

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**BULANIK KONTROLÖRLERDE
DİLSEL BİÇİMLENDİRİCİLERİN OPTİMAL
AYARLANMASINA DAYALI TASARIM YÖNTEMİ**

**YÜKSEK LİSANS TEZİ
Elek. Müh. Cenk ULU**

Anabilim Dalı : ELEKTRİK MÜHENDİSLİĞİ

**Programı : KONTROL ve OTOMASYON
MÜHENDİSLİĞİ**

AĞUSTOS 2006

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**BULANIK KONTROLÖRLERDE
DİLSEL BİÇİMLENDİRİCİLERİN OPTİMAL
AYARLANMASINA DAYALI TASARIM YÖNTEMİ**

**YÜKSEK LİSANS TEZİ
Elek. Müh. Cenk ULU
(504041103)**

**Tezin Enstitüye Verildiği Tarih : 15 Ağustos 2006
Tezin Savunulduğu Tarih : 22 Ağustos 2006**

**Tez Danışmanı : Prof.Dr. İbrahim EKSİN
Diğer Jüri Üyeleri Prof.Dr. Müjde GÜZELKAYA
Yrd.Doç.Dr. Şima Etaner UYAR**

AĞUSTOS 2006

ÖNSÖZ

Bu çalışmanın oluşturulmasında yardımlarını esirgemeyen başta danışman hocam Prof. Dr. İbrahim EKSİN olmak üzere Prof. Dr. Müjde GÜZELKAYA, Araş. Gör. Engin YEŞİL ve sevgili arkadaşım Gaye SAĞLAM' a, yaşam boyu maddi ve manevi desteklerinden ötürü aileme ve son olarak can dostum Kerem TOKER' e sonsuz teşekkürler ederim.

Ağustos, 2006

Cenk ULU

İÇİNDEKİLER

KISALTMALAR	iv
TABLO LİSTESİ	v
ŞEKİL LİSTESİ	vi
SEMBOL LİSTESİ	ix
ÖZET	x
SUMMARY	xi
1. GİRİŞ	1
2. BULANIK KONTROLÖRLER	6
2.1. Bulanık Kontrolörlerin Temel Yapısı	6
2.2. Bulanık Kontrolör Çeşitleri	7
3. DİLSEL BİÇİMLENDİRİCİLER	11
4. ÖNERİLEN YÖNTEMİN YAPISI	16
4.1. Ayarlama Mekanizmasının Yapısı	17
5. SİMÜLASYON SONUÇLARI	22
5.1. Uygulamalarda Kullanılan PI Tipi Bulanık Kontrolörün Yapısı	23
5.2. Çevrim Dışı Yöntemlerle Dilsel Biçimlendiricilerin İncelenmesi	25
5.2.1. Zaman sabitinin iki katı büyüklüğünde ölü zaman değerine sahip doğrusal sistem üzerinde yapılan çalışmalar	26
5.2.2. Doğrusal olmayan sistem üzerinde yapılan çalışmalar	34
5.3. Ayar Parametre Sayısının Azaltılması ve Yöntemin Zaman Sabitinin Beş Katı Büyüklüğünde Ölü Zaman Değerine Sahip Doğrusal Sistem Üzerinde Uygulanması	42
5.4. Yöntemin Doğrusal Olmayan Sistem Üzerinde Uygulanması	49
6. SONUÇLAR VE TARTIŞMALAR	53
KAYNAKLAR	55
EK 1	58
ÖZGEÇMİŞ	104

KISALTMALAR

PD	: Proportional-Derivative (Orantı-Türev)
PI	: Proportional-Integral (Orantı-İntegral)
PID	: Proportional-Integral-Derivative (Orantı-İntegral-Türev)
PB	: Pozitif Büyük
NB	: Negatif Büyük
S	: Sıfır
GA	: Genetik Algoritma

