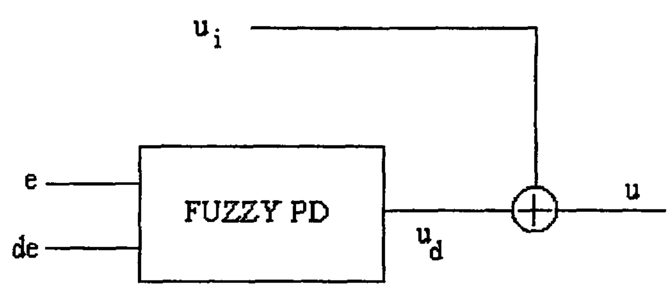
BÖLÜM 3. BULANIK DENETLEÇ TİPLERİ

3.1. Birinci Tip Denetleç

Bu tip denetleçlerde bulanık önerme “ eğer E ve C ise öyleyse U “ biçimindedir. Hata ve hatanın türevinin alacağı değerlere göre bir çıkış işareti oluşturulacaktır. Bu yapı bulanık PD denetlece denktir. Burada problem, bu yapıya bulanık olarak entegral işlevinin nasıl ekleneceğidir. Her ne kadar sistem hakkında pek az şey biliniyorsa da sürekli hal hatasından kolayca sistem tipini belirlemek mümkün olabilir. Eğer sistem tipi 0 ‘ dan büyük ise bu gibi durumlarda entegral işlevine gerek kalmaz ve algoritma denklem (3.1) ‘deki yapıyı alır:

$$u=u_d+u_i \tag{3.1}$$

Burada u denetlecin toplam çıkışı, u_d bulanık denetimden ve u_i önceden bilinen referans ve k_p sistemin kazancı olmak üzere r / k_p sayısından gelen çıkıştır. Bu yapının blok diyagramı şekil 3.1.‘de verilmiştir.



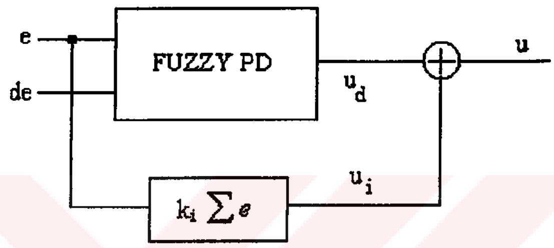
Şekil 3.1. Birinci Tip denetleç

3.2. İkinci Tip Denetleç

Bazı durumlarda sistemin statik kazancının bilinmesi mümkün olmayabilir. Bu hallerde entegral işlevinin denetim yapısına sokulması istenir. Bu tip denetleçlerde, yapı bir önceki yapıdır, ancak burada;

$$u_i=k_i \sum e \tag{3.2}$$

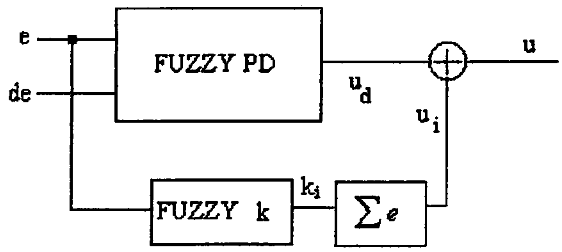
şeklindedir. Buna ait diyagram şekil 3.2.'de gösterilmiştir.



Şekil 3.2. İkinci Tip denetleç

3.3. Üçüncü Tip Denetleç

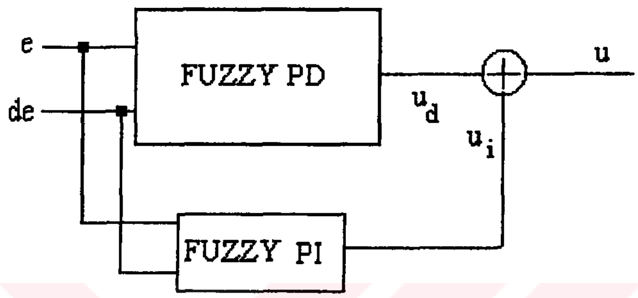
Birinci ve ikinci tip denetleçlere tam anlamıyla bulanık gözüyle bakılamaz; çünkü bulanık olmayan bileşenler içermektedirler. Bir ileri adım entegral işlevinin de bulanıklaştırılmasıdır. Buna ilişkin diyagram şekil 3.3.'de verilmiştir.



Şekil 3.3. Üçüncü Tip denetleç

3.4. Dördüncü Tip Denetleç

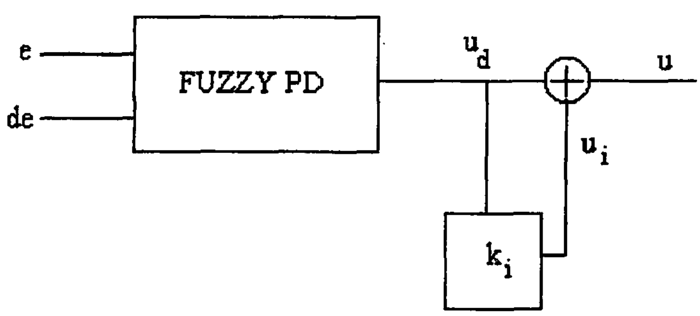
Bu tip denetleçte önermenin yapısı " eğer e(t) ve eğer de(t) ise du(t)" biçimindedir. Bulanık küme kullanılarak " eğer E ve eğer C ise DU " biçiminde de yazılabilir. Bulanık PD denetimden tek farkı çıkış büyüklüğü sisteme uygulanacak sayı değil, bu sayının değişim hızıdır (türevidir). Dolayısıyla çıkış, bu sayının entegrali alınarak bulunur. Bu denetlece ait diyagram şekil 3.4.de verilmiştir.



Şekil 3.4. Dördüncü Tip denetleç

3.5. Beşinci Tip Denetleç

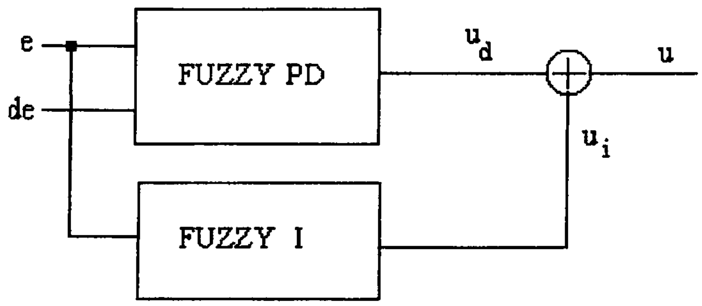
Bu denetleçte entegrasyon, hataya göre değil, bulanık PD kısmının çıkışına göre yapılır. Buna ait diyagram şekil 3.5.'de gösterilmiştir.



Şekil 3.5. Beşinci Tip denetleç

3.6. Altıncı Tip Denetleç

Bir diğer yöntem, dördüncü tip denetleci basitleştirmektir. Bulanık entegral işlevi "eğer E öyleyse DU" biçimine getirilirse bu amaç gerçekleşir. Bu denetlece ait diyagram ise şekil 3.6. da verilmiştir.



Şekil 3.6. Altıncı Tip denetleç

Bulanık denetleçler; hata, hatanın türevi ve entegraline göre önceden hesaplanmış bir çıkış üreten kombinezonsal bir devre gibi düşünülebilir: Giriş sinyalleri hata ve bundan türetilen sinyaller olan bir kombinezonsal devre. Burada en önemli problem bu karar tablosunu oluşturan katsayıların bulunmasıdır.

BÖLÜM 4. KÜMELER ve BULANIK BAĞINTILAR

4.1. Kümeler

4.1.1. Keskin Kümeler

Klasik anlamda bir küme, kümenin elemanı olarak adlandırılan nesnelerin bir koleksiyonudur. Bu küme değişik dünyalardan olabilir. Örneğin:

$$K=\left\{x \in N \mid 3 \langle x \langle 10\right\} \quad (4.1)$$

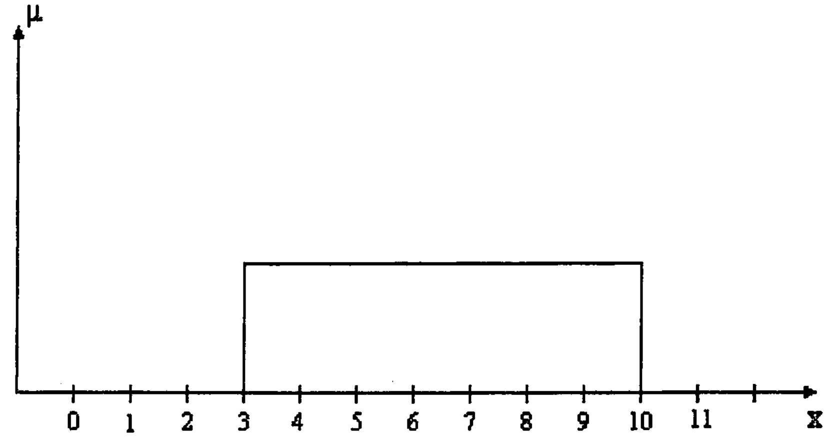
Eleman olarak 10'dan küçük sayıları içeren özelliğe sahip olan tüm doğal sayılar kümesi kümenin karakteriksel fonksiyonu veya üyelik fonksiyonu μ_K , K kümesinde mevcut olan X temel kümesinin bütün x elemanları için 1 değerini ve elemanı olmayan her x elemanı için 0 değerini alır.

$$\mu_K = x \rightarrow \{0,1\} \quad (4.2)$$

$$\mu_K(x) = \begin{cases} 1 & \text{eğer } x \in K \\ 0 & \text{eğer } x \notin K \end{cases} \quad (4.3)$$

Derece iki değerlidir. Yani eleman ya kümenin içindedir ya da içinde değildir.

Keskin kümeler için 3 temel işlem vardır. Birleşme, kesişim ve tümleyen.



Şekil 4.1. K Kümesinin Karakteriksel Fonksiyonu

4.1.2 Bulanık Kümeler

Bulanık küme kavramı bir bakıma keskin kavramının (klasik küme) genelleştirilmiş hali olarak düşünülebilir. Eğer bir x değişkeninin alabileceği tüm değerler aralığına X dersek X={x}'e bulanık küme teorisinde söylem uzayı (universe of discourse) adı verilir. Böylece her bulanık küme için, söylem uzayı içerisinde X, 0 ve 1 arasında her değeri alan ve adına üyelik fonksiyonu adı verilen $\mu_{(x)}$ ile tanımlanmıştır.

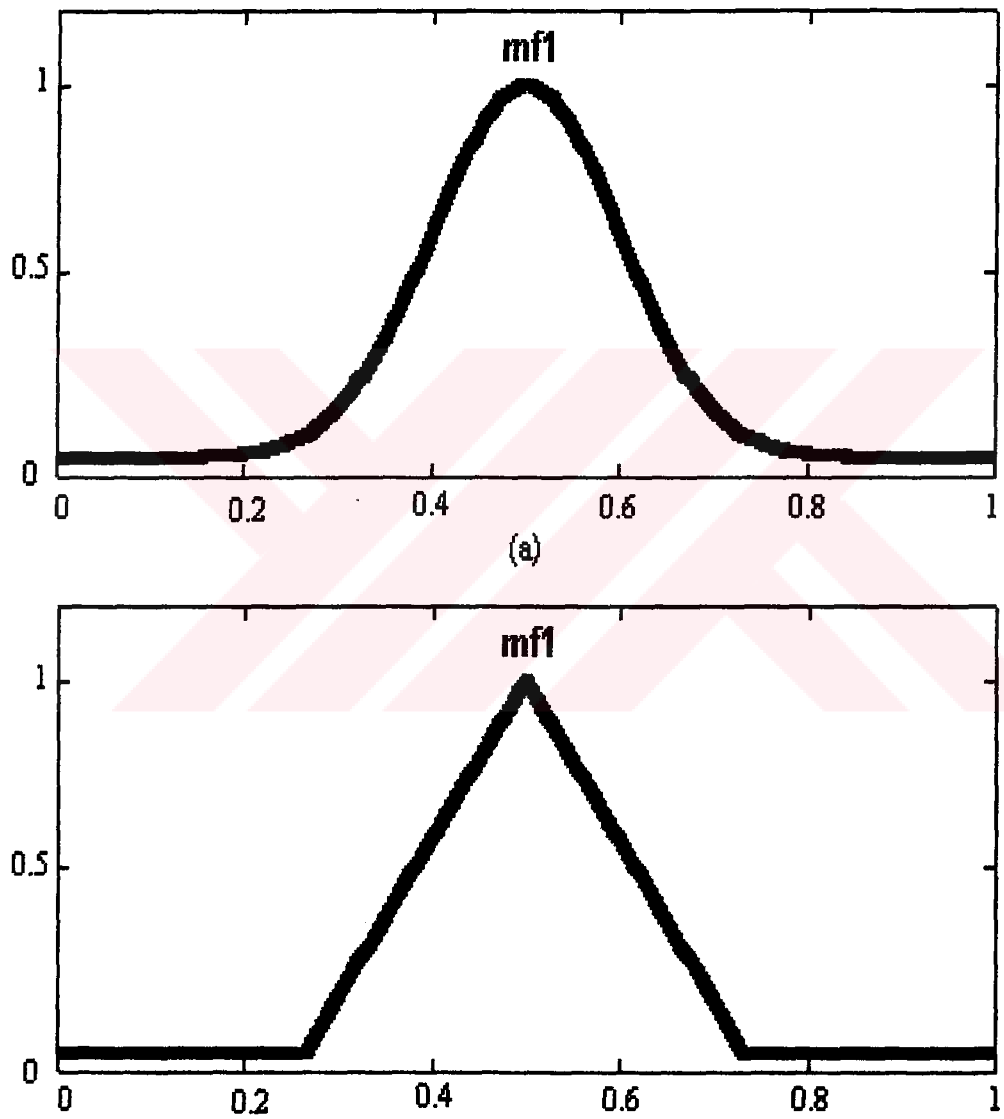
Üyelik fonksiyonları, fonksiyonların yapısına göre iki ayrı şekilde incelenmektedir.

1-Sürekli Üyelik Fonksiyonları:

Araştırmacılar tarafından bir çok matematiksel fonksiyon bu tip üyelik fonksiyonu üretmek için kullanılmıştır. Örneğin şekil 4.2.a 'da çan eğrisi biçiminde sürekli bir üyelik fonksiyonu görülmektedir.

2-Yarı Sürekli Üyelik Fonksiyonları:

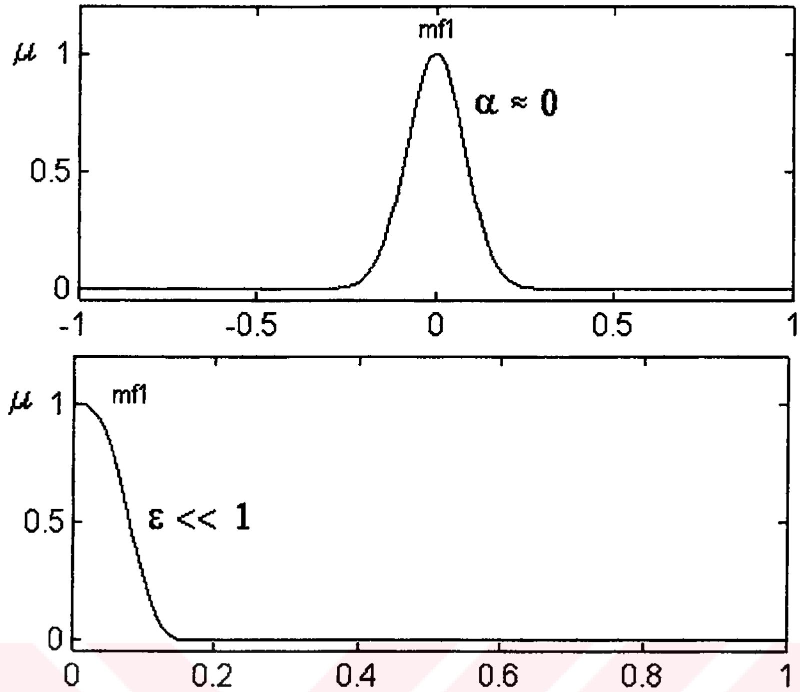
Bu türdeki fonksiyonların en önemli özelliği bir kaç değer ile tanımlanabilmeleridir. Örneğin şekil 4.2.b.'de sadece üç değerle tanımlanan üçgen üyelik fonksiyonu görülmektedir. Benzer şekilde bir yamuk üyelik fonksiyonu çizmek için dört değere ihtiyaç olduğu açıktır.



(b)

Şekil 4.2. a) Sürekli Üyelik Fonksiyonu b) Üç değerle tanımlanan Üçgen Üyelik Fonksiyonu

Üyelik ağırlığı belirli bir değerin bulanık küme içerisinde yer almasının güvenirliğinin veya eminliğinin bir işaretidir. (Şekil 4.3.)



Şekil 4.3. $\alpha \approx 0$ ve $\varepsilon \ll 1$'in Bulanık Gösterimi

Üyelik işlevlerinin biçimi, denetlenen sürecin özelliklerine göre değişik şekillerde olabilir. Genelde üçgen, çan ve yamuktur.

Bulanık kümede de tanımlama klasik kümelerde olduğu gibi elemanların listelenmesi yoluyla yapılır. Bu durumda sadece elemanların gösterimi yeterli değildir (Klasik kümelerde otomatik olarak üyelik fonksiyon değeri 1 ile sınırlıdır). Her bir eleman için kümeye ait olma derecesi de (üyelik fonksiyon derecesi) açıkça verilmelidir.

4.1.2.1. Bulanık Küme Gösterimi

Bu ifadeler altında bulanık kümeler şu şekilde tanımlanabilir.

$$F = \left\{ x, \mu_F(x) \mid x \in X \right\} \tag{4.4}$$

X kümesinin içindeki F, bulanık kümesidir. Denklem (4.5) ‘deki gösterim ise

$$\mu_F : x \rightarrow [0,1] \qquad (4.5)$$

F bulanık kümesinin üyelik fonksiyonu olarak ifade edilir. Fonksiyon, X temel kümesinin her bir x elemanına karşılık üyelik derecesini düzenler. Bir tek değer çifti $(x,\mu_F(x))$ singleton olarak bilinir. Temel kümenin listelenen bütün elemanları dışında kalanların üyelik derecesi sıfırdır.

Bulanık kümenin gösteriminde usul açısından belirli bir standart yoktur. Örneğin Zadeh’in yazım şekli bir singleton için $\mu_F(x)/x$ olarak literatüre girmiştir. Buna uygun olarak bulanık küme için

$$F=\left\{\mu_F(x)/x \,\middle|\, x\in X\right\} \qquad (4.6)$$

yazılır. Tekli elemanların listelenmesi Zadeh’in yazım şekline göre;

$$F=\mu_F(x_1)/x_1 + \ldots +\mu_F(x_n)/x_n=\sum_{i=1}^{n} \mu_F(x_i)/x_i \qquad (4.7)$$

yazılır. Burada “/ ” işareti; değer çiftini ,“+ “ ise sıralamayı karakterize eder.

F bulanık kümesi, tanımlı olduğu X uzayı sürekli ise;

$$F = \int_{X} A(x)/x \qquad (4.8)$$

ile ifade edilir

Bulanık küme kuramı içinde bir çok kavram tanımı vardır. Bu tanımlar genel olarak klasik kümenin genelleştirmesini gösterse de keskin olmayan kümeler için önem arz etmektedir.

Tanım 4.1. : F bulanık kümesi, x temel kümesinin içinde ise;

$$Y(F)=\max_{x \in X} \mu_F(x) \tag{4.9}$$

yükseklik olarak adlandırılır.

Tanım 4.2. : Üyelik fonksiyon derecesi 1 olan (Y(F)=1) F bulanık kümesine, normal bulanık küme adı verilir, aksi halde subnormaldir.

Keskin olmayan kavramlar sadece normal bulanık küme ile modellenecektir.

Tanım 4.3. : F bulanık kümesi, X temel kümesinin içinde ise;

$$S(F)=\left\{x \in X \mid \mu_F(x) \rangle 0\right\} \tag{4.10}$$

F kümesinin desteği 1 olarak adlandırılır. Bazen taşıyıcı ya da etki aralığı olarak da ifade edilir.

Tanım 4.4. : F bulanık kümesinin içinde ise;

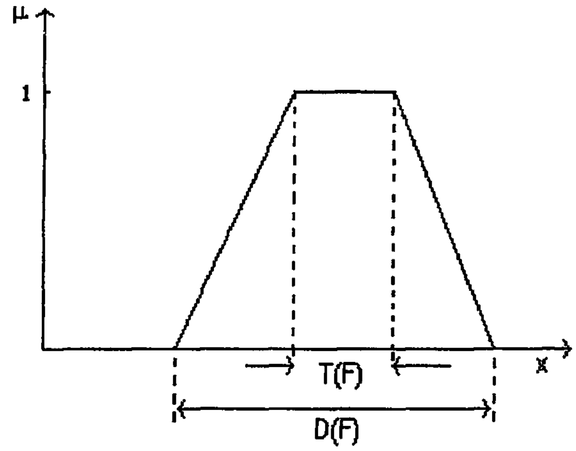
$$T(F)= \left\{x \in X \mid \mu_F(x)=0\right\} \tag{4.11}$$

ile F kümesinin toleransı ifade edilir.