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1. Çıkarım mekanizmasında kullanılan kural tablosu	17
Tablo 5.1. Bulanık kontrolörde kullanılan kural tablosu	24
Tablo 5.2. Doğrusal sistem için düzgün seçilmemiş ölçekleme çarpanları	27
Tablo 5.3. Doğrusal sistem için ölçekleme çarpanlarının düzgün seçilmemesi durumunda belirlenen ortak dilsel biçimlendirici değerleri	28
Tablo 5.4. Doğrusal sistem için ölçekleme çarpanlarının düzgün seçilmemesi durumunda belirlenen farklı dilsel biçimlendirici değerleri	30
Tablo 5.5. Doğrusal sistem için düzgün seçilmiş ölçekleme çarpanları	32
Tablo 5.6. Doğrusal sistem için ölçekleme çarpanlarının düzgün seçilmesi durumunda belirlenen dilsel biçimlendirici değerleri	33
Tablo 5.7. Doğrusal olmayan sistem için düzgün seçilmemiş ölçekleme çarpanları	35
Tablo 5.8. Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmemesi durumunda belirlenen ortak dilsel biçimlendirici değerleri	36
Tablo 5.9. Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmemesi durumunda belirlenen farklı dilsel biçimlendirici değerleri	36
Tablo 5.10. Doğrusal olmayan sistem için düzgün seçilmiş ölçekleme çarpanları	37
Tablo 5.11. Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmesi durumunda belirlenen ortak dilsel biçimlendirici değerleri	38
Tablo 5.12. Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmesi durumunda belirlenen farklı dilsel biçimlendirici değerleri	38
Tablo 5.13. Doğrusal sistem için düzgün seçilmiş ölçekleme çarpanları	42
Tablo 5.14. Doğrusal sistemde sekiz parametrenin aratıldığı yöntemin ilk hali için belirlenen k değerleri	43
Tablo 5.15. Doğrusal sistem için ölçekleme çarpanlarının düzgün seçilmesi durumunda belirlenen farklı dilsel biçimlendirici değerleri	44
Tablo 5.16. İki parametrenin aratıldığı yöntemin son hali için belirlenen k değerleri	47
Tablo 5.17. Doğrusal olmayan sistem için düzgün seçilmemiş ölçekleme çarpanlarının kullanılması durumunda belirlenen k parametreleri	50
Tablo 5.18. Doğrusal olmayan sistem için düzgün seçilmiş ölçekleme çarpanlarının kullanılması durumunda belirlenen k parametreleri	50

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1(a) : Temel parametreleri değiştirerek üyelik fonksiyonlarını ayarlama yöntemi	4
Şekil 1.1(b) : Dilsel biçimlendiriciler kullanarak üyelik fonksiyonlarını ayarlama yöntemi	4
Şekil 2.1 : Bulanık kontrolörün temel yapısı	6
Şekil 2.2 : PD tip bulanık kontrolörün blok diyagramı	8
Şekil 2.3 : PI tip bulanık kontrolörün blok diyagramı	8
Şekil 2.4 : İki kural tablosu kullanarak elde edilen PID tip bulanık kontrolör blok diyagramı	9
Şekil 2.5 : Tek kural tablosu kullanılarak elde edilen PID tip bulanık kontrolör blok diyagramı	9
Şekil 3.1 : “Çok daha çok” ve “daha az” dilsel biçimlendiricilerin üyelik fonksiyonları üzerindeki etkisi	12
Şekil 3.2 : Zıt yönlü dilsel biçimlendiricilerin üyelik fonksiyonu üzerindeki etkisi	13
Şekil 3.3(a) : NB, S ve PB üyelik fonksiyonları üzerinde “daha az” dilsel biçimlendiricisinin etkisi	14
Şekil 3.3(b) : NB, S ve PB üyelik fonksiyonları üzerinde “çok daha çok” dilsel biçimlendiricisinin etkisi	14
Şekil 4.1 : Çıkarım mekanizmasında kullanılan üyelik fonksiyonları	18
Şekil 4.2 : Önerilen yöntemin blok diyagramı	21
Şekil 5.1 : Doğrusal sistem uygulaması için PI tipi bulanık kontrolörde kullanılan üyelik fonksiyonları	24
Şekil 5.2 : Doğrusal olmayan sistem uygulaması için PI tipi bulanık kontrolörde kullanılan üyelik fonksiyonları	24
Şekil 5.3 : Bölgesel dilsel biçimlendirici belirleme yönteminin üyelik fonksiyonları üzerinde gösterimi	26
Şekil 5.4 : Çevrim dışı yöntemlerle dilsel biçimlendiricilerin incelendiği ölü zamanlı doğrusal sistemin Simulink blok diyagramı	27
Şekil 5.5 : Doğrusal sistem için düzgün seçilmemiş ölçekleme çarpanları ile elde edilen sistem yanıtı	27
Şekil 5.6(a) : Doğrusal sistem için belirlenen tek bölgesi ortak dilsel biçimlendirici değerleri ile elde edilen yeni üyelik fonksiyonları ..	29
Şekil 5.6(b) : Doğrusal sistem için belirlenen iki bölgesi ortak dilsel biçimlendirici değerleri ile elde edilen yeni üyelik fonksiyonları ..	29
Şekil 5.6(c) : Doğrusal sistem için belirlenen dört bölgesi ortak dilsel biçimlendirici değerleri ile elde edilen yeni üyelik fonksiyonları ..	29
Şekil 5.6(d) : Doğrusal sistem için belirlenen sekiz bölgesi ortak dilsel biçimlendirici değerleri ile elde edilen yeni üyelik fonksiyonları ..	29

Şekil 5.7	: Doğrusal sistem için belirlenen ortak dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları	29
Şekil 5.8(a)	: Doğrusal sistem için belirlenen tek bölgeli farklı dilsel biçimlendirici değerleri ile elde edilen yeni üyelik fonksiyonları	31
Şekil 5.8(b)	: Doğrusal sistem için belirlenen iki bölgeli farklı dilsel biçimlendirici değerleri ile elde edilen yeni üyelik fonksiyonları	31
Şekil 5.9	: Doğrusal sistem için belirlenen farklı dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları	31
Şekil 5.10	: Doğrusal sistem için düzgün seçilmiş ölçekleme çarpanları ile elde edilen sistem yanıtı	32
Şekil 5.11	: Doğrusal sistem için belirlenen dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları ...	33
Şekil 5.12	: Doğrusal sistem için düzgün seçilmiş ve düzgün seçilmemiş ölçekleme çarpanları ile elde edilen en iyi sonuçların karşılaştırılması	34
Şekil 5.13	: Doğrusal olmayan sistemin kapalı çevrim Simulink blok diyagramı	35
Şekil 5.14(a)	: Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmemesi durumunda belirlenen ortak dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları ...	37
Şekil 5.14(b)	: Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmemesi durumunda belirlenen farklı dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları ...	37
Şekil 5.15(a)	: Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmesi durumunda belirlenen ortak dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları ...	39
Şekil 5.15(b)	: Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmesi durumunda belirlenen farklı dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları ...	39
Şekil 5.16	: Doğrusal olmayan sistem için düzgün seçilmiş ve düzgün seçilmemiş ölçekleme çarpanları ile elde edilen en iyi sonuçların karşılaştırılması	40
Şekil 5.17	: Sekiz parametrenin aratıldığı yöntemin ilk hali için ölü zamanlı doğrusal sistemin Simulink blok diyagramı	43
Şekil 5.18	: Doğrusal sistemde sekiz parametrenin aratıldığı yöntemin ilk halinin uygulanması sonucu elde edilen hata, hatanın türevi, kontrol işareti ve sistem yanıtı	44
Şekil 5.19	: Doğrusal sistemde sekiz parametrenin aratıldığı yöntemin ilk halinin uygulanması sonucu kontrolöre çevrim içi olarak atanan dilsel biçimlendiricilerin grafiği	45
Şekil 5.20	: Doğrusal sistem üzerinde sekiz parametrenin aratıldığı yöntemin ilk halinin uygulanması ve tek bölgeli farklı dilsel biçimlendirici değerlerinin atanması ile elde edilen sonuçların karşılaştırılması	45
Şekil 5.21	: İki parametrenin ayarlandığı yöntemin son halinin Simulink blok diyagramı	46

Şekil 5.22 : İki parametrenin aratıldığı yöntemin son hali için belirlenen k değerlerinin uygulanması sonucu doğrusal sistemde elde edilen hata, hatanın türevi, kontrol işareti ve sistem yanıtı	47
Şekil 5.23 : Yöntemin son halinin uygulanması ile doğrusal sistemde kontrolöre çevrim içi olarak atanan dilsel biçimlendirici değerleri ..	48
Şekil 5.24 : Önerilen yöntemin iki parametre aratıldığı son hali ve sekiz parametre aratıldığı ilk hali ile elde edilen sonuçların karşılaştırılması	48
Şekil 5.25 : Önerilen yöntemin uygulandığı doğrusal olmayan sistem kapalı çevrim Simulink blok diyagramı	49
Şekil 5.26 : Doğrusal olmayan sistemde yeni yöntemin ve bölgesel dilsel biçimlendiricilerin uygulanması sonucu elde edilen sistem yanıtları	51

SEMBOL LİSTESİ

$\mu(x)$	: Üyelik derecesi
R^+	: Pozitif reel sayılar
e	: Hata
Δe	: Hatanın değişimi
u	: Kontrol işareti
Δu	: Kontrol işaretinin değişimi
α	: Dilsel biçimlendirici değeri
$DAR(x)$	: Daraltıcı tip dilsel biçimlendirici
$GEN(x)$	: Genişletici tip dilsel biçimlendirici
$ZIT(x)$	: Zıt yönlü dilsel biçimlendirici
L	: Simetri eksen
z_0	: Durulama ile elde edilen keskin çıkış değeri
λ	: Çıkarım mekanizmasının çıkış değeri
k	: Dilsel biçimlendirici belirlenmesi için kullanılan çıkış ölçekleme çarpanı
$H(s)$	: Transfer fonksiyonu
$f(h)$	: Performans indeksi
m	: Örnekleme sayısı
mp	: Aşım değeri
tp	: Tepe zamanı
ts	: Yerleşme zamanı
ess	: Kararlı hal hatası
$\mu_x(\omega_i)$	: Bulanık çıkarım değeri
ω_i	: Çıkarıma denk düşen tekil tip çıkış üyelik fonksiyonu değeri
$g1, g2$	: Giriş ölçekleme çarpanları
$c1$	: Çıkış ölçekleme çarpanı