Bir bulanık kümenin desteği, temel kümenin elemanlarından üyelik fonksiyon derecesi sıfırdan büyük olan elemanları içerir. Buna karşılık tolerans ise üyelik derecesi 1 olan elemanlardan oluşur.

Şekil 4.4.'deki yamuk biçimli kümede bu tanımlar belirtilmiştir.

Üçgen biçimli kümede tolerans sadece bir model değerden oluşur. Singleton şeklindeki kümede tolerans ve destek aynı model değerden meydana gelir. Klasik kümelerde ise, tolerans ve desteğin aynen eşit olduğu görülür.

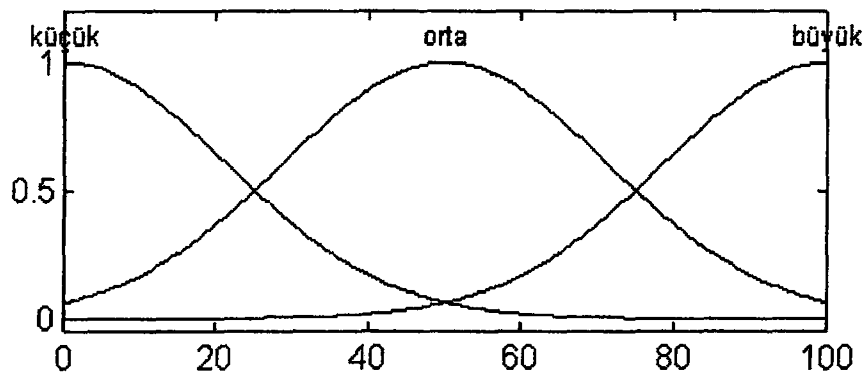


Şekil 4.4. Yamuk Biçimli Üyelik Fonksiyonu

Tanım 4.5. : Bir destek kümesi gerçel bir sayı olduğu zaman ve herhangi bir [a,b] aralığında tüm $x \in [a,b]$ için aşağıdaki formül geçerli ise;

$$\mu_A(x) \geq \mu_A(a) \wedge \mu_A(b) \qquad (4.12)$$

A bulanık kümesi konvekstir. Şekil 4.5.'de konveks kümeler görülmektedir.



Şekil 4.5. Konveks Üyelik Fonksiyonları

4.1.2.2. Bulanık Kümelerde İşlemler

A ve B, x uzayında tanımlı iki bulanık küme ise aşağıdaki küme işlemleri tanımlanabilir;

Tanım 4.6. : A ve B gibi iki bulanık kümenin bileşimi C bulanık kümesidir. C=A∪B ya da A+B biçiminde yazılır. Toplam kümesinin üyelik fonksiyonu şu şekilde tanımlanır.

$$A \cup B = \int_0^x (\mu_A(x) \vee \mu_B(x)) / x \tag{4.13}$$

$$\mu_c(x) = \max\{\mu_A(x), \mu_B(x)\} \tag{4.14}$$

Tanım 4.7. : A ve B gibi iki bulanık kümenin kesişimi C, C=A∩B şeklinde yazılır. Üyelik fonksiyonu şu şekilde ifade edilir;

$$A \cap B = \int_0^x (\mu_A(x) \wedge \mu_B(x)) / x \tag{4.15}$$

$$\mu_c(x) = \min\{\mu_A(x), \mu_B(x)\} \tag{4.16}$$

Tanım 4.8. : A bulanık kümesinin tümleyeni $\overline{A}$ olarak gösterilir ve üyelik fonksiyonu için;

$$\overline{A} = \int 1 - \frac{\mu_A(x)}{x} \tag{4.17}$$

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \tag{4.18}$$

şeklinde ifade edilir.

Bulanık kümelerin tanımlanma biçimlerini, niteliksel ifadeler kullanarak arttırmak mümkündür. "Tam;oldukça" gibi bir A bulanık kümesi düşünelim;

$$\text{Tam } A=\sum_i \frac{\mu_A^2(x_i)}{x_i} \tag{4.19}$$

Oldukça A ise, söylem uzayı boyunca C miktarı kadar hareket tanımlar.

$$\mu_{\text{oldukça}}\, A(x)=\mu_A(x+c) \tag{4.20}$$

İkili bulanık kümenin kesişiminin oluşması ya da ikili keskin olmayan ifadenin AND bağlantısını gerçekleştirmek için operatörler T-norm olarak ifade edilir. Bileşim kümesi veya OR-bağlantısının gerçekleşmesi için lojiksel eşdeğerler S-norm veya T-norm olarak ifade edilir.

Tanım 4.9. : Bir t-normu ile iki argümanlı bir fonksiyon tanımlanır.

öyle ki;

$$t:[0,1]\times[0,1]\rightarrow[0,1] \tag{4.21}$$

a- $y \geq x$, $z \geq w$ için $y \, t \, z \geq x \, t \, w$ özelliği gösterir.
b- değişim özelliği vardır
c- birleşim özelliği vardır
d- $x \, t \, 0=0$; $x \, t \, 1 = x$ koşulunu sağlar.

Tanım 4.10 : Bir s normu ile iki argumanlı bir fonksiyon tanımlanır. Öyle ki;

$$s: [0,1]\times[0,1]\rightarrow[0,1] \tag{4.22}$$

a- $x \leq y$, $w \leq z$ için $x \, s \, w \leq y \, s \, z$ koşulunu sağlar.

b- değişim özelliği vardır

c- birleşim özelliği vardır

d- x s 0=x ; x s 1 =1 koşulunu sağlar.

Tablo 4.1. T- ve S- Normları

"AND" T-norm $T(\mu_A(x), \mu_B(x))$	**"OR"** S-norm $S(\mu_A(x), \mu_B(x))$
Minimum $MIN(\mu_A(x), \mu_B(x))$	**Maksimum** $MAX(\mu_A(x), \mu_B(x))$
Drastik (Etkili) Çarpım $MIN(\mu_A(x), \mu_B(x))$ Eğer $MAX(\mu_A(x), \mu_B(x))=1$ ise 0 aksi halde	**Drastik (etkili) Toplam** $MAX(\mu_A(x), \mu_B(x))$ Eğer $MIN(\mu_A(x), \mu_B(x))=0$ ise 1 aksi halde
Sınırlı Fark (Lukasiewicz-AND) $MAX(0, \mu_A(x)+ \mu_B(x)-1)$	**Sınırlı Toplam** (Lukasiewicz—OR) $MIN(1, \mu_A(x)+ \mu_B(x))$
Einstein-Çarpımı $(\mu_A(x)\ \mu_B(x))/(2-(\mu_A(x)+\mu_B(x)\ \mu_A(x)\ \mu_B(x)))$	**Einstein-Toplamı** $(\mu_A(x)+\mu_B(x))/(1+\mu_A(x)\ \mu_B(x))$
Hamacher-Çarpımı $(\mu_A(x)\ \mu_B(x))/((\mu_A(x)+\mu_B(x)-\mu_A(x)\ \mu_B(x))$	**Hamacher-Toplamı** $(\mu_A(x)+\mu_B(x)-2(\mu_A(x)\mu_B(x))/(1-\mu_A(x)\mu_B(x))$
Cebirsel-Çarpım $\mu_A(x)\ \mu_B(x)$	**Cebirsel-Toplam** $(\mu_A(x) + \mu_B(x)-(\ \mu_A(x)\ \mu_B(x))$
Yager-Operatörü $1-MIN((1-\mu_A(x))^p+(1-\mu_B(x))^p)^{1/p},1)$ $p \in R$	**Yager-Operatörü** $MIN\ ((\mu_A(x)^p+\mu_B(x)^p)^{1/p},1)$ $p \in R$

Bir A kümesinin, bir B bulanık kümesi tarafından içerilmesi için $\forall x \in A(x) \leq B(x)$ koşulunun sağlanması gerekmektedir. Bu tanıma ek olarak A'nın bir x noktasında B tarafından içerilme seviyesinin tanımı yapılabilir;

$$|A \subset B|(x) = A(x) \phi B(x) \qquad (4.23)$$

öyle ki;

$$A(x) \phi B(x) = \text{surp}[c \in [0,1] \mid A(x) \text{ t } c \leq B(x)] \qquad (4.24)$$

4.2. Bulanık İlişkiler

R bulanık küme ilişkisi

R: $x \times y \rightarrow [0,1]$ şeklinde gösterilir.

Bulanık ilişki, x ve y uzaylarından seçilen bir (x,y) çiftini aralarındaki bağıntı oranında $[0,1]$ aralığında bir değere karşı düşürür.

Tanım 4.11. : $x \times y$ uzayında tanımlı ve belirli bir R ilişkisi için her "$x \in X$"e karşı düşen bir $y \in Y$ bulma işlemine "Kompozisyon İşlemi " adı verilir. Bu da girişi bilinen bir sistemin, bu girişe ait çıkışının hesaplanması işlemine denktir.

Kompozisyon işlemi aşağıdaki gibi tanımlanır.

1- sup-t:
$x \in X$ bulanık kümesi ve $R \in F(x,y)$ bulanık ilişkisinin sup-t kompozisyonu;

$$Y(y) = \text{supr}_x[X(x) \text{ t } Y(y)] \qquad (4.25)$$

ile tanımlıdır.

2- inf-s:

$x \in X$ bulanık kümesi ve $R \in F(x,y)$ bulanık ilişkisinin inf-s kompozisyonu;

$$Y(y)=(X \diamondsuit Y)/y=\inf r_x[X(x) \text{ s } R(x,y)] \qquad (4.26)$$

ile tanımlıdır.

Bulanık bağıntı sırasıyla farklı X ve Y söylem evrelerindeki bulanık A giriş değişkeniyle, bulanık B çıkış değişkeni arasındaki koşullu önerme, dilsel bir önermeyle verilir.

$$A \rightarrow B \text{ veya eğer A o halde B} \qquad (4.27)$$

Bu önerme "{giriş koşulu}",{çıkış koşulu B}'yi gerektirir diye çevrilir. Burada A neden, B ise sonuçtur.

Bu gerektirmeli bağıntı "R", A ve B kümelerinin kartezyen çarpımı olarak gösterilir.

$$R=A \times B \qquad (4.28)$$

Sonlu kümeler için R aşağıda gösterilen üyelik fonksiyonuna sahiptir.

$$\mu_R(x,y)=\mu_{A\times B}(x,y)=\min[\mu_A(x),\mu_B(y)] \qquad x \in X : y \in Y \qquad (4.29)$$

Alternatif olarak, sürekli diziler için;

$$R=A \times B \int_{x \times y} \frac{\mu_A(x) \cap \mu_B(y)}{(x,y)} \qquad (4.30)$$

Bu eşitlik genellikle matris biçimindedir. Bundan dolayı

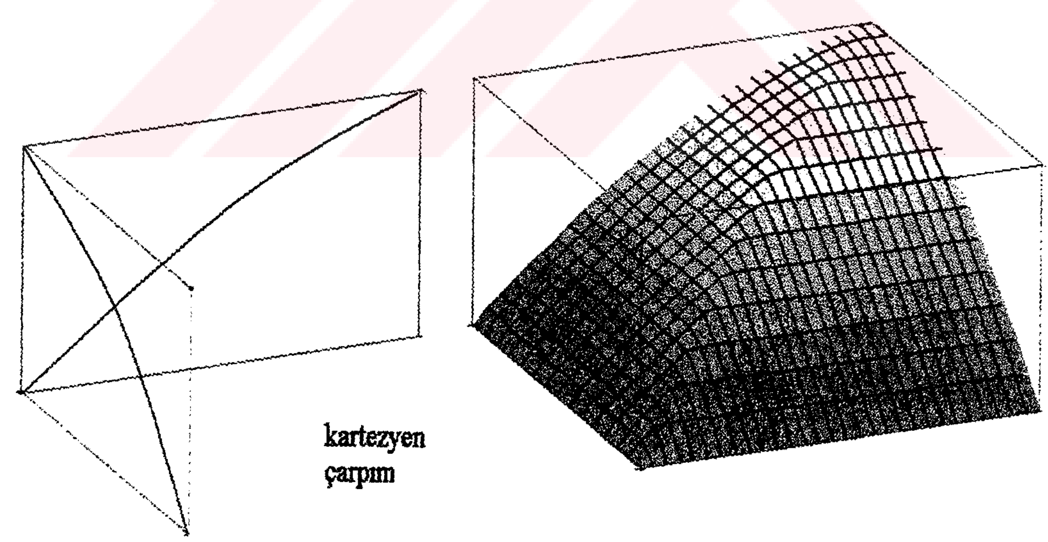
$$R = A \times B = \sum_{x \times y} \frac{\mu_A(x) \cap \mu_B(y)}{(x,y)} \tag{4.31}$$

ya da;

$$\begin{array}{l} A \times B = \min[\mu_A(x_1),\mu_B(y_1)] \ldots\ldots\ldots\ldots \min[\mu_A(x_1),\mu_B(y_n)]. \\ \quad . \\ \quad . \\ \quad . \\ \quad \min[\mu_A(x_m),\mu_B(y_1)] \ldots\ldots\ldots\ldots \min[\mu_A(x_m),\mu_B(y_n)] \end{array} \tag{4.32}$$

şeklindedir.

Burada $X=\{x_1,x_2,\ldots x_m\}$ ve $Y=\{y_1,y_2,\ldots y_n\}$ 'dir.



Şekil 4.6. Bulanık Bağıntının Grafiksel Gösterimi

Bulanık bir koşullu önerme ikiden fazla farklı söylem uzayı içerecektir. Üç farklı söylem evreni düşünelim. U,Y ve W; bunlara karşı düşen A,B ve C bulanık kümeleri olsun.

Eğer {A} o halde {B} o halde {C} ise

Bulanık bağıntı (4.33) eşitlinde gösterildiği gibidir.

$$R=A\times B\times C=\int_{U\times Y\times W} \frac{\mu_A(u)\cap \nabla\mu_B(y)\cap \mu_C(w)}{(u,y,w)} \qquad (4.33)$$

Bu ifade üç boyutlu bir ilişki matrisi olarak ifade edilir. (4.34)

$$A\times B\times C=\sum_{U\times Y\times W} \frac{\mu_A(u)\cap \mu_B(y)\mu_C(w)}{(u,y,w)} \qquad (4.34)$$

Bir denetim sisteminin formülasyonu sırasında her biri bir R^1 bulanık kümesine sahip olan, bir kaç bulanık koşullu önermenin kombinasyonunu oluşturmak gerekir. Eğer durum buysa, bütün bağıntıyı teşkil etmek için tek R^1 değerlerini birleştirmek gerekir.

Dolayısıyla;

$$R=R^1+R^2+...+R^N \qquad (4.35)$$

Burada R^i, i. yasa tarafindan oluşturulmuş bulanık kümedir. N yasaların sayısıdır.

Keskin olmayan "EĞER ... O HALDE" kuralının modellenmesi için gerçekleştirme şekillerinden en basit ve en çok kullanılanı MİN operatörüdür. Çıkartım mekanizması Mamdani çıkartım esasına dayanıyorsa MAX-MIN çıkartımından söz edilir. Cebirsel çarpım kullanımında ise MAX-PROD çıkartımı telaffuz edilir.

Bulanık çıkartım için diğer operatörlerin bazıları tablo 4.2.'de verilmiştir. Bu operatörlerin bir kısmı çok özel bulanık mantık uygulamaları ile ilgili olabilir.

Tablo 4.2. Bulanık Denetimde Rastlanmayan Implikasyon Operatörlerinden Bazıları

Bulanık-İmlikasyon $\mu_{A \Rightarrow B}(x,y)$	
Zadeh-implikasyon	$MAX(MIN(\mu_A(x), \mu_B(y)), 1-\mu_A(x))$
Lukasiewicz-implikasyon	$MIN(1, 1-\mu_A(x)+ \mu_B(y))$
Kleene-Dienes-implikasyon	$MAX(1-\mu_A(x), \mu_B(y))$
Gödel-implikasyon	1 Eğer $\mu_A(x) < \mu_B(y)$ $\mu_B(y)$ aksi halde
Sharp-implikasyon	1 Eğer $\mu_A(x) < \mu_B(y)$ 0 aksi halde

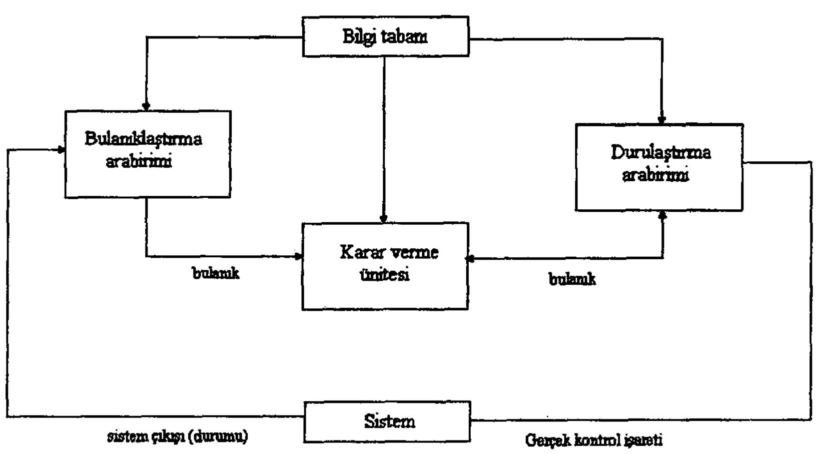
BÖLÜM 5. BULANIK DENETLEÇ

Bulanık denetleçler PID gibi klasik denetleçlere alternatif olarak geliştirilmiştir. Bulanık denetleçte amaç matematiksel modeli tam olarak belirlenmeyen sistemlerin denetimi olduğundan, sistemin modeliyle ilgilenilmez. Sistemle ilgili bilgiler sistemi denetlemekte olan bir operatörden edinilen denetim kuralları, denetlecin yapısına eklenir. FLC (Fuzzy Logic Controller-Bulanık Mantık Denetleç) otomatik denetim stratejisi içindeki uzman bilgisine dayanan dilsel denetim stratejisini dönüştürmek için kullanılır.

5.1. Bulanık Denetleç Yapısı

FLC'nin yapımında ilk olarak hedef belirlenmelidir. Buna göre bulanıklaştırma ve durulaştırma, bilgi tabanının üretilmesi ve bulanık denetim kuralları, bulanık gerektirmenin tanımı ve bulanık mantık mekanizmasının analizi belirlenir.

Bir bulanık denetleç dört ilkesel bileşenden oluşur (şekil 5.1.)



Şekil 5.1. Bulanık denetlecin Temel Yapısı

1.Bulanıklaştırma Arabirimi: Aşağıdaki fonksiyonları içerir.

a- Giriş değişkenlerinin değerlerini ölçer.

b- Giriş değişkenlerinin değerlerini ölçekleyerek uygun uzaylara karşı düşürür.

c- Giriş işaretini uygun dilsel değerlere dönüştüren bulanıklaştırma (bilgi tabanında tanımlanmış bulanık kümelerden birinin etiketi olan linguistic (dilsel)bir değişkene dönüştürür) fonksiyonunu üretir.

2. Bilgi Tabanı: Uygulama alan bilgisi ve mevcut olan denetim amaçlarını oluşturur. Bir " veri tabanı " ve dilsel kural tabanından oluşur.

a- Veri tabanı gerekli tanımları sağlar. Bunlar dilsel denetim kurallarını ve bulanık bilgi idaresini tanımlamak için kullanılır.

b- Kural tabanı, denetim amaçlarını ve uzmanların politikasını karakterize eder.

3.Karar Verme Ünitesi: FLC'nin özüdür. İnsanın bulanık kavramlarla düşünen karar verme yeteneğine yani bulanık çıkarım kurallarını kullanarak bulanık denetim hareketlerinden sonuç çıkarım yeteneğine sahiptir.

4. Durulaştırma Arabirimi: Aşağıdaki fonksiyonları sağlar.

a- Denetlecin çıkış işaretini ilgili uzaya karşı düşüren bir ölçekleme haritası oluşturur.

b- Çıkarılmış bulanık denetim hareketinden bulanık olmayan denetim hareketini oluşturur.

5.1.1. Bulanıklaştırma Arabirimi

Bulanıklaştırma doğal bir dildeki kesinliğin az olmasına ve belirsizliğe bağlıdır. Uygulamalarda gözlenen bilgi keskindir ve rasgeledir. Bulanıklaştırma için ilgili fiziksel büyüklüğün tanım aralığı ile işaret elde edeceği bulanık kümenin uzayının denk veya benzer olması gerekmektedir.