BULANIK KONTROLÖRLERDE DİLSEL BİÇİMLENDİRİCİLERİN OPTİMAL AYARLANMASINA DAYALI TASARIM YÖNTEMİ

ÖZET

Klasik kontrol yaklaşımları büyük ölçüde kontrol edilecek sistemin doğrusallaştırılmış modeline ihtiyaç duymaktadır. Oysa gerçek dünyada çoğu sistem doğrusal olmayan ve daha karmaşık bir yapıya sahip olduğundan dolayı genellikle bu sistemleri tanımlamak için yeterli matematiksel ifadeler mevcut değildir. Bu yüzden ilgili kontrolörlerin tasarlanması oldukça zordur. Diğer yandan bazı sistemler için sistem yapısı belirlenebilse bile sistemin ya da modelin parametrelerinin belirlenmesi sorunları ortaya çıkabilmekte, bu tür sistemler için kontrolör tasarlamak zaman alıcı olmakta ve tasarım yüksek maliyetlere neden olabilmektedir.

Bulanık mantık yöntemi yardımıyla kontrolör tasarımı, klasik yöntemlerin etkisiz kaldığı en karmaşık yapıli sistemlerde bile daha basit bir şekilde gerçekleştirilebilmektedir. Bulanık mantık yöntemi oldukça basit ve insan akıl yürütme şekline benzerdir ve bu yüzden tasarım yöntemleri matematiksel modelin bilinmesine gerek duymaz. Bu nedenle bulanık kontrolörler klasik yöntemlere makul ve etkili alternatifler sağlamaktadır. Yine de bulanık kontrolörlerin tasarımında uzman bilgi ve deneyimlerine ihtiyaç duyulmaktadır. Uzman kişi değişkenlere uygun dilsel tanımlamaları yaparak üyelik fonksiyonlarını ve kontrol için uygun kuralları belirler. Kurallar, kural sayısı ve dilsel ifade sayısının uzman arzusuna bağlı olması nedeniyle bulanık kontrolörün performansı doğal olarak bu uzman kişinin doğru seçimleri yapmasına bağlı olmaktadır. Bir çok araştırmacı bulanık kontrolör tasarımında uzman bilgisine olan bu bağılılığı ortadan kaldırmak için yeni metotlar önermişlerdir. Üyelik fonksiyonları ve kuralların çeşitli algoritmalar tarafından otomatik olarak belirlenebilmesine rağmen, kullanılan değişken sayısı arttığında, kontrol edilen sistemin parametreleri değiştiğinde ya da çevre koşulları değiştiğinde çeşitli zorluklar ortaya çıkmaktadır. Bu noktada, basit ve az sayıda üyelik fonksiyonu ve kural içeren uyarlamalı bir bulanık kontrolör tasarımına ihtiyaç duyulmaktadır.

Bugüne kadar üyelik fonksiyonlarının ayarlanması için yapılan çalışmaların neredeyse tamamında, üyelik fonksiyonları çevrim dışı olarak belirlenmiş ve bulunan optimal ya da optimale yakın bu değerler kontrolörün yapısına işlenerek sabit bir üyelik fonksiyonu yapısı kullanılmıştır. Bu çalışmada, bulanık kontrolörün üyelik fonksiyonlarını dilsel biçimlendiriciler ile çevrim içi olarak değiştirmek için yeni bir yöntem önerilmiştir. Ayarlamalar bir çıkarım mekanizması yoluyla yapılmaktadır. Çıkarım mekanizması, kontrol edilen sistemden ölçülen hata ve hatanın türevi sinyallerine göre üyelik fonksiyonlarının şeklini değiştirmektedir. Bu yeni yöntemle bulanık kontrolörün yapısal parametreleri çevrim içi olarak değiştirilebildiğinden, kontrol edilmesi zor bir sistemde bile iyi bir performans elde edilebilmektedir. Bu yeni yöntem doğrusal olmayan ve ölü zamanlı doğrusal sistemler üzerinde uygulanmış ve yöntemin geçerliliği simülasyon sonuçları ile ispat edilmiştir.

A DESIGN METHODOLOGY FOR FUZZY LOGIC CONTROLLERS BASED ON OPTIMAL TUNING OF LINGUISTIC HEDGES

SUMMARY

Classical control approaches mostly rely on the linearized models of the controlled plants. However, there is generally no adequate mathematical expression to describe most of the systems since they usually possess nonlinear and more complex structures in the real world. Therefore, the related controllers are difficult to design. On the other hand, even if the system structure can be determined for some systems, the problems of determining parameters of system or the model may emerge, it takes a very long time to design a controller