En sık kullanılan bulanıklaştırma algoritması, x_0 fiziksel büyüklüğü, üyelik fonksiyonu $\mu_A(x)$ olan bir A bulanık kümesine dönüştürür. Öyle ki, x_0 dışındaki noktalarda $\mu_A(x)=0$; x_0 noktasında ise $\mu_A(x)=1$ dir.

5.1.2. Bilgi Tabanı

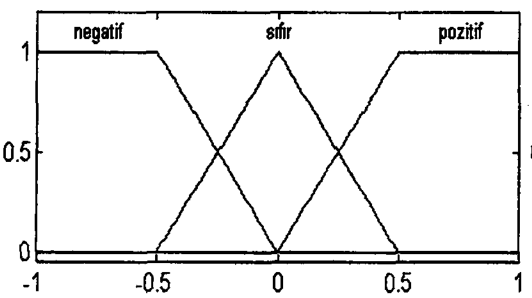
FLC'nin bilgi tabanı iki kısımdan oluşur. Veri tabanı ve bulanık denetim kural tabanı.

5.1.2.1. Veri Tabanı

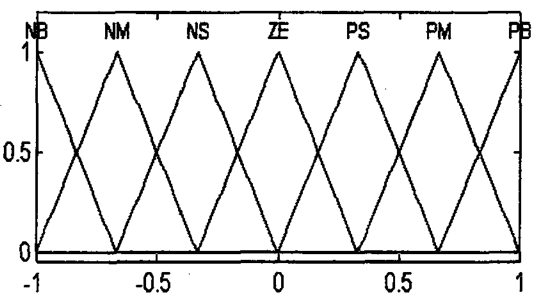
Bilgi tabanına ilişkin fikirler FLC'deki denetim kuralları ve bulanık bilgi değerlendirmesini karakterize etmek için kullanılır. Bu fikirler öznel olarak tanımlanır, deneyime ve operatör yargısına bağlıdır. Buna ek olarak bir terim kümesinin üyelik fonksiyonlarının doğru seçimi uygulama başarısı için önemli bir rol oynar. Bulanık kümenin (üyelik fonksiyonunun) seçimi özneldir.

Bulanık kümeleri üyelik fonksiyonları ile veya ayrık üyelik dereceleriyle tanımlamak olasıdır. Ne ölçüde hassas bir denetimin yapılmak istendiğine bağlı olarak belirli sayıda düzey saptanmalıdır. Sistemden elde edilen gerçek değerler, bir ölçekleme faktörü yardımıyla bu düzeylere karşı düşürülür. Düzeylerin sayısı arttıkça denetimin niteliği yükselir.

Temel bulanık kümeler (dilsel terimler), Negatif Büyük, Negatif Orta, Negatif Küçük, Sıfır, Pozitif Küçük, Pozitif Orta ve Pozitif Büyüktür.



Şekil 5.2. 3 Terimli Kaba Bulanık Kısım



Şekil 5.3. 7 Terimli Hassas Bulanık Kısım

5.1.2.2. Kural Tabanı

FLC'de bulanık sistemin dinamik davranışı; uzman bilgisine dayanarak dilsel tanımlı kural kümesiyle karakterize edilir. Uzman bilgisi genelde şu şekilde verilir.

EĞER (durum kümesini karşılar) O HALDE (sonuçlar kümesi) (5.1)

Eğer- O Halde kurallarının öncel ve sonuçları dilsel terimlerle birleştirilir. Bunlara bulanık durum ifadeleri denir. Bir bulanık denetim kuralı, bir bulanık koşul ifadesidir. Öncel, uygulama alanındaki koşul; sonuçsal ile denetim altındaki sistemin denetim hareketidir. Temel olarak bulanık denetim kuralları, denetim politikasını ve alan bilgisini ifade etmek için uygun bir yoldur. Bundan başka birkaç dilsel değişken öncelleri ve sonuçsalları kapsayabilir. Bu durumda sistem çok girişli-çok çıkışlı (MIMO; "multi input-multi output") bulanık sistem olarak ele alınacaktır.

Bulanık denetim kuralları,

EĞER R_1=x ; A_1 ve y ; B_1 ise O HALDE z ; C_1

EĞER R_2=x ; A_2 ve y ; B_2 ise O HALDE z ; C_2

.

.

.

EĞER R_n=x ; A_n ve y ; B_n ise O HALDE z ; C_n (5.2)

x,y; süreç durum değişkenleri ; z denetim değişkeni, A_i, B_i, C_i ; U,Y,W söylem uzayındaki x,y,z dilsel değerleridir. i=1,2,...,n

$$\mu R_i \cong_{(A_i \text{ ve} B_i \to C)} (u,v,w)$$
$$= [\mu_A(u) \text{ ve } \mu_B(v)] \to \mu_{Ci}(w) \quad (5.3)$$

$R_i \cong (A_i \text{ ve } B_i) \to C_i$: u×v×w 'deki bulanık gerektirmedir.

Denetim kurallarının oluşturulmasında dört yol vardır.

1- Uzman deneyimi ve denetim mühendisliği bilgisi: Bulanık denetim kuralları bulanık durum ifadeleri şeklindedir. Bu ifadeler önceldeki durum değişkenlerine ve sonuçsaldaki denetim değişkenleriyle ilişkilidir. Bulanık denetim kuralları, insan davranış ve karar analizinin karakterize edilmesi için doğal bir çalışma çerçevesi sağlar.

2- Operatörün denetim hareketlerine bağlı olarak: Bazı endüstriyel uygulamalarda tam bir matematiksek model elde etmek mümkün olmayabilir. Böyle uygulamalarda bir operatörün deneyimine ve sezgisine güvenilir. İlgili operatörün aksiyonları gözlenerek bulanık koşullu deyimler türetilir.

3- Sistemin bulanık modeline dayanarak: Sistemin dinamik karakteristiklerinin dilsel anlatımı, sistemin bulanık modelini oluşturur. Bulanık modelin oluşturulması zordur, ama kural tabanının güvenirliği ilk iki yönteme göre daha iyi olur.

4- Öğrenmeye dayanarak: Bir çok FLC insanın karar verme davranışını taklit etmek için oluşturulmuştur. Fakat birkaçı insan öğrenmesi üzerinde yoğunlaştı. Şöyle ki; bulanık denetim kurallarını oluşturma yeteneği ve deneyime bağlı olarak geliştirme yeteneği gibi kendi kendini organize eden denetleçler tanımlanmıştır.

5.1.2.2.1. Denetim Kurallarının Yeterliliği

Bulanık denetim kuralları aşağıdaki yeterlilikleri sağlamalıdır.

1- Tamamlılık: Olası her bulanık durum için bir denetim işareti üretebilmelidir.

2- Nicelik: Gerekli kuralların sayısını belirlemenin genel bir yolu yoktur. Çünkü kuralların sayısını denetlecin performansı, denetim stratejisi, olası giriş kombinasyonlarının sayısı, bulanık küme tanımları gibi pek çok etken belirler.

3- Tutarlılık: Çelişik kurallar beklenmeyen veya istenmeyen sonuçlara neden olabilir. Tutarsızlık, koşul kısımları hemen, hemen aynı, sonuç kısımları farklı birden fazla kuralın varlığı durumunda ortaya çıkabilir.

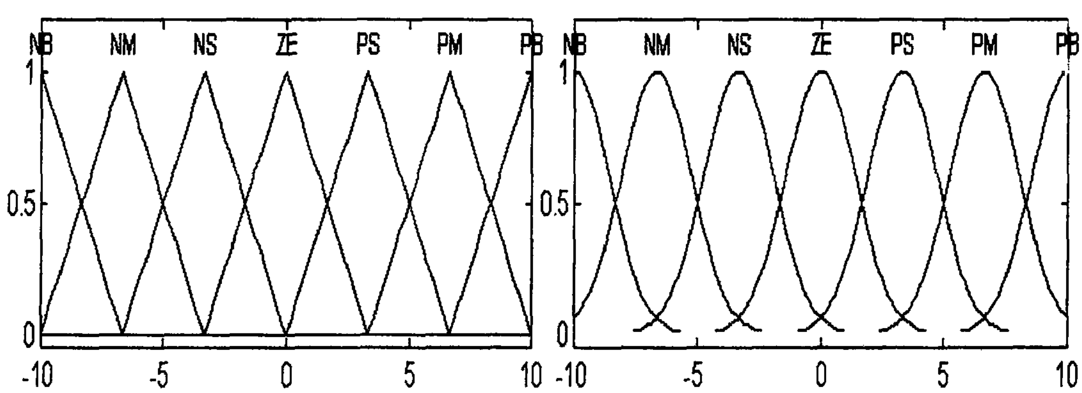
4- Etkileşim: Denetim kuralları (i=1,2,...n) " Eğer x ; A_i ise y ; C_i dir " şeklinde ise x;A_k iken sonucun C_k olması beklenir. Fakat denetim kurallarının etkileşmesinden ötürü, ortaya çıkan denetim aksiyonu, C_k'nın bir alt veya üst kümesi olabilir. Bu durumun oluşup oluşmaması kuralları değerlendiren metoda bağlıdır.

5.1.3. Karar Verme Ünitesi (Çıkartım Merkezi)

Bir FLC'de kullanılan çıkartım mekanizması genellikle tipik bir uzman sisteminde kullanılan daha da basittir. Çünkü bir kuralın sonuçsalı diğer bir öncele uygulanmaz. Başka bir değişle FLC'deki denetim sistemleri, bir derece ilerdeki veriye bağlı olduğundan çıkartım mekanizmasını dizilmiş olarak kullanır.

5.1.3.1. Çıkartım Yöntemi 1

Bulanık değişkenlerin sürekli ve kesikli olmak üzere iki tipi vardır (Şekil 5.4., Tablo 5.1.). Bulanık denetimde, negatif ya da pozitif değerler alabilen giriş ve çıkış değerler uzayı [-1,1] yaygın olarak bu değerlerde standardize edilmiştir.



Şekil 5.4. Sürekli Bulanık Değişkenler (gauss ve üçgen)

Tablo 5.1. ayrık (kesikli) bulanık değişkenler

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
PB	0	0	0	0	0	0	0	0	0	0	3	7	10
PM	0	0	0	0	0	0	0	0	3	7	10	7	3
PS	0	0	0	0	0	0	3	7	10	7	3	0	0
ZE	0	0	0	0	3	7	10	7	3	0	0	0	0
NS	0	0	3	7	10	7	3	0	0	0	0	0	0
NM	3	7	10	7	3	0	0	0	0	0	0	0	0
NB	10	7	3	0	0	0	0	0	0	0	0	0	0

Tablo 5.1., kesikli bulanık değişkenlere örnektir. Derece 0'dan 10'a kadar tamsayılarla ifade edilebilir. Giriş ve çıkış değişkenlerinin bölgesi -6'dan 6'ya kadar olan tam sayı bölgesinde ayrıktır.

Bulanık sonuç yönteminde yaygın olarak yedi bulanık değişken kullanılır. Parametrelerin ayarlanması gerekmez. Standart değişkenler kullanılır. Denetim kuralları şu şekilde olsun;

Eğer $x;A_i$ ve $y;B_i$ ise o halde $z;C_i$ 'dir $(i=1,2,...n)$ (5.4)

n, denetim kurallarının sayısı ; i, denetim kuralının numarası ; A,B,C bulanık kümeler ; x,y giriş değişkenleri ve z denetim parametrisidir, sistem iki girişli ve tek çıkışlıdır. $x=x_0$ ve $y=y_0$ olsun. Bu değerlerin, tüm kuralların öncelleriyle uyuşumu hesaplanmalıdır. Bu uyuşum ilgili bulanık kümenin elemanının üyelik derecesi cinsinden ifade edilir.

Bir kuralın uyuşumu (5.5) 'de gösterildiği gibi hesaplanır:

$$W_i=A_i(x_0)*B_i(y_0) \qquad (i=1,2,...n) \qquad (5.5)$$

n, kural tablosundaki kuralların sayısı ; i, uyuşumu (çıkartımı) hesaplanan kuralın numarası ve *, çarpmadır. i. kuralın çıkartım sonucu:

$$z \;\; W_iC_i\text{'dir ama } W_iC_i(z)=W_i\times C_i(z) \qquad (5.6)$$

olur. Bazen çarpma işlemi yerine "min" operatörü kullanılır. Minimum işleminde aşağıdaki durum vardır.

$$W_iC_i(z)=W_i\wedge C_i(z)$$

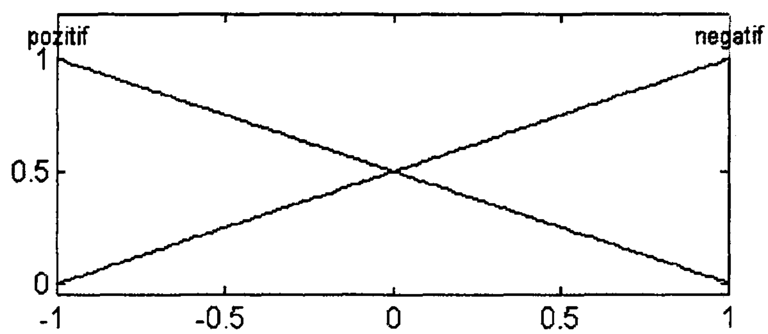
sonra $B=W_1C_1\cup....\cup W_nC_n$ (5.7)

Bu çıkartım yönteminde üç aşama vardır.

1-Giriş uygunluğu ve kuralların öncellerinin hesaplanması
2-Her bir kural için çıkartım sonuçlarının bulunması
3-Tam çıkartım sonucunun bulunması

5.1.3.2 Çıkartım Yöntemi 2

Bu yöntem özellikle monoton üyelik fonksiyonlarına sahip bulanık değişkenleri için uygundur. İki tipi vardır. Bunlar Negatif ve Pozitif. Fakat eğimde değişimler vardır.



Şekil-5.5. Monoton Üyelik Fonksiyonları

Kural tabanı şu iki kuraldan oluşsun.

Eğer x N;yP ise z N'dir

Eğer x P;yN ise z P'dir.

x,y girişleri sırasıyla x_0, y_0 olsun. Öncellerin uyuşumu ilk yöntemdeki gibi hesaplanır ve W_1, W_2 ile adlandırılır. z_1 ve z_2 bulanık olmayan çıkartım sonuçları her bir kural için şöyle hesaplanır.

$$W_1=N(z_1)\ ,\ W_2=P(z_2) \qquad (5.8)$$

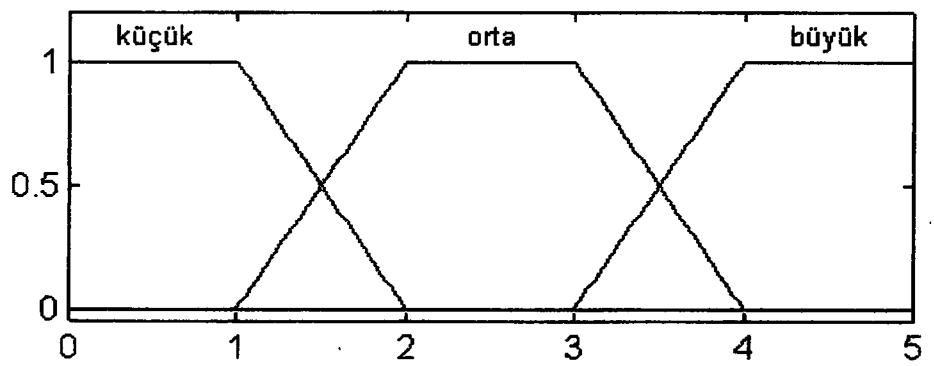
Tüm çıkartım sonucu

$$z=(w_1z_1+w_2z_2)\ /\ (w_1+w_2) \qquad (5.9)$$

5.1.3.3. Çıkartım Yöntemi 3

Bu yöntemde kullanılan önceller, bulanık önermelerle oluşturulur. Sonuçlar giriş ve çıkışların standart ilişki denklemleridir.

Öncellerde kullanılan bulanık ifadeler şekilde 5.6'da görülmektedir. Her biri düz çizgilerden oluşan trapezoidal üyelik fonksiyonlarına sahiptirler.



Şekil 5.6. Trapezoidal Üyelik Fonksiyonları

Denetim kuralları:

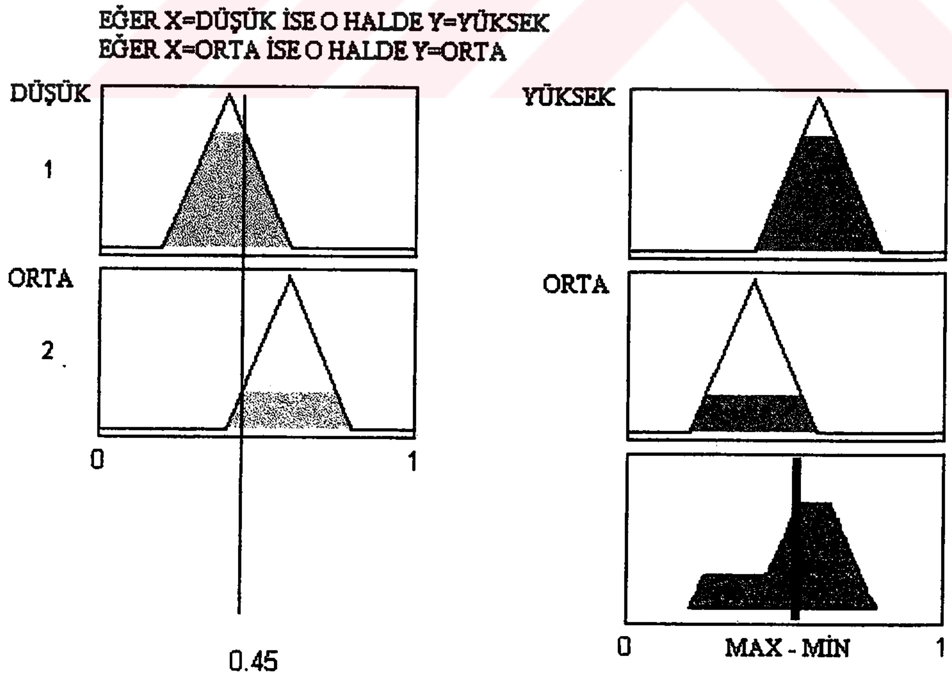
Eğer xA_1 , yB_1 ise $z=f(x,y)$'dir

Eğer xA_2 , yB_2 ise $z=g(x.y)$'dir.

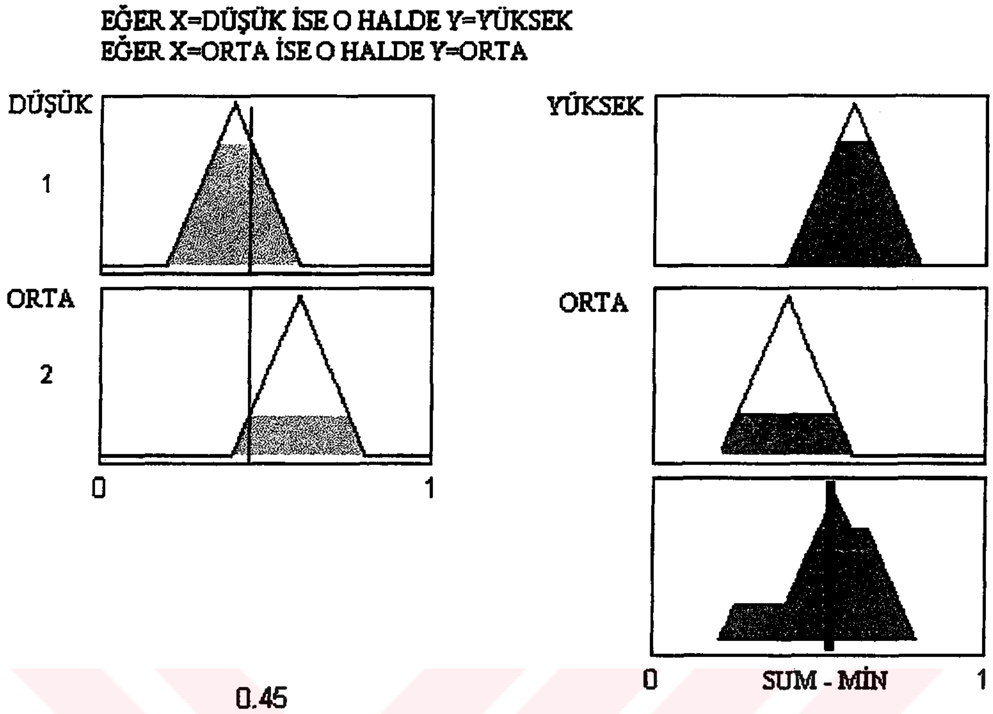
olsun. $x=x_0$, $y=y_0$ için öncellerin uyuşumu W_1 ve W_2 ise çıkartım sonuçları verilen kuralların sonuç kısmındaki eşitliklerden hesaplanır. Bunlar genelde lineer denklemlerdir. Tüm çıkartım sonucu:

$$z=[w_1f(x_0,y_0)+w_2g(x_0,y_0)] / (w_1+w_2) \text{ olur.} \qquad (5.10)$$

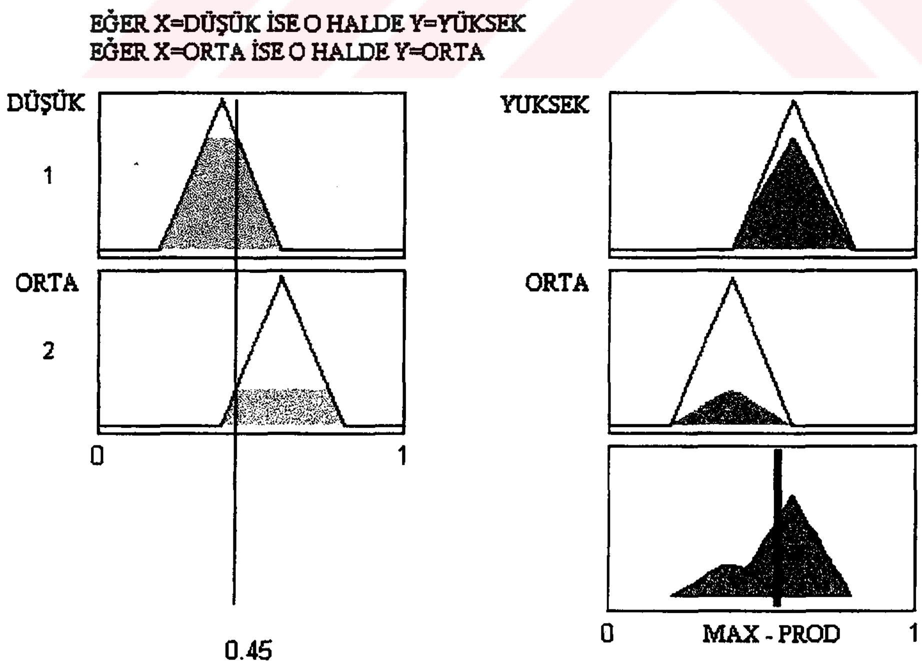
Bulanık denetleç, yukarıdaki çıkarsama yöntemlerinden birini kullanarak denetim aksiyonunu çıkartır. Bunu, sisteme uygulanacak olan gerçek denetim işaretine dönüştürmek durulaştırma arabiriminin görevidir.



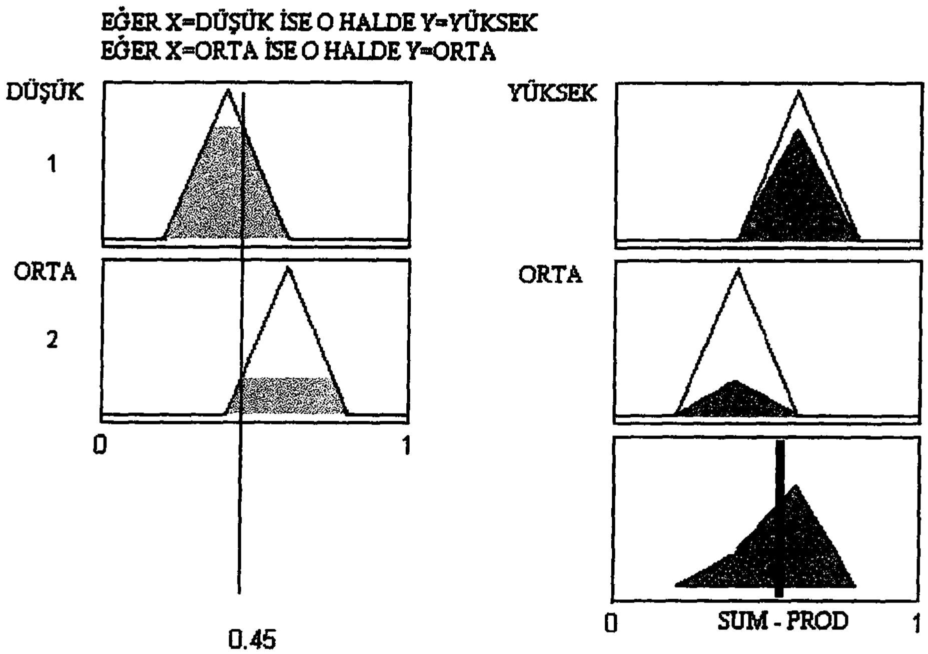
Şekil 5.7. Max - Min Çıkartım Yöntemi



Şekil 5.8. Sum - Min Çıkartım Yöntemi



Şekil 5.9. Max - Prod Çıkartım Yöntemi



Şekil 5.10 Sum-Prod Çıkartım Yöntemi

5.1.4. Durulaştırma Arabirimi

Çıkartım olayının sonucu $\mu_{yTop}(y)$ üyelik fonksiyonlu toplam-bulanık kümedir, daha sonra durulandırmanın görevi, bu bulanık kümeden en uygun " anlamlı " keskin çıkış değeri üretmektir.

Hangi durulaştırma algoritmasının seçileceği konusunda sistematik bir prosedür bulunmamakla birlikte, sıkça kullanılan durulaştırma yöntemleri aşağıda tanıtılacaktır.

1- Maksimum Değeri Alma Yöntemi:
Bu algoritmada bulanık küme içinde en büyük üyelik değerine sahip olan eleman ' kesin değer ' olarak kabul edilir. Bu yöntemin zayıf yanı, birden fazla maksimum değerin olması durumunda karşılaşılan karar verme güçlüğüdür. Ayrıca yöntem, konveks olmayan bulanık kümelerde yanlış sonuçlara yol açmaktadır.

2- Maksimumların Ortalamasını Alma Yöntemi:

Yukarıda sözü edilen, birden fazla maksimum değerin olması durumunda çözüm üreten bir yöntemdir. Bu durumda, maksimum değerlerin ortalaması ' kesin değer ' olarak alınır.

3- Ağırlık Merkezi Yöntemi:

Bu yöntemde, bulanık kümenin ağırlık merkezine karşı düşen değer 'kesin değer' olarak alınır:

$$Z=(\sum_{i=1}^{n} \mu_i(x_i).x_i)/(\sum_{i=1}^{n} \mu_i(x_i)) \qquad (5.11)$$

Burada n, kuantalama seviyesi; x_i, bulanık kümece sağlanan bir değer ; $\mu_i(x_i)$, x_i değerinin üyelik derecesi ve z, 'kesin değer' dir.

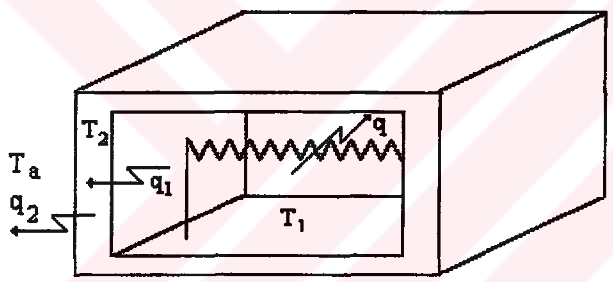
Ağırlık merkezi yöntemi, yukarıda tanıtılan diğer iki yönteme göre daha iyi ve daha güvenilir sonuçlar vermekle birlikte, konveks olmayan bulanık kümelerde karşılaşılan karar verme sorununun üstesinden gelememektedir.

BÖLÜM 6. MATEMATİKSEL MODELLER

6. 1 Sürecin Matematiksel Modeli

Sıcaklığı denetelenmek istenen düzenek şekil 6.1.'de gösterilmiştir. Rezistans sisteme "q" ısıl akısını sağlar. T_1 sistemin, T_2 tankın ve T_a ise dış ortam sıcaklığıdır.

Düzenek camdan dizayn edilmiştir. Model çıkarılmadan önce kısaca sistemin geometrisi tanıtılırsa;



Şekil 6.1.Sıcaklığı Denetlenen Düzenek

Genişlik:	0.3	[m]
Boy:	0.4	[m]
Yükseklik:	0.6	[m]
Kalınlık (L):	0.005	[m]
Camın ısıl geçirgenliği (k_{cam}):	0.78	[J / m K s]
Suyun ısıl geçirgenliği (k_{su}):	0.39	[J / m K s]
Camın özgül ısısı (c_{cam}):	501.3	[J / kg. K]
Suyun özgül ısısı (c_{su}):	4178	[J / kg. K]
Camın yoğunluğu (d_{cam}):	1112	[kg / m^3]
Suyun yoğunluğu (d_{su}):	1000	[kg / m^3]

Isıl akı (q) $= -k \cdot A \cdot \dfrac{dT}{dx}$

$$= k \cdot A \cdot \frac{T_H - T_C}{L}$$

$$= \frac{T_H - T_C}{R} \text{ dir.} \tag{6.1}$$

Sistemden tanka aktarılan ısıl akı;

$$q_1 = \frac{T_1 - T_2}{R_1} \tag{6.2}$$

Tanktan dışarıya aktarılan ısıl akı;

$$q_2 = \frac{T_2 - T_a}{R_2} \tag{6.3}$$

Enerjinin korunumu yasasına göre;

$$C_1 \frac{dT_1}{dt} = q - \left(\frac{T_1 - T_2}{R_1}\right) \tag{6.4}$$

$$C_2 \frac{dT_2}{dt} = \left(\frac{T_1 - T_2}{R_1}\right) - \left(\frac{T_2 - T_a}{R_2}\right) \tag{6.5}$$

şeklindedir.

$x_1 = T_1$ ve $x_2 = T_2$ durum değişkenleri olarak düşünülürse:

$$x_1' = -\left(\frac{1}{R_1 C_1}\right) \cdot x_1 + \left(\frac{1}{R_1 C_1}\right) \cdot x_2 + \frac{1}{C_1} \cdot q_2 \tag{6.6}$$

$$x_2' = \left(\frac{1}{R_1C_2}\right)\cdot x_1 - \left(\frac{1}{R_1}+\frac{1}{R_2}\right)\frac{1}{C_2}\cdot x_2 + \frac{1}{R_2C_2}\cdot T_a \tag{6.7}$$

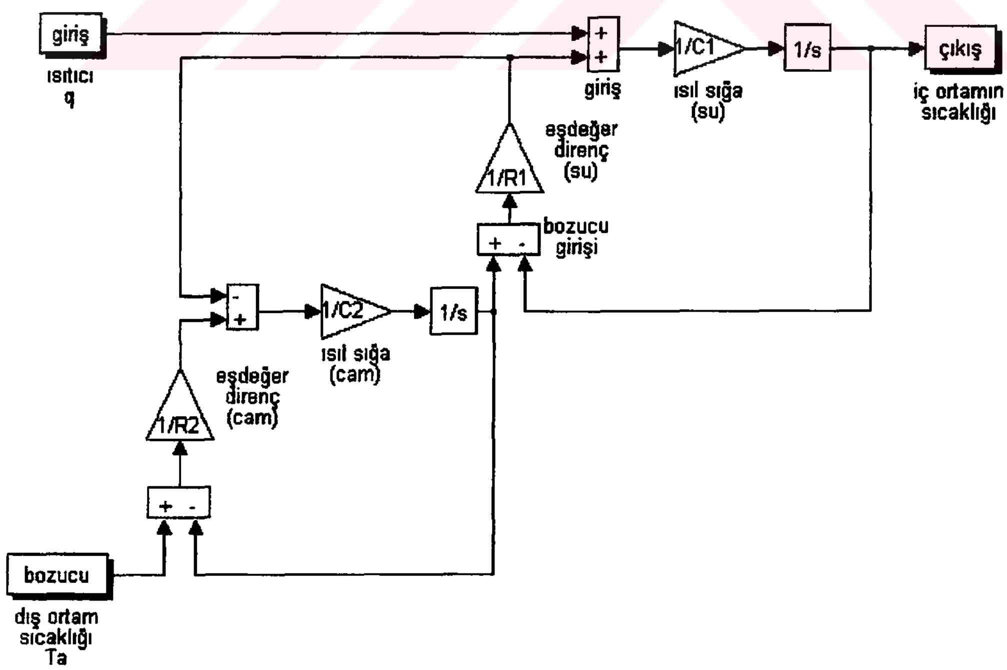
$y = T_1$ yazılabilir. (6.8)

Bu denklemler çözülürse;

$$\Rightarrow y = \left(\frac{R_2\left(sR_1C_2 + 1 + {R_1}/{R_2}\right)}{s^2R_1C_1R_2C_2 + s\left(R_1C_1 + R_2C_2 + R_2C_1\right) + 1}\right)\cdot q + \left(\frac{1}{s^2R_1C_1R_2C_2 + s\left(R_1C_1 + R_2C_2 + R_2C_1\right) + 1}\right)\cdot T_a \tag{6.9}$$

elde edilir.

Bu denklemler ışığında sistemin blok modeli şekil 6.2. 'de verilmiştir.



Şekil 6.2. Sıcaklığı Denetlenen Sistemin Blok Modeli

Blokta kullanılan sabitlerin sayısal değerlerini hesaplarsak;

A_1 (İç yüzeyin alanı) = [(0.3*0.4) + (0.4*0.6) + (0.6*0.3)] *2
= 1.08 [m^2]

A_2 (Dış yüzeyin alanı) =[(0.305*0.405) + (0.405*0.605) + (0.605*0.305)] *2
=1.106 [m^2]

V_1 (Suyun hacmi) = 0.4 * 0.6 * 0.3
=0.072 [m^3]

V_2 (Camın hacmi) = [0.405 * 0.605 * 0.305]-0.072
=2.732 e^{-3} [m^3]

$R_{1,2}$ eşdeğer ısıl dirençlerinin hesaplanması ;

$R_1 = R_{su}$ = Isıtıcı, yüzey uzaklığı / (Suyun ısıl geçirgenliği * Alan)
≅ 0.2 /(0.39 * 1.08)
=0.474 [K. s / J]

$R_2 = R_{cam}$ = Cam Kalınlığı / (Camın ısıl geçirgenliği * Alan)
= 0.005 /(0.78 * 1.106)
=5.795 e-3 [K. s / J]

Suyun kütlesi = (hacim) * yoğunluk
= (0.4 * 0.3 * 0.6) * 1000
= 48 [kg]

Camın kütlesi =2.732 e^{-3} * 1112
=3.037 [kg]

ısıl kapasitelerin hesaplanması;

ısıl kapasite ($C = \Delta Q / \Delta T$) = özgül ısı * kütle

$C_1 = C_{su}$ = 4178 * 48

= 200544 [J / K]

$C_2 = C_{cam}$ = 501.3 * 3.037

= 1522.448 [J / K]

Bulunan sayısal değerler sistem modelinde yerine yazılırsa;

$$y = \left(\frac{5.795e^{-3}\left(s \cdot 0.474 \cdot 1522.448 + 1 + \frac{0.474}{5.795e^{-3}} \right)}{s^2 \cdot 0.474 \cdot 200544 \cdot 5.795e^{-3} \cdot 1522.448 + s \cdot \left(0.474 \cdot 200544 + 5.795e^{-3} \cdot 1522.448 + 5.795e^{-3} \cdot 200544 \right) + 1} \right) \cdot q$$
$$+ \left(\frac{1}{s^2 \cdot 0.474 \cdot 200544 \cdot 5.795e^{-3} \cdot 1522.448 + s \cdot \left(0.474 \cdot 200544 + 5.795e^{-3} \cdot 1522.448 + 5.795e^{-3} \cdot 200544 \right) + 1} \right) \cdot T_a \quad (6.10)$$

$$y = \left(\frac{4.181 \cdot s + 0.479}{s^2 \cdot 838656.124 + s \cdot 96228.831 + 1} \right) \cdot q + \left(\frac{1}{s^2 \cdot 838656.124 + s \cdot 96228.831 + 1} \right) \cdot T_a \quad (6.11)$$

6.2. Denetim Elemanı

Sistem için gereken q girişi "Tam dalga-çift yollu doğrultucu" devre tarafından sürülen bir resistans ile sağlanmaktadır. Böyle bir devrenin girişine E_m Sin(wt) gibi bir işaret uygulandığında α tetikleme açısı olmak üzere

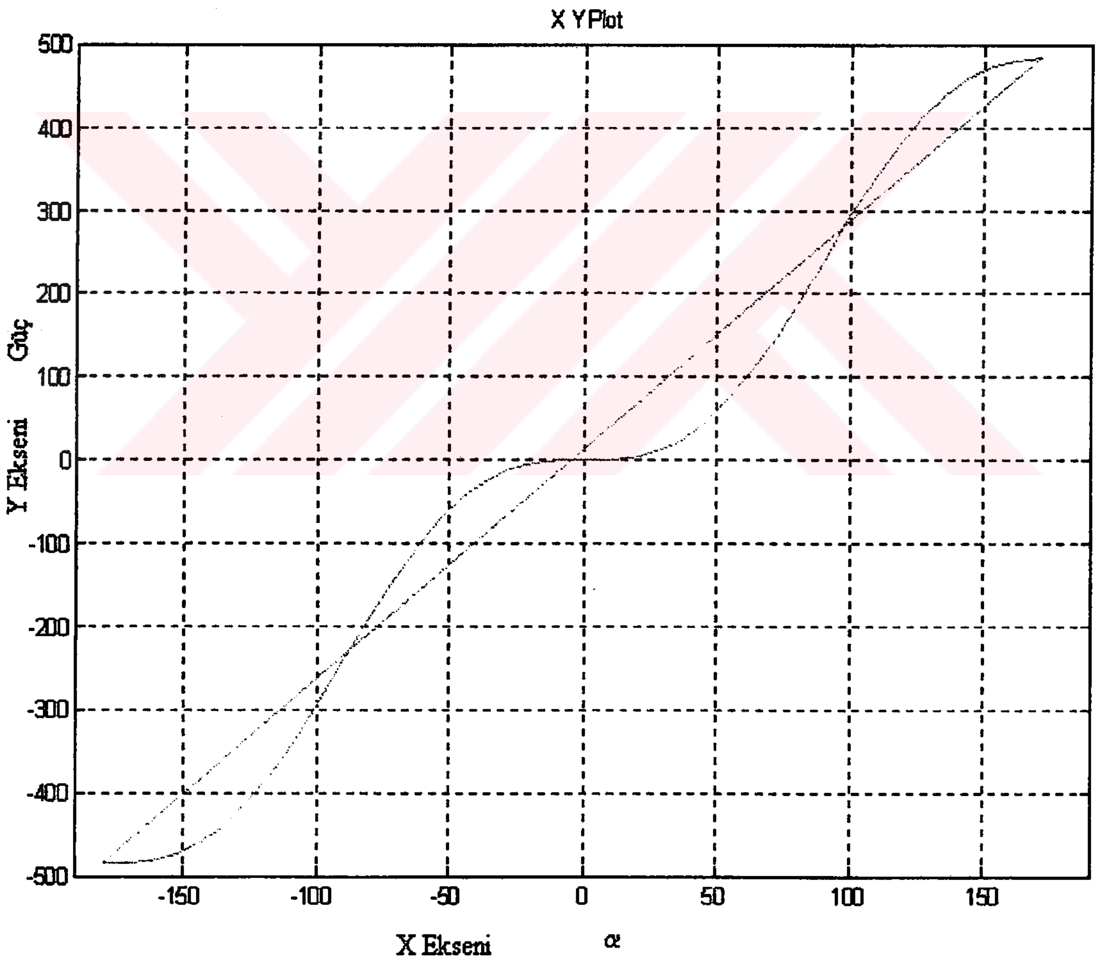
$$E_{eff} = \left[\left(\frac{1}{T} \right) \int_0^T \left(E_m \cdot Sin(wt) \right)^2 dwt \right]^{1/2} \quad (6.12)$$

denklemine göre resistans üzerine düşen efektif gerilim;

$$E_{eff} = \left[\left(\frac{1}{\pi}\right)\int_0^{\pi}\left(E_m \cdot Sin(wt)\right)^2 dwt\right]^{1/2} \quad (6.13)$$

$$E_{eff} = E_m\left[\frac{1}{2} - \frac{\alpha}{2\pi} + \frac{1}{4\pi}Sin(2\alpha)\right]^{1/2} \quad (6.14)$$

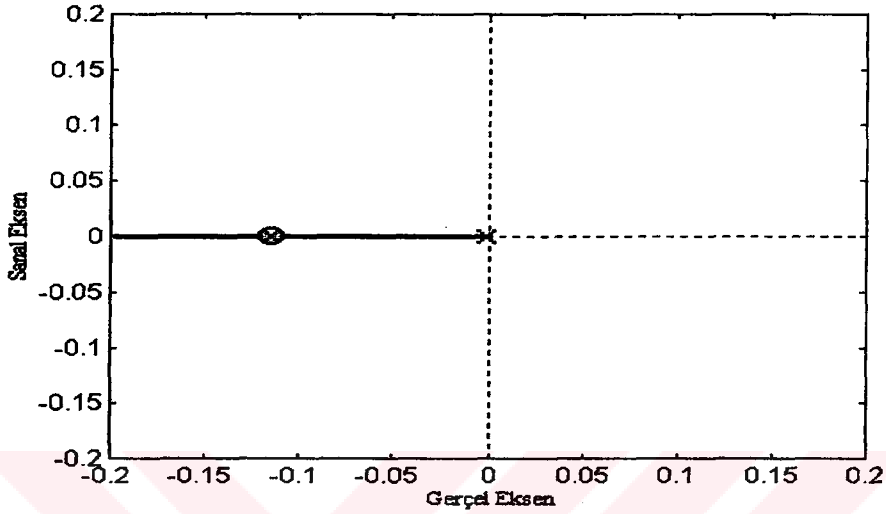
tetikleme açısı α'ya göre sisteme verilebilecek güç şekil 6.3 'de verilmiştir.



Şekil 6.3. α Tetikleme Açısına Göre Sisteme Verilebilecek Güç

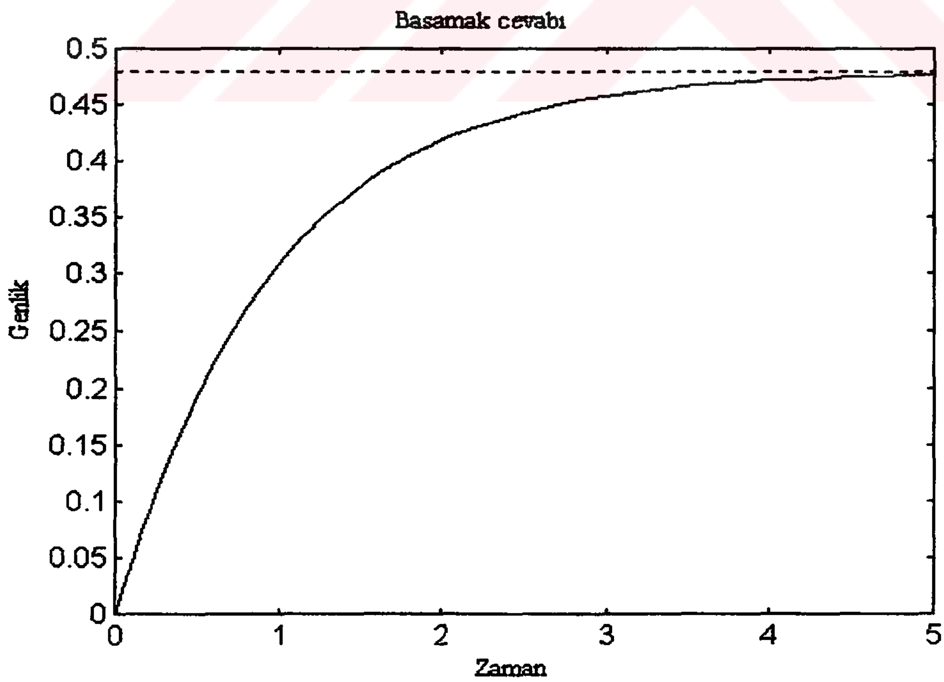
6.3. Sistem Karakteristikleri

sistemin kök-yer eğrisi şekil 6.4 'de verilmiştir.

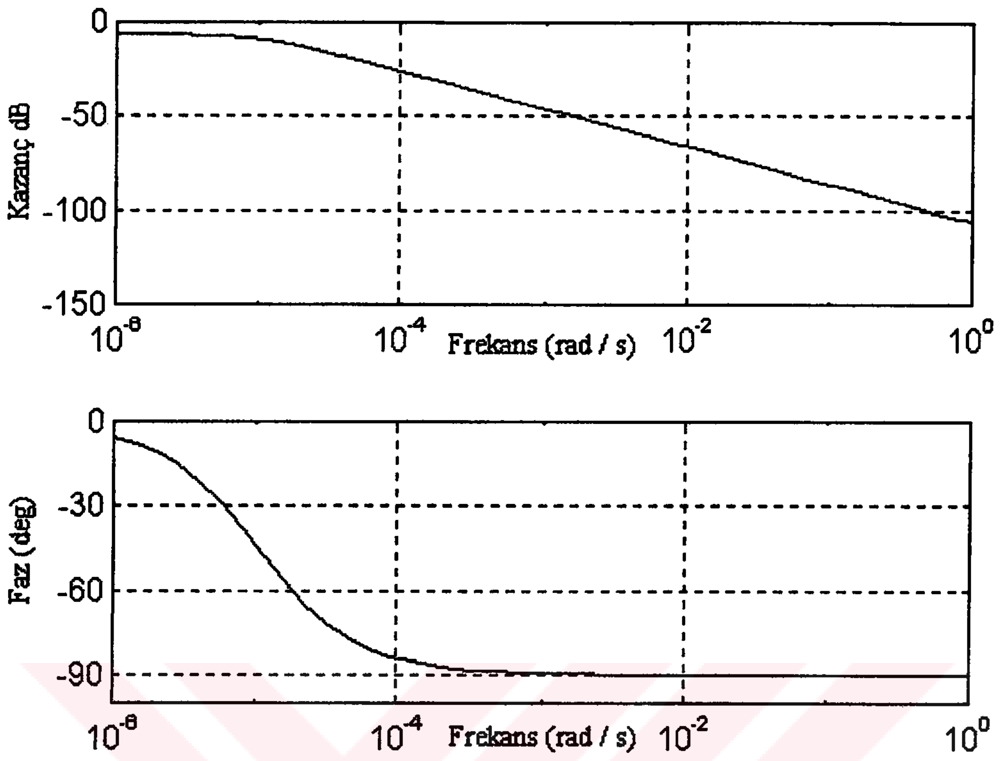


Şekil 6.4. Sıcaklığı Denetlenen Sistemin Kök Yer Eğrisi

bu sistemin basamak cevabı ise şekil 6.5 'de verilmiştir.



Şekil 6.5. Sıcaklığı Denetlenen Sistemin Basamak Cevabı



Şekil 6.6. Sıcaklığı Denetlenen Sistemin Frekans Cevabı

Şekil 6.6 'da sistemin frekans cevabı görülmektedir.

Şimdiye kadar sistemin sürekli domendeki davranışları incelendi, oysa sistemin denetimi bilgisayar yardımı ile yapılacağı için sistem ayrık zamanda incelenmeli ve örnekleme zamanı üzerinde durulmalı.

Sistemin transfer fonksiyonu denklem (6.8)'de belirtilmişti. Bu fonksiyonda bulunan T_2 sistem için bozucu bir giriştir ve denetlenemez. Ayrık zamandaki incelemelerde bozucu giriş olmadığı düşünülecektir. Dolayısı ile sistemin s düzlemindeki transfer fonksiyonu (6.13) denkleminde gösterildiği gibidir.

$$\Rightarrow H(s) = \frac{4.181 \cdot s + 0.479}{\left(1 + s \cdot 96228.831 + s^2 838656.124\right)} \tag{6.15}$$

olur.

Her şeyden önce ayrık zamandaki kısıtlama örnekleme aralığıdır. Bilgisayarda denetimi yapılacak bu sistemde 20 ms'lik örnekleme zamanı kullanmaktayız.

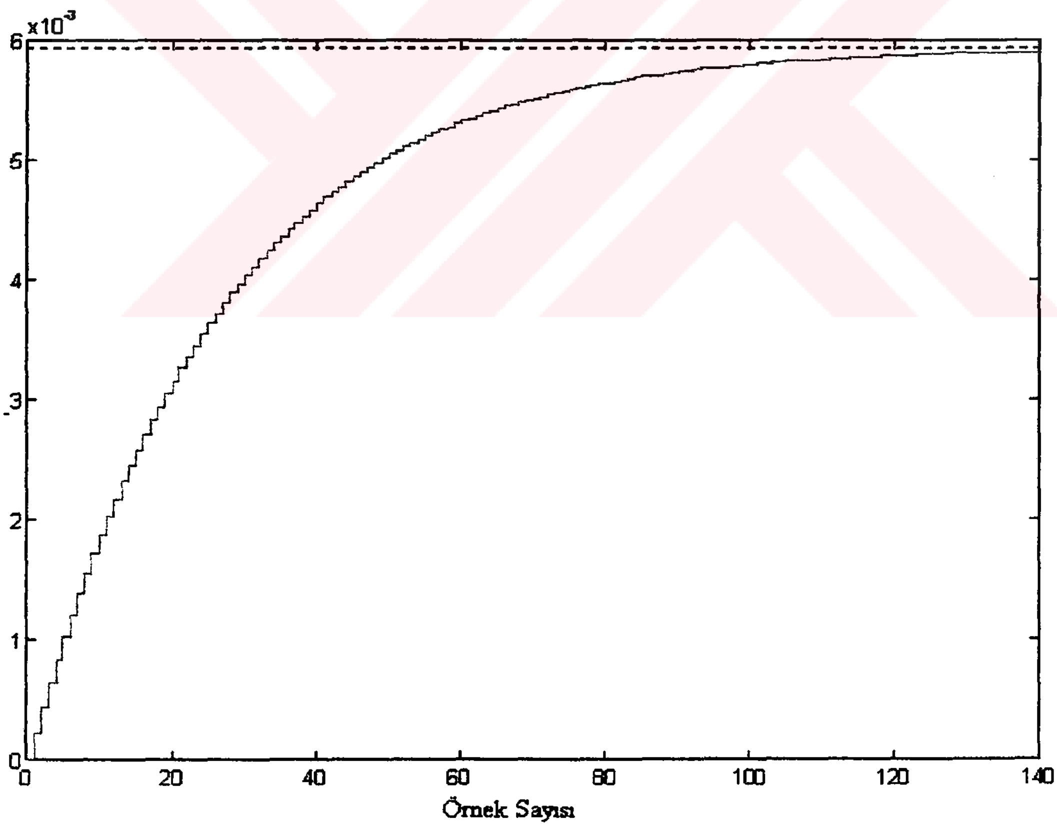
T_s =20 ms için sistemin frekans cevabı ayrık düzlemde çizilmiştir (Şekil 6.8.)

z düzleminde sistemin transfer fonksiyonu;

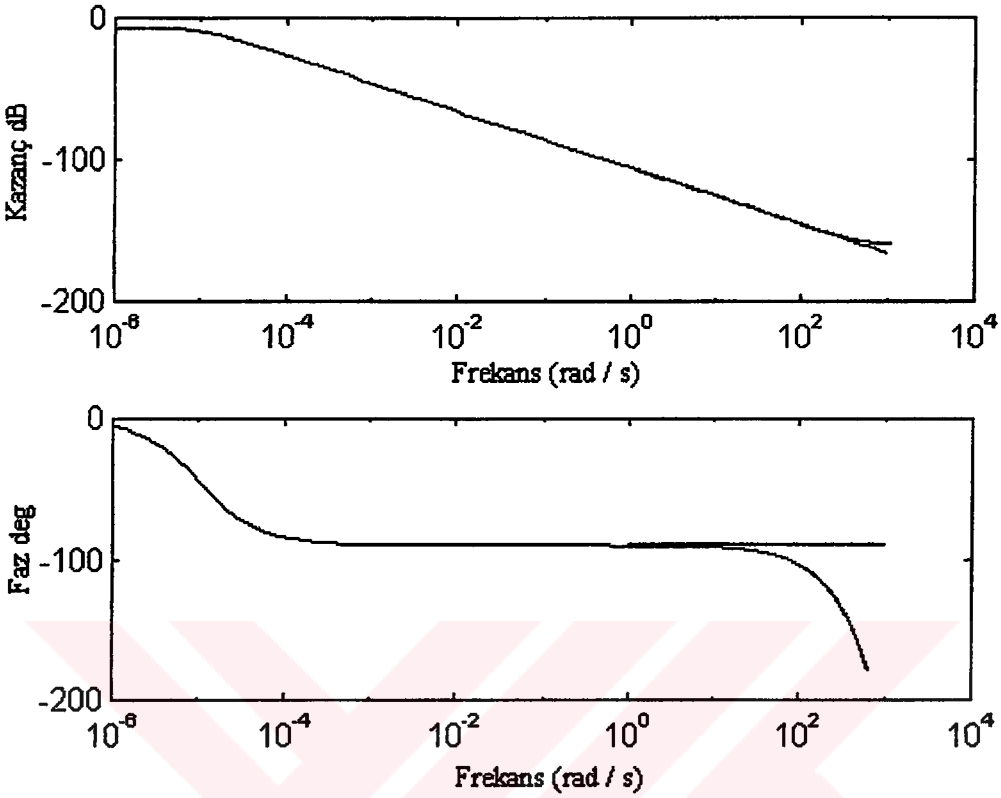
$$H(z) = \frac{9.971e^{-8} \cdot z - 9.948e^{-8}}{\left(z^2 - 1.998z + 0.9977\right)} \tag{6.16}$$

olur.

Ayrık zamanda birim basamak cevabı şekil 6.7 'de verilmiştir.

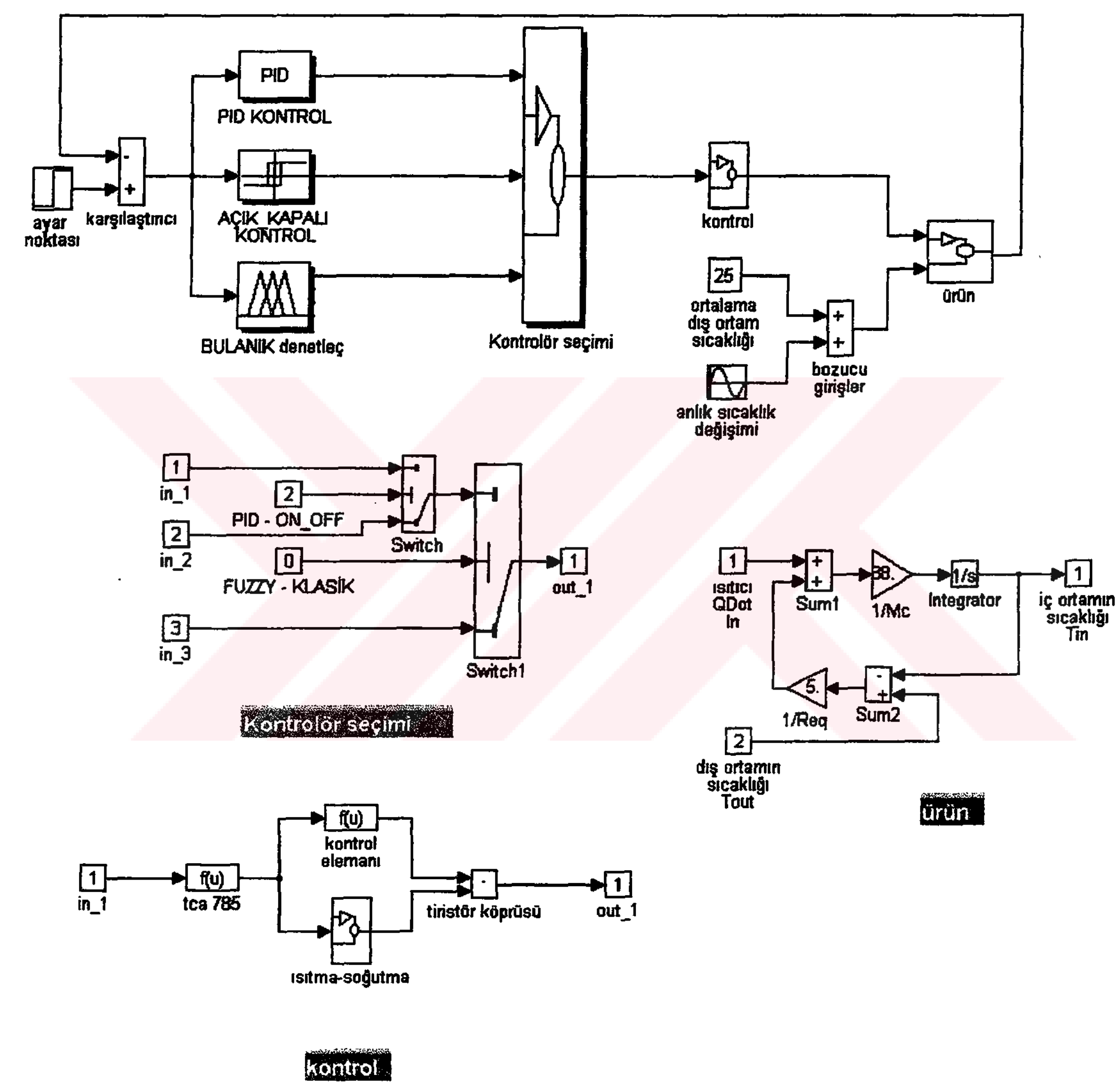


Şekil 6.7 Sıcaklığı Denetlenen Sistemin Ayrık zamanda Birim Basamak Cevabı



Şekil 6.8. Sıcaklığı Denetlenen Sistemin Ayrık zamanda Frekans Cevabı

BÖLÜM 7.SİSTEMİN FARKLI DENETLEÇ KİPLERİ İLE SİMÜLASYONU



Şekil 7.1. Simulasyonlarda Temel Alınan Sistemin Blok Gösterimi

7.1. Denetim Parametrelerinin Bulunmasında Ziegler ve Nichols Yöntemi:

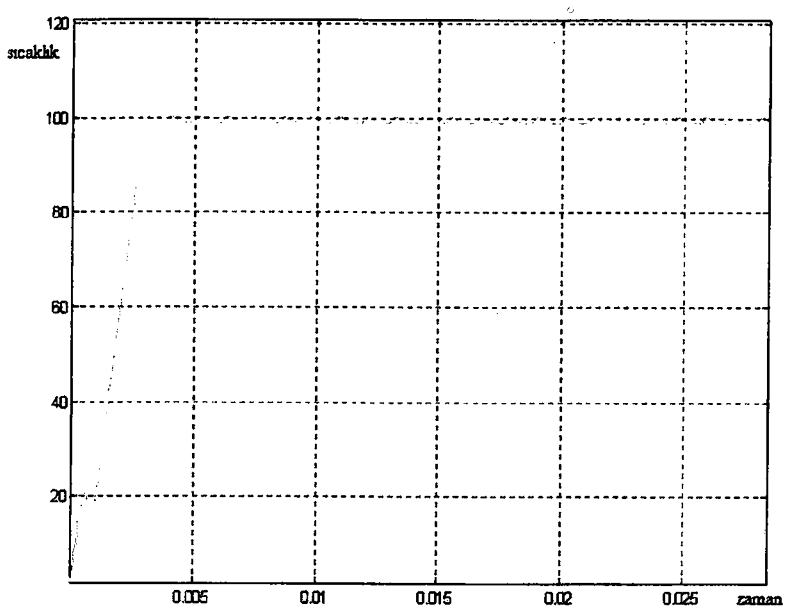
Ziegler ve Nichos (1963) deney ve analizlerden yola çıkarak uygulanması kolay ve pratikte başarılı olan denetim kuralları geliştirmişlerdir. Yaptıkları gözlemlerde fark etmişlerdir ki mutlak hatanın entegralini $I = \int_0^{\infty} |e(t)| dt$ minimum kılan denetleç kazancında ikinci aşma birincinin ¼'ü kadardır (çeyrek bozulma kriteri olarak bilinir). Bu kazanç değerinde yükselme zamanı (t_r), oturma zamanı (t_s) ve karalılık en uyumlu haldedir. Bu saptamadan yola çıkarak geliştirdikleri kurallarla iki şart sağlanarak bulunan değerler işlenip uygun entegral, türev ve oransal kazanç sabitleri hesaplanabilir. Öncelikle oransal denetim haricinde bütün denetim kipleri kaldırılarak kararlılık sınırları içerisinde k_{pu} gibi bir kazançla denetim yapılmaktadır. Oransal kazancın bu değeri en yüksek kazanç olarak adlandırılır. Fakat bu koşul yeterli değildir. Örneğin ikinci dereceden bir sistem "$0 < k_{pu} < \infty$" için karalıdır. En yüksek bu kazanç değerinde sistemin s düzleminde sanal eksen üzerinde kutuplarının olması gerekmektedir. Sistemin kalan kökleri negatif gerçel kısımlara sahip olmalıdır ki osilasyonlar bastırılabilsin.

T_u bu osilasyonların periyodu olarak alınırsa PID denetim kazançları ;

$$k_p + {k_i}/{s} + {k_d \cdot s}/{(\tau_s + 1)} \tag{7.1}$$

(7.1) formülüne göre şöyle hesaplanır.;

P denetim kuralı:	k_p=0.5 k_{pu}	k_i=0	k_d=0
PI denetim kuralı:	k_p=0.45 k_{pu}	k_i=0.45 k_{pu}/(0.83 T_u)	k_d=0
PID denetim kuralı:	k_p=0.6 k_{pu}	k_i=0.6 k_{pu}/(0.5 T_u)	k_d=0.6 k_{pu}/(0.125 T_u)



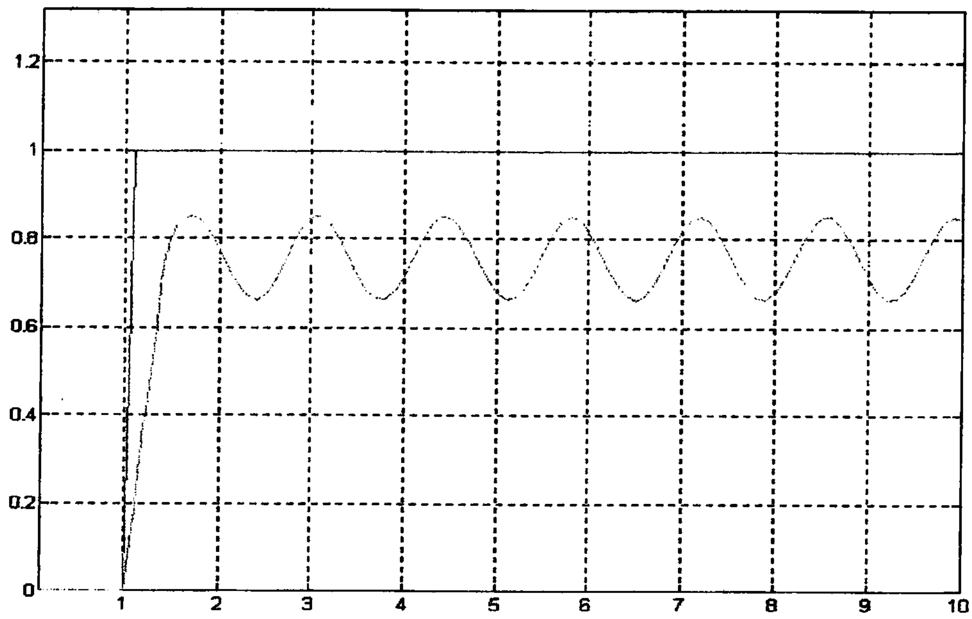
Şekil 7.2.Açık Kapalı Denetimin Basamak Girişine Cevabı

Sistem açık kapalı denetim kipinde salınımlar yapar. Açık kapalı çevrim kazancı arttırılmış bir oransal denetimden farklı değildir salınımları azaltmanın ve oturma zamanı ile yükselme zamanını sistemden istenen verime göre ayarlanmalıdır.
Genel olarak bir sistemden beklenen özellikler şunlardır:

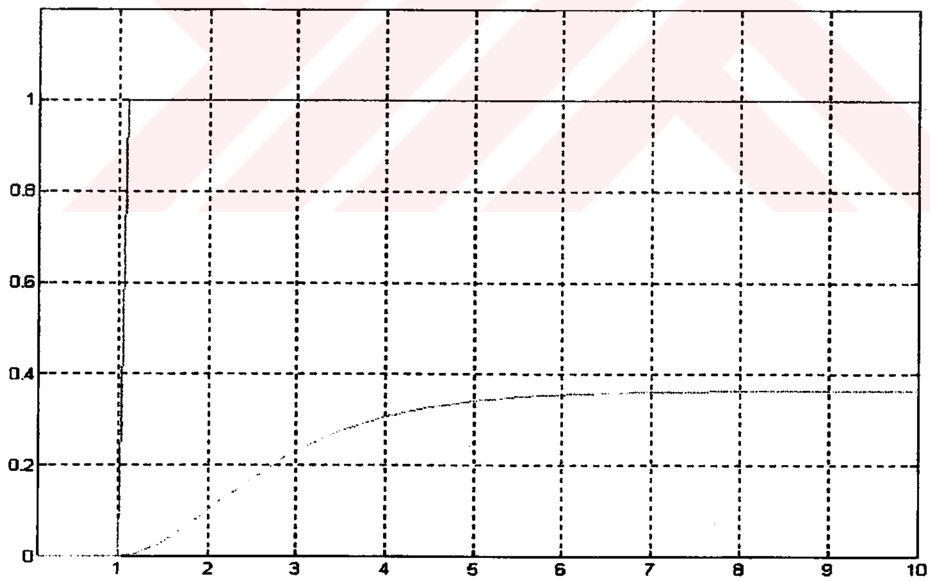
a- Bir bozucu ortaya çıktığında, denetlenen değişken olabildiğince çabuk olarak istenen değere geri dönmelidir
b- Denetim sistemi kararlı olmalı ve salınımların oluşmasına eğilim göstermemelidir. Duyarlılıktaki artış bozucu etkenlere olan tepkiyi hassaslaştıracağından bu iki madde arasında uyum sağlanmalıdır.

Şekil 7.3.'de farklı denetim kiplerinde oransal, entegral ve türevsel kazancın yetersiz değerlerinde sistemden nasıl cevap alınacağı gösterilmiştir.

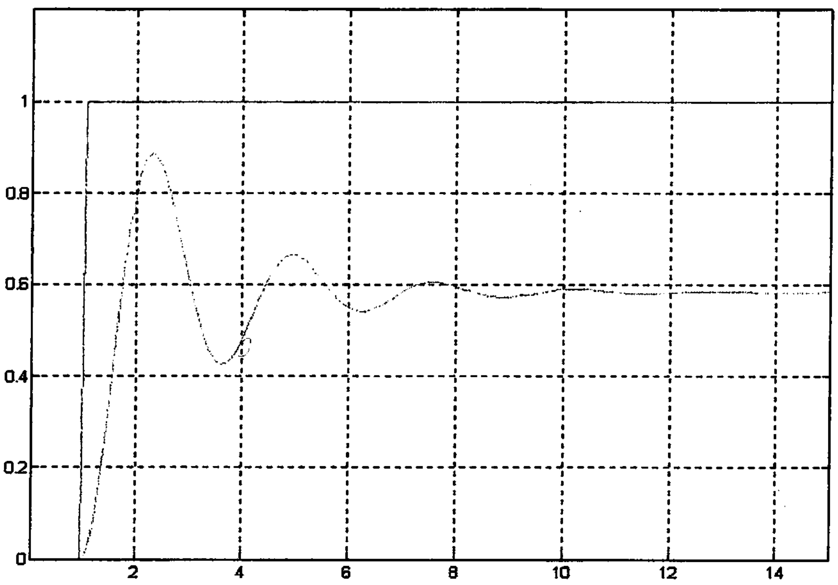
Oransal denetim: oransal denetim genelde kararlı bir sistemdir. Ancak çeşitli yük koşulları altında her zaman ölçülen değişkenin istenen değerde olmasını sağlayamaz.



Şekil 7.3.Çok Duyarlı Denetim Çevrinmeli

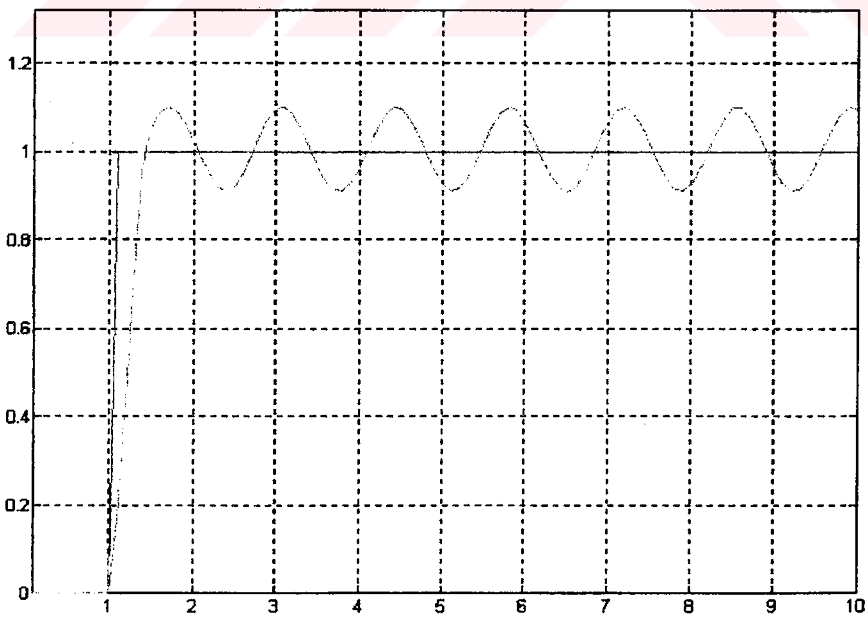


Şekil 7.4. Yetersiz Duyarlıkta

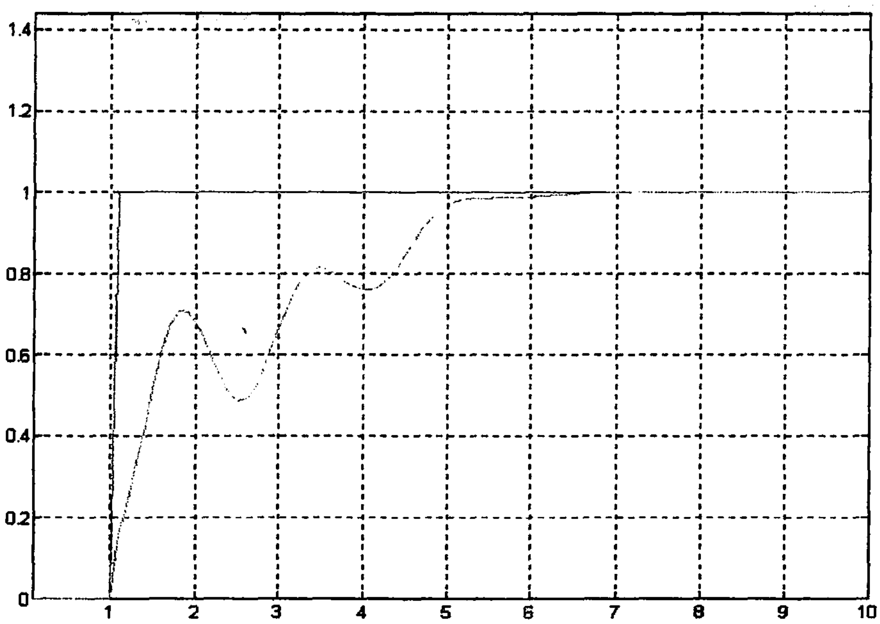


Şekil 7.5. Doğru Duyarlıkta

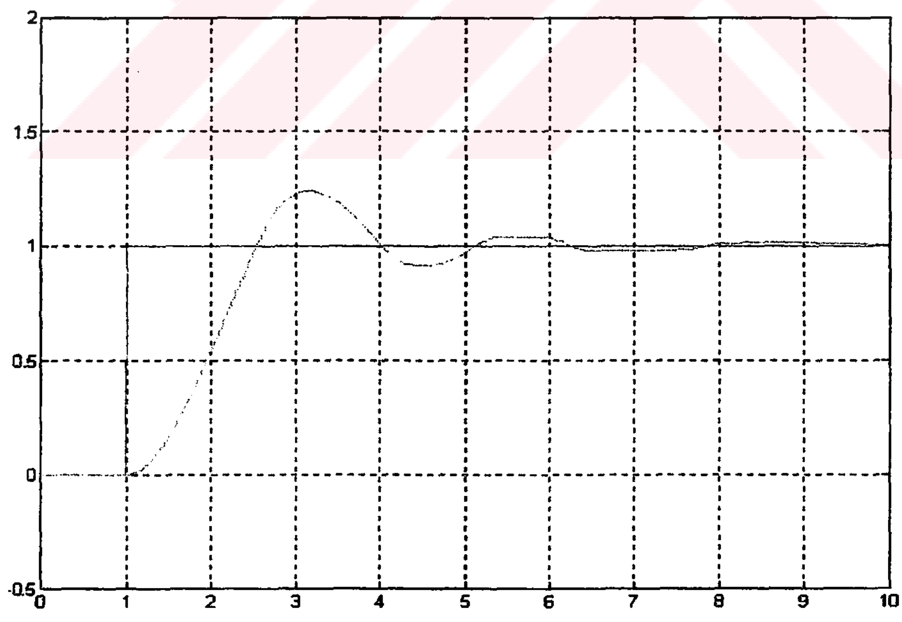
PI DENETİM: entegral denetim ölçülen değişkeni her zaman istenen değere geri döndürür, ancak sistemi daha az kararlı yapmaya yönelir ve sistemin doğasında var olan salınımların frekansını azaltır.



Şekil 7.6. Entegral (belki de oransal) Olarak Çok Duyarlı Sistem

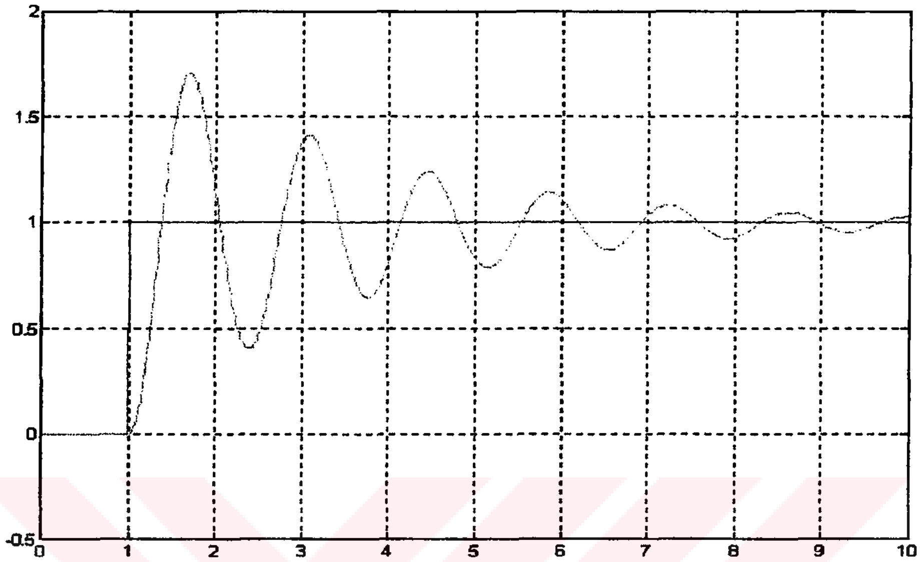


Şekil 7.7. Doğru Oransal Duyarlıkta, Entegral Duyarlığı Yetersiz Sistem

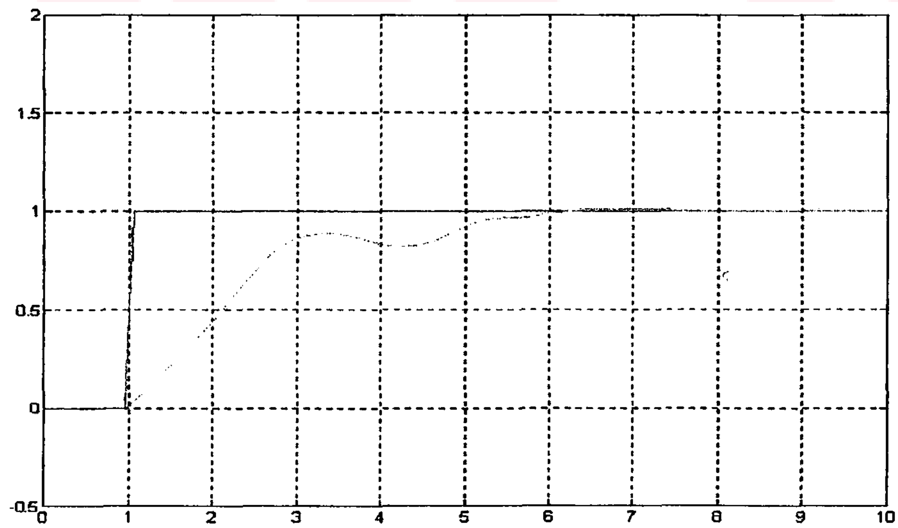


Şekil 7.8. Doğru Oransal ve Entegral Duyarlıkta Sistem

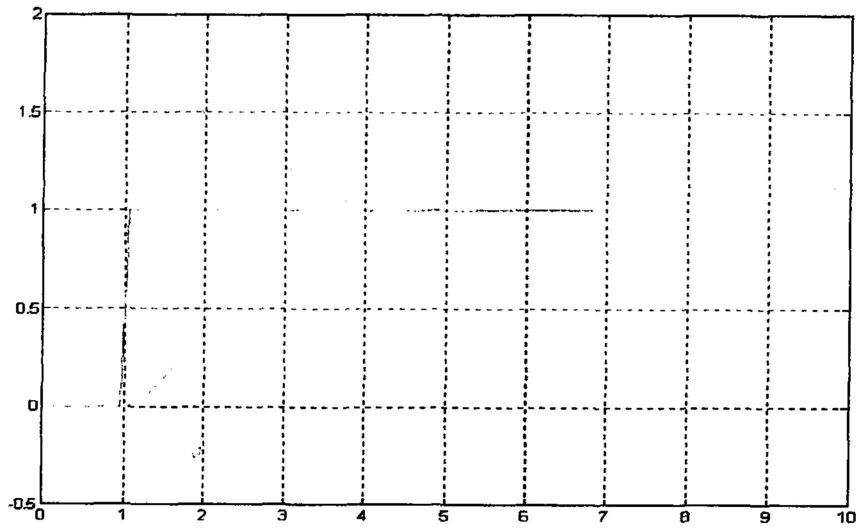
PID denetim: Türevsel denetimin ilavesi ile sistem özelliklerine bağlı olarak bazı sistem döngülerini kararlı yapmaya ve sistemdeki salınımların frekansını arttırmaya yönelir



Şekil 7.9. Doğru Oransal Duyarlıkta, Entegral Duyarlılığı Fazla Yetersiz Türevsel Duyarlıklı Sistem

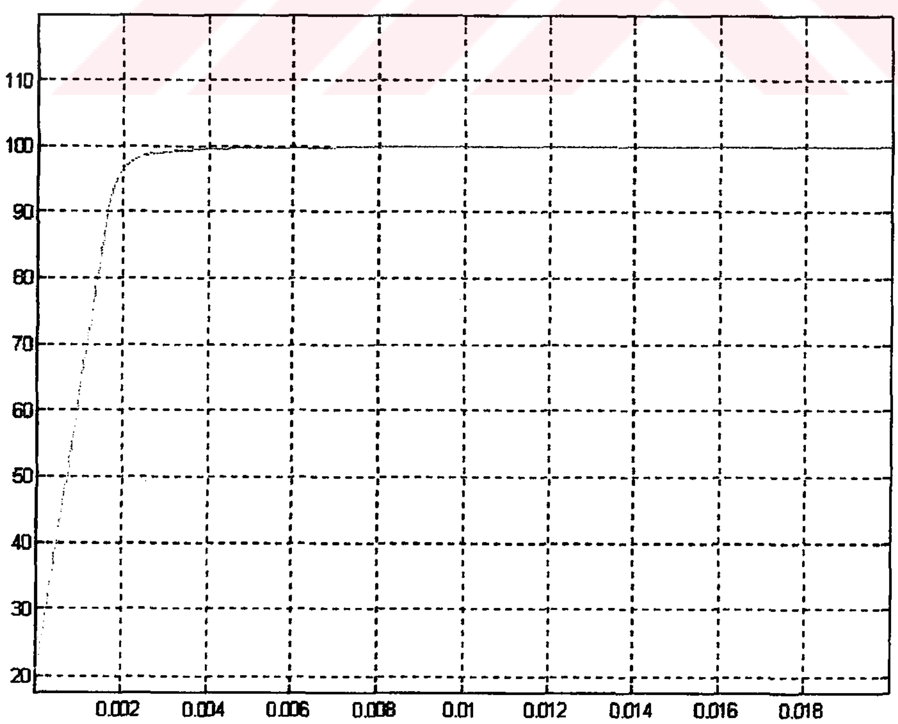


Şekil 7.10. Doğru Oransal Duyarlıklı, Yetersiz Entegral Duyarlıkla ve Fazla Türevsel Duyarlıkla Denetlenen Sistem



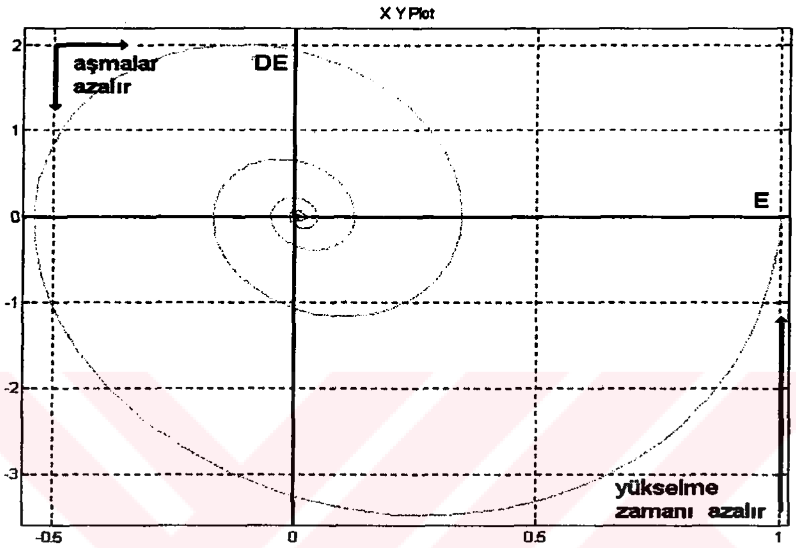
Şekil 7.11. Doğru PID Kazançları ile Kontrol Edilen Sistem

Bu özellikler dikkate alınarak yapılan simulasyonlar sonucu sistemin yavaş değişen bir süreç olmasından dolayı türevsel etkiye gerek kalmadan istenen değere ulaştığı gözlenmiştir.

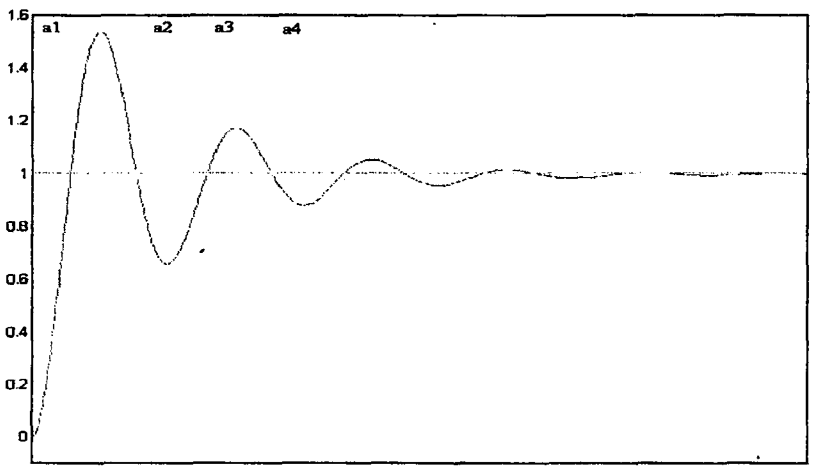


Şekil 7.12. Sistemin PI Kontrolü

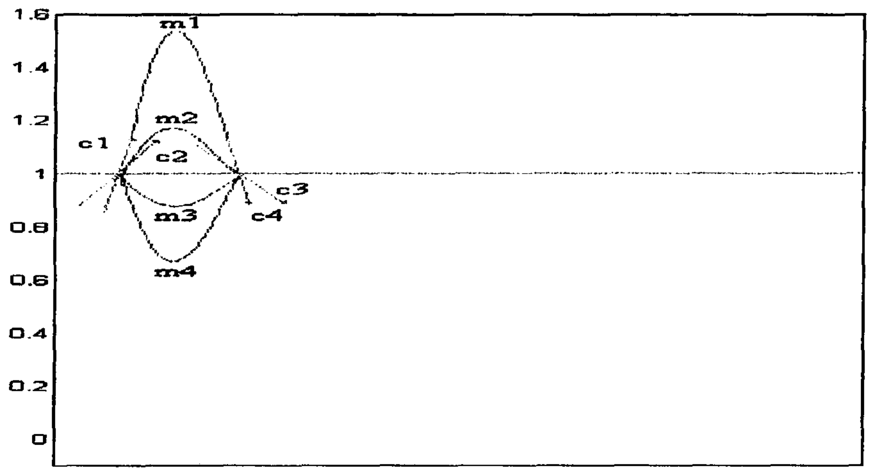
Bulanık denetim Simulasyonları sistemin klasik denetim simulasyonları sonucunda kazanılan operatör bilgisi ile Tablo 7.1.'de verilen kurallar oluşturulmuştur. Ayrıca bir sistematik olmamakla beraber sistemin e-de grafikleri kural oluşturmada yardımcı olmaktadır.



Şekil 7.13. Sistem Tepkisinin "E-DE" Girişine Cevabı



Şekil 7.14. Kural Matrisinin Oluşturulmasında Temel Alınan Aralıklar



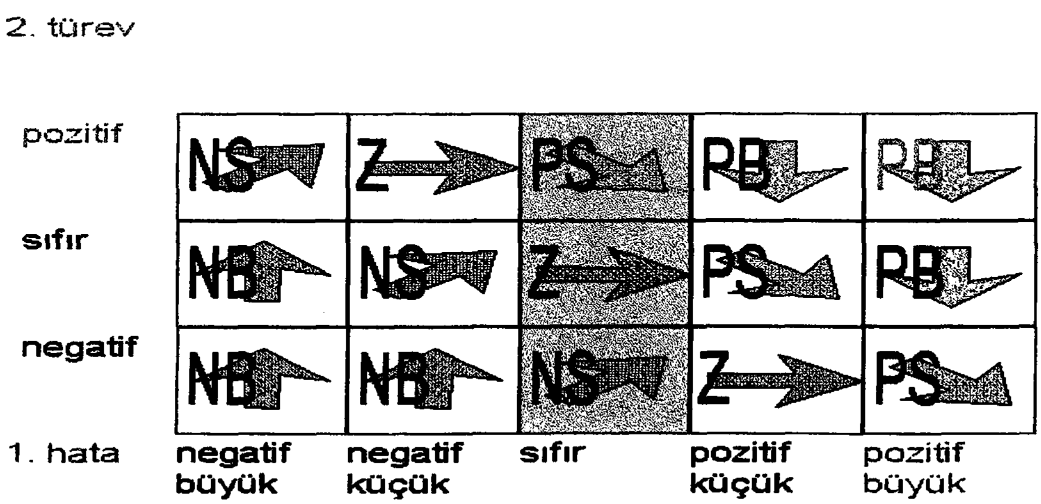
Şekil 7.15. Hatanın ve Türevin Bulanık Ölçeklendirilimesi

Şekil 7.14. ve 7.15. ışığında aşağıdaki taslak matris hazırlanmıştır.

Tablo 7.1. Kural Matrisinin Temel Bölgeleri

E \ DE	Negatif büyük	negatif küçük	sıfır	pozitif küçük	pozitif büyük
negatif	a2	a2	c1	a1	a1
sıfır	m4	m3	sıfır	m2	m1
pozitif	a3	a3	c4	a4	a4

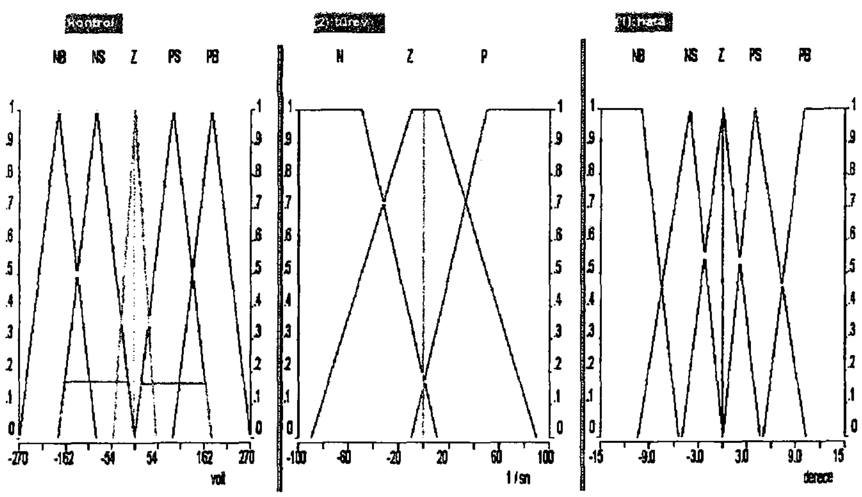
Tablo 7.2. Kural Matrisi



Bu kurallar aşağıdaki şekilde de ifade edilebilir

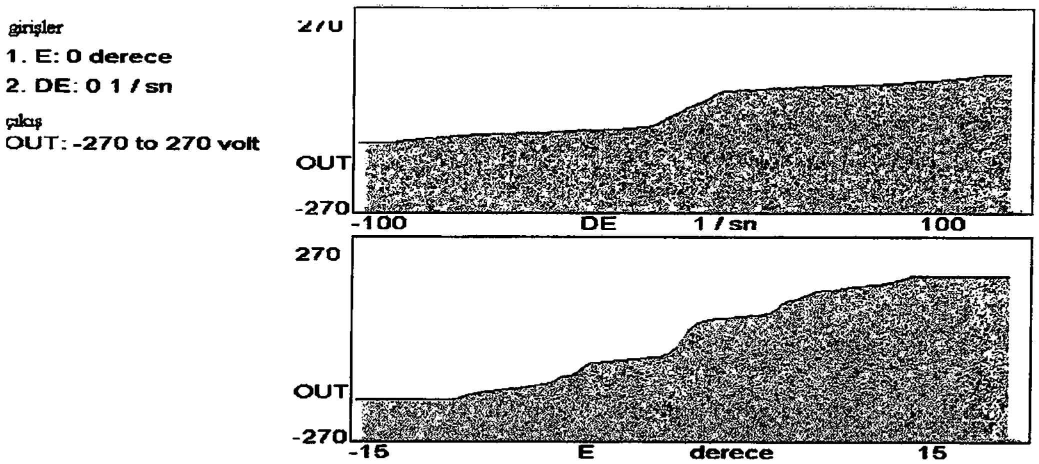
1. (e==nb) & (de==n) => (out=ns) (1)
2. (e==nb) & (de==z) => (out=nb) (1)
3. (e==nb) & (de==p) => (out=nb) (1)
4. (e==ns) & (de==n) => (out=ze) (1)
5. (e==ns) & (de==z) => (out=ns) (1)
6. (e==ns) & (de==p) => (out=nb) (1)
7. (e==z) & (de==n) => (out=ps) (1)
8. (e==z) & (de==z) => (out=ze) (1)
9. (e==z) & (de==p) => (out=ns) (1)
10. (e==ps) & (de==n) => (out=pb) (1)
11. (e==ps) & (de==z) => (out=ps) (1)
12. (e==ps) & (de==p) => (out=ze) (1)
13. (e==pb) & (de==n) => (out=pb) (1)
14. (e==pb) & (de==z) => (out=pb) (1)
15. (e==pb) & (de==p) => (out=ps) (1)

üyelik fonksiyonları şekil 7.16'da verilen geometriye sahiptir

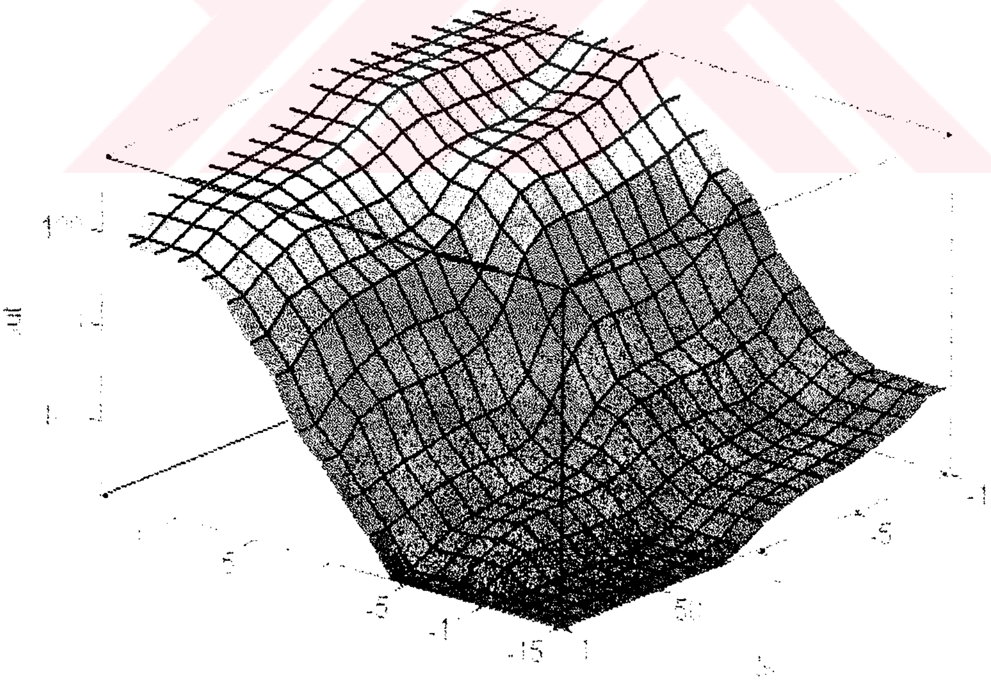


Şekil 7.16. Üyelik Fonksiyonları

Tüm bu kurallar ve üyelik fonksiyonları altında sistem şekil 7.17'de gösterilen çıkış karekteristiklerine sahiptir.

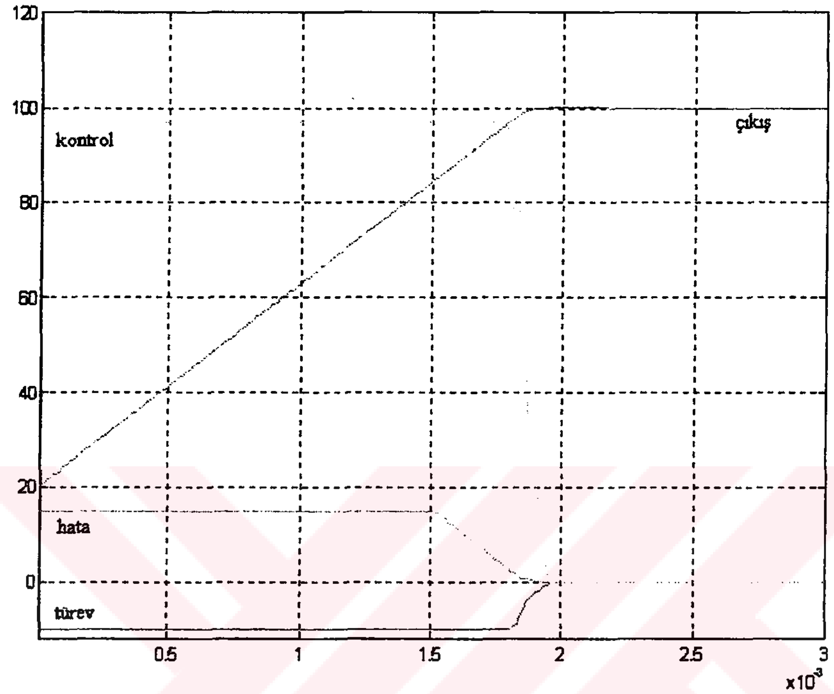


Şekil 7.17. Çıkışın Profil Görüntüsü



Şekil 7.18 Çıkış Karekteristiği

Bu yapıda tasarlanmış bulanık denetleç ile kumanda edilen sistemin simulasyon sonucu şekil 7.19'da verilmiştir



Şekil 7.19. Bulanık Kontrol Simulasyon Sonuçları

SONUÇLAR ve ÖNERİLER

Bu çalışmada bulanık denetleç ile bir sürecin sıcaklık denetimi amaçlanmıştır. Sıcaklığın denetlenmesinin sebebi operatör bilgisine kolayca sahip olabilmektir. Ayrıca böyle bir sistem tüm klasik denetim mekanizmaları içinde en sık işlenen konudur. Sistem modelinin oldukça kolay çıkarılabilmesi, klasik denetleç tasarımını basitleştirmektedir. Modeli kolaylıkla çıkarılabilen bir sistemin bulanık mantıksal denetleçlerle denetlenmesi ilk bakışta bir tutarsızlık gibi görünebilir. Fakat bu çalışmada amaç, bir uzman bilgisi ile hazırlanan bulanık denetlecin, klasik süreç denetleçleri ile karşılaştırmasını yapmak ve yeterli denetimin sağlanıp sağlanmadığını incelemektir. Sistemin modelinin tam olarak bilinmesi ile klasik denetleçlerle optimum tasarım sağlanabilmektedir. Zaten bu çalışmada gerçekleştirilen bulanık denetlecin klasik denetleçten daha iyi bir sonuç vermesi beklenmemektedir.

Çalışmada elde edilen sonuçlar grafikler halinde sunulmuştur. Bulanık mantık simulasyonlarında ve pratik uygulamada max-min çıkartım yöntemi ve ağırlık merkezi ile durulaştırma metodu kullanılmıştır. Çalışmanın pratik safhasında, basic programlama diliyle yazılan program bilgisayarda koşturularak, sistemin istenen tepkiyi vermedi sağlanmıştır. Bu aşamada karşılaşılan en büyük güçlük, bulanık ölçeklendirmedir. Zira programın işletim hızına bağlı olarak elde edilen değişim bilgisi başlangıçta tasarlanan ölçek ile uyuşmamıştır. Uygun ölçeklendirme yapılarak sistemden istenen verim alınmıştır.

Bu çalışma sonucunda aşağıdaki önerilerde bulunulabilir.

1- Sistemin modeli tam olarak biliniyorsa klasik denetim elemanlarının kullanılması daha verimlidir. Uygun hesaplanan denetim katsayıları ile sistemde en iyi denetim sağlanabilir.

2- Bulanık denetleç tasarımında iyi bir uzman bilgisi gerekmektedir. Fakat unutulmamalıdır ki, uzman bilgisi genelde sürecin son denetim elemanını yansıtmaktadır ve bu bilgi direk denetim kurallarını oluşturmamalıdır. Aksi halde bulanık denetleçlerde tutarsız çıkışlar elde edilir.

3- Bulanık denetlecin klasik tiplere göre bir başka avantajı, denetim stratejisini serbestçe belirleyebilmektedir. Ayar değeri, istenen bir değer için diğer özelliklere (örneğin oturma zamanına) göre oldukça kolay ayarlanabilmektedir.

KAYNAKLAR

1- F. MARTIN Mc NEIL, ELLEN THRO, 1994 Fuzzy Logic , Academic Press, USA

2- RAINER PALM, DIMITER DRIANKOV, HANS HELLENDOORN, 1997, Model Based Fuzzy Control, Spriner, Berlin

3- A. FRANK D'SOUZA, 1988, Design of Control Systems, Prentice Hall, USA

4- BENJAMİN C. KUO, 1995, Automatic Control Systems, Prentice Hall, USA

5- BENJAMİN C. KUO, DUANE C. HANSELMAN, 1994, Matlab Tools for Control System Analysis and Design, Prentice Hall, USA

6- DAVID HALLIDAY, ROBERT RESNİCK, 1962, Physics for Students of Science and Engineering, John Wiley & Sons, Tokyo

7- M. KEMAL SARIOĞLU, 1992, Dijital Kontrol Sistemleri , Sistem Yayıncılık, İstanbul

8- BEKİR ÇAKIR, 1997, Bulanık Kontrol Ders Notları, Kocaeli Ü., Kocaeli

9- ORHAN TUNÇÖZ, 1994, Otomatik Kontrol Ders Notları, Kocaeli Ü., Kocaeli

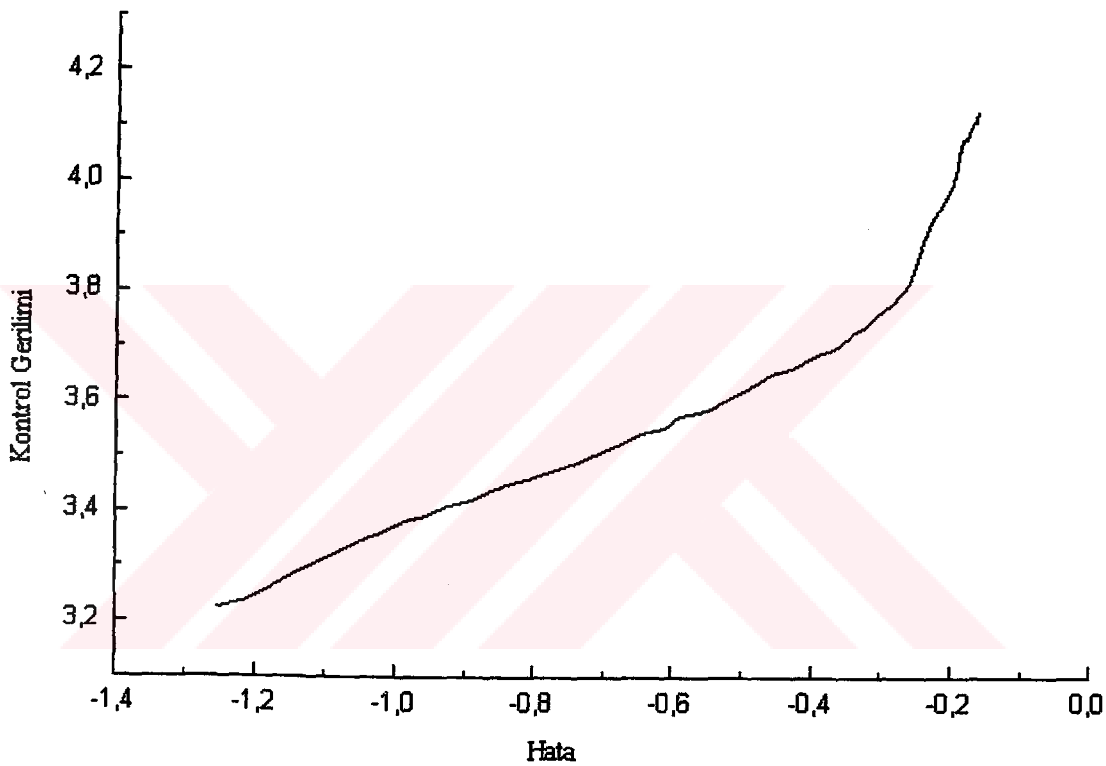
10- CHUEN CHIEN LEE, 1990, Fuzzy Logic in Control Systems, IEEE Transactions on Systems Man and Cybernetics, 20-2

11- YING-SHIEH KUNG, CHANG-MING LIAW, 1994, Afuzzy Controller Improving aLınear Model Follwing Controller for Motor Drives, IEEE Transactions on Fuzzy Sysstems, 2-3

EK A

Hata	Kontrol Gerilimi
-1,254	3,224
-1,211	3,24
-1,174	3,263
-1,133	3,293
-1,093	3,318
-1,058	3,343
-1,019	3,362
-0,99	3,38
-0,955	3,39
-0,923	3,409
-0,89	3,421
-0,861	3,437
-0,823	3,455
-0,8	3,462
-0,769	3,476
-0,739	3,489
-0,713	3,503
-0,68	3,52
-0,659	3,534
-0,637	3,545
-0,613	3,552
-0,589	3,572
-0,563	3,58
-0,545	3,589
-0,522	3,605
-0,498	3,619
-0,477	3,635
-0,459	3,649
-0,439	3,659
-0,422	3,664
-0,399	3,68
-0,363	3,699
-0,34	3,726
-0,322	3,738
-0,306	3,758
-0,287	3,774
-0,272	3,801
-0,26	3,818
-0,248	3,86
-0,24	3,889
-0,227	3,935

-0,218	3,95
-0,204	3,99
-0,196	4,03
-0,187	4,08
-0,181	4,08
-0,175	4,1
-0,17	4,11
-0,168	4,11
-0,165	4,13



Şekil A.1 Ölçüm sonuçları

EK

Bulanık Kontrol Programı

```
DECLARE SUB ReadKnowledgeBase ()
DECLARE SUB Fuzzymap ()

COMMON Rule() AS INTEGER

DIM SHARED NumInputs AS INTEGER
DIM SHARED NumOutputs AS INTEGER
DIM SHARED NumRules AS INTEGER
DIM SHARED NumFS(6) AS INTEGER
DIM SHARED PWFS(6, 10, 3, 1) AS INTEGER
DIM SHARED FinValue(4) AS INTEGER
DIM SHARED FoutValue(1) AS INTEGER
DIM SHARED SHAPE(6) AS INTEGER

DEFINT A-Z
z = 1
delta! = 1
dallan:
start! = TIMER

CLS
  SELECT CASE Q$
        CASE IS = "S"
              OPEN "TEST.FDT" FOR INPUT AS #1
              DO UNTIL EOF(1)
                    INPUT #1, FileString$
```

```
SELECT CASE FileString$
CASE "NUM_INPUTS"
        INPUT #1, Count$
        NumInputs = VAL(Count$)
CASE "INPUT1"
        INPUT #1, Count$
        NumFS(0) = VAL(Count$)
CASE "INPUT2"
        INPUT #1, Count$
        NumFS(1) = VAL(Count$)
CASE "INPUT3"
        INPUT #1, Count$
        NumFS(2) = VAL(Count$)
CASE "INPUT4"
        INPUT #1, Count$
        NumFS(3) = VAL(Count$)
CASE "INPUT5"
        INPUT #1, Count$
        NumFS(4) = VAL(Count$)
CASE "NUM_OUTPUTS"
        INPUT #1, Count$
        NumOutputs = VAL(Count$)
CASE "OUTPUT1"
        INPUT #1, Count$
        NumFS(5) = VAL(Count$)
CASE "OUTPUT2"
        INPUT #1, Count$
        NumFS(6) = VAL(Count$)
CASE "NUM_RULES"
        INPUT #1, Count$
        NumRules = VAL(Count$)
CASE ELSE
```

```
                    END SELECT
                LOOP
                CLOSE #1
                REDIM SHARED Rule(NumFS(0), NumFS(1), NumFS(2),
NumFS(3), NumFS(4)) AS INTEGER
                 ReadKnowledgeBase

        taban = 640
                OUT (taban + 24), 0
                OUT (taban + 17), 0
                DO
                LOOP UNTIL (INP(taban + 20) AND 1) = 1
                FinValue(0) = INP(taban + 18)
 'PRINT "Enter Input One: ", FinValue(0)
            IF NumInputs > 1 THEN
                FinValue(1) = (FinValue(0) - ilk) '/ delta! * 1000

 'PRINT "Enter Input Two: ", FinValue(1)
            END IF
            IF NumInputs > 2 THEN
                INPUT "Enter Input Three: ", FinValue(2)
            END IF
            IF NumInputs > 3 THEN
                INPUT "Enter Input Four: ", FinValue(3)
            END IF
            IF NumInputs > 4 THEN
                INPUT "Enter Input Five: ", FinValue(4)
            END IF


                 Fuzzymap
```

```
CLS
        'PRINT "Enter Input One: ", FinValue(0)
 'IF NumInputs > 1 THEN
        'PRINT "Enter Input Two: ", FinValue(1)
 'END IF
 'IF NumInputs > 2 THEN
        'PRINT "Enter Input Three: ", FinValue(2)
 'END IF
 'IF NumInputs > 3 THEN
        'PRINT "Enter Input Four: ", FinValue(3)
 'END IF
 'IF NumInputs > 4 THEN
        'PRINT "Enter Input Five: ", FinValue(4)
 'END IF

 'LOCATE 6, 1
 'PRINT "Output One: "; FoutValue(0)
 lsb1 = FoutValue(0) MOD 256
 msb1 = FIX(FoutValue(0) / 256)
 taban = 640
 LOCATE 10, 1
 'PRINT "msb1= "; msb1, "lsb1= "; lsb1
 OUT (taban + 8), lsb1
 OUT (taban + 9), msb1
 OUT (taban + 12), 0

 'LOCATE 15, 1
 'PRINT "delta"; delta!
 'IF NumOutputs > 1 THEN 'PRINT "Output Two: "; FoutValue(1)
```

```
        finish! = TIMER
                delta! = (finish! - start!) * 100000

        'LOCATE 10, 1
        'PRINT "delta= "; delta!
                        ilk = FinValue(0)

    GOSUB dallan

END
```

Alt programlar:

```
SUB Fuzzymap

DIM GrndArea AS LONG, GrndFirstMom AS LONG
DIM FuzSum AS LONG
DIM MFHIT(0 TO 4, 0 TO 10) AS LONG
DIM Consequent AS LONG
DIM Area1 AS LONG, FirstMoment1 AS LONG
DIM Area2 AS LONG, FirstMoment2 AS LONG
DIM Area4 AS LONG, FirstMoment4 AS LONG
DIM Area AS LONG, FirstMom AS LONG
DIM Inter1 AS LONG, Inter2 AS LONG
DIM HitsonDim(4) AS LONG
DIM FSHit(4, 10) AS LONG

FOR x = 0 TO NumInputs - 1
        IF (SHAPE(x) = 3) THEN
                FOR Y = 0 TO NumFS(x) - 1
                        IF (FinValue(x) <= PWFS(x, Y, 0, 0) OR FinValue(x) >=
PWFS(x, Y, 2, 0)) THEN
```

```
                    MFHIT(x, Y) = 0
                ELSEIF (FinValue(x) <= PWFS(x, Y, 1, 0) AND PWFS(x, Y,
0, 1) = 255) THEN
                    MFHIT(x, Y) = 255
                ELSEIF (FinValue(x) >= PWFS(x, Y, 1, 0) AND PWFS(x, Y,
2, 1) = 255) THEN
                    MFHIT(x, Y) = 255
                ELSEIF (FinValue(x) <= PWFS(x, Y, 1, 0)) THEN
                    MFHIT(x, Y) = ((FinValue(x) - PWFS(x, Y, 0, 0)) *
255&) / (PWFS(x, Y, 1, 0) - PWFS(x, Y, 0, 0))
                ELSE
                    MFHIT(x, Y) = ((PWFS(x, Y, 2, 0) - FinValue(x)) *
255&) / (PWFS(x, Y, 2, 0) - PWFS(x, Y, 1, 0))
                END IF
            NEXT Y
        ELSEIF (SHAPE(x) = 4) THEN
            FOR Y = 0 TO NumFS(x) - 1
                IF (FinValue(x) <= PWFS(x, Y, 0, 0) OR FinValue(x) >=
PWFS(x, Y, 3, 0)) THEN
                    MFHIT(x, Y) = 0
                ELSEIF (FinValue(x) <= PWFS(x, Y, 1, 0) AND PWFS(x, Y,
0, 1) = 255) THEN
                    MFHIT(x, Y) = 255
                ELSEIF (FinValue(x) >= PWFS(x, Y, 1, 0) AND PWFS(x, Y,
3, 1) = 255) THEN
                    MFHIT(x, Y) = 255
                ELSEIF (FinValue(x) <= PWFS(x, Y, 1, 0)) THEN
                    MFHIT(x, Y) = ((FinValue(x) - PWFS(x, Y, 0, 0)) *
255&) / (PWFS(x, Y, 1, 0) - PWFS(x, Y, 0, 0))
                ELSEIF (FinValue(x) <= PWFS(x, Y, 2, 0)) THEN
                    MFHIT(x, Y) = 255
                ELSE
```

```
                    MFHIT(x, Y) = ((PWFS(x, Y, 3, 0) - FinValue(x)) *
255&) / (PWFS(x, Y, 3, 0) - PWFS(x, Y, 2, 0))
                END IF
            NEXT Y
        END IF
NEXT x
GrndArea = 0
GrndFirstMom = 0
e = 0
DO
    d = 0
    DO
        c = 0
        DO
            b = 0
            DO
                a = 0
                DO
    FuzSum = 255
    IF (MFHIT(4, e) < FuzSum AND NumInputs > 4) THEN
        FuzSum = MFHIT(4, e)
    END IF
    IF (MFHIT(3, d) < FuzSum AND NumInputs > 3) THEN
        FuzSum = MFHIT(3, d)
    END IF
    IF (MFHIT(2, c) < FuzSum AND NumInputs > 2) THEN
        FuzSum = MFHIT(2, c)
    END IF
    IF (MFHIT(1, b) < FuzSum AND NumInputs > 1) THEN
        FuzSum = MFHIT(1, b)
    END IF
    IF (MFHIT(0, a) < FuzSum) THEN
```

```
            FuzSum = MFHIT(0, a)
        END IF
        IF (FuzSum <> 0) THEN
            Consequent = Rule(a, b, c, d, e)
            IF (Consequent = -1) THEN
                GOTO Skip
            END IF  'Don't care rule
            FOR I = 0 TO NumOutputs - 1
                IF (SHAPE(I + 5) = 1) THEN  'Output is a singleton
                    GrndArea = GrndArea + FuzSum
                    GrndFirstMom = GrndFirstMom + FuzSum * PWFS(5,
Consequent, 0, 0)
                ELSEIF (SHAPE(I + 5) = 3) THEN  'Output is a triangle
                    Inter1 = PWFS(5, Consequent, 0, 0) + (FuzSum *
(PWFS(5, Consequent, 1, 0) - PWFS(5, Consequent, 0, 0))) / 255&
                    Inter2 = PWFS(5, Consequent, 2, 0) - (FuzSum *
(PWFS(5, Consequent, 2, 0) - PWFS(5, Consequent, 1, 0))) / 255&
                    IF (PWFS(5, Consequent, 0, 1) = 255) THEN
                        Area1 = FuzSum * Inter1
                        FirstMoment1 = (Area1 * Inter1) / 2
                        Area2 = FuzSum * (Inter2 - Inter1)
                        FirstMoment2 = Area2 * (Inter2 - (Inter2 -
Inter1) / 2)
                    ELSE
                        Area1 = FuzSum * (Inter1 - PWFS(5,
Consequent, 0, 0)) / 2
                        FirstMoment1 = Area1 * (Inter1 - ((Inter1 -
PWFS(5, Consequent, 0, 0)) / 3))
                        Area2 = FuzSum * (Inter2 - Inter1)
                        FirstMoment2 = Area2 * (Inter1 + (Inter2 -
Inter1) / 2)
                    END IF
```

```
IF (PWFS(5, Consequent, 2, 1) = 255) THEN
        Area4 = FuzSum * (PWFS(5, Consequent, 2, 0)
- Inter2)
        FirstMoment4 = Area4 * (Inter2 + ((PWFS(5,
Consequent, 2, 0) - Inter2) / 2))
ELSE
        Area4 = FuzSum * (PWFS(5, Consequent, 2, 0)
- Inter2) / 2
        FirstMoment4 = Area4 * (Inter2 + ((PWFS(5,
Consequent, 2, 0) - Inter2) / 3))
END IF
Area = Area1 + Area2 + Area4
FirstMom = FirstMoment1 + FirstMoment2 +
FirstMoment4

GrndArea = GrndArea + Area
GrndFirstMom = GrndFirstMom + FirstMom
ELSEIF (SHAPE(I + 5) = 4) THEN  'Is a trapezoid
Inter1 = PWFS(5, Consequent, 0, 0) + (FuzSum *
(PWFS(5, Consequent, 1, 0) - PWFS(5, Consequent, 0, 0))) / 255
Inter2 = PWFS(5, Consequent, 3, 0) - (FuzSum *
(PWFS(5, Consequent, 3, 0) - PWFS(5, Consequent, 2, 0))) / 255
IF (PWFS(5, Consequent, 0, 1) = 255) THEN
        Area1 = FuzSum * Inter1
        FirstMoment1 = (Area1 * Inter1) / 2
        Area2 = FuzSum * (Inter2 - Inter1)
        FirstMoment2 = Area2 * (Inter2 - (Inter2 -
Inter1) / 2)
ELSE
        Area1 = (FuzSum * (Inter1 - PWFS(5,
Consequent, 0, 0))) / 2
```

```
                        FirstMoment1 = Area1 * (Inter1 - (Inter1 -
PWFS(5, Consequent, 0, 0)) / 3)
                        Area2 = FuzSum * (Inter2 - Inter1)
                        FirstMoment2 = Area2 * (Inter1 + (Inter2 -
Inter1) / 2)
                    END IF
                    IF (PWFS(5, Consequent, 3, 1) = 255) THEN
                        Area4 = FuzSum * (PWFS(5, NumFS(5) - 1, 3,
0) - Inter2)
                        FirstMoment4 = Area4 * ((PWFS(5, NumFS(5)
- 1, 3, 0) - Inter2) / 2)
                    ELSE
                        Area4 = FuzSum * (PWFS(5, Consequent, 3, 0)
- Inter2) / 2
                        FirstMoment4 = Area4 * (Inter2 + (PWFS(5,
Consequent, 3, 0) - Inter2) / 3)
                    END IF
                    Area = Area1 + Area2 + Area4
                    FirstMom = FirstMoment1 + FirstMoment2 +
FirstMoment4
                     GrndArea = GrndArea + Area
                     GrndFirstMom = GrndFirstMom + FirstMom
                END IF
            NEXT I
Skip:
    END IF
                        a = a + 1
                    LOOP WHILE (a < NumFS(0))
                    b = b + 1
                LOOP WHILE (b < NumFS(1) AND NumInputs > 1)
                c = c + 1
            LOOP WHILE (c < NumFS(2) AND NumInputs > 2)
```

```
            d = d + 1
        LOOP WHILE (d < NumFS(3) AND NumInputs > 3)
        e = e + 1
LOOP WHILE (e < NumFS(4) AND NumInputs > 4)
IF (GrndArea <> 0) THEN
        FoutValue(0) = (GrndFirstMom / GrndArea)
ELSE
        FoutValue(0) = 0
END IF
END SUB

SUB ReadKnowledgeBase STATIC

OPEN "TEST.FDT" FOR INPUT AS #1
SEEK #1, 1
DO UNTIL EOF(1)
        INPUT #1, FileString$
        SELECT CASE FileString$
        CASE "NUM_INPUTS"
                INPUT #1, Count$
                NumInputs = VAL(Count$)
        CASE "INPUT1"
                INPUT #1, Count$
                NumFS(0) = VAL(Count$)
        CASE "INPUT2"
                INPUT #1, Count$
                NumFS(1) = VAL(Count$)
        CASE "INPUT3"
                INPUT #1, Count$
                NumFS(2) = VAL(Count$)
        CASE "INPUT4"
                INPUT #1, Count$
```

```
                CASE "START"
                    FOR J = 0 TO NumFS(I) - 1
                        FOR K = 0 TO SHAPE(I) - 1
                            INPUT #1, Count$
                            PWFS(I, J, K, 0) = VAL(Count$)
                            INPUT #1, Count$
                            PWFS(I, J, K, 1) = VAL(Count$)
                        NEXT K
                    NEXT J
                END SELECT
                LOOP WHILE FileString$ <> "END$"
            NEXT I
        CASE "OUTPUTS FUZZY SETS"
            FOR I = 0 TO NumOutputs - 1
                DO
                INPUT #1, FileString$
                SELECT CASE FileString$
                CASE "OUTPUT"
                    INPUT #1, Count$
                    M = 5
                CASE "COUNT"
                    INPUT #1, Count$
                    IF NumFS(I + M) <> VAL(Count$) THEN
                        END
                    END IF
                CASE "SHAPE"
                    INPUT #1, Count$
                    SHAPE(I + M) = VAL(Count$)
                CASE "START"
                    FOR J = 0 TO NumFS(I + M) - 1
                        FOR K = 0 TO SHAPE(I + M) - 1
                            INPUT #1, Count$
```

```
        NumFS(3) = VAL(Count$)
    CASE "INPUT5"
        INPUT #1, Count$
        NumFS(4) = VAL(Count$)
    CASE "NUM_OUTPUTS"
        INPUT #1, Count$
        NumOutputs = VAL(Count$)
    CASE "OUTPUT1"
        INPUT #1, Count$
        NumFS(5) = VAL(Count$)
    CASE "OUTPUT2"
        INPUT #1, Count$
        NumFS(6) = VAL(Count$)
    CASE "NUM_RULES"
        INPUT #1, Count$
        NumRules = VAL(Count$)
    CASE "INPUTS FUZZY SETS"
        FOR I = 0 TO NumInputs - 1
            DO
            INPUT #1, FileString$
            SELECT CASE FileString$
            CASE "INPUT"
                INPUT #1, Count$
                M = 0
            CASE "COUNT"
                INPUT #1, Count$
                IF NumFS(I) <> VAL(Count$) THEN
                    END
                END IF
            CASE "SHAPE"
                INPUT #1, Count$
                SHAPE(I) = VAL(Count$)
```

```
                        PWFS(I + M, J, K, 0) = VAL(Count$)
                        IF (SHAPE(I + M) > 1) THEN
                            INPUT #1, Count$
                            PWFS(I + M, J, K, 1) =
VAL(Count$)
                        END IF
                    NEXT K
                NEXT J
            'CASE "END$"
            'CASE ELSE

            END SELECT
            LOOP WHILE FileString$ <> "END$"
        NEXT I
    CASE "RULES"
        FOR H = 0 TO 4
            IF NumFS(H) = 0 THEN Temp(H) = 0 ELSE Temp(H) =
NumFS(H) - 1
        NEXT H
        FOR I = 0 TO Temp(0)
            FOR J = 0 TO Temp(1)
                FOR K = 0 TO Temp(2)
                    FOR L = 0 TO Temp(3)
                        FOR M = 0 TO Temp(4)
                            INPUT #1, Rule(I, J, K, L, M)
                        NEXT M
                    NEXT L
                NEXT K
            NEXT J
        NEXT I
    CASE ELSE
```

```
        END SELECT
    LOOP
    CLOSE #1
    END SUB
```

ÖZGEÇMİŞ

21. 08. 1974 İzmit doğumlu. İlk öğrenimini İzmit'te tamamladıktan sonra Kocaeli Anadolu Lisesinde orta ve lise öğrenimini tamamladı. 1992 yılında girdiği Yıldız Üniversitesinde 1996 yılında lisans öğrenimini bitirdikten sonra aynı yıl başlamış olduğu yüksek lisans öğreniminde tez aşamasındadır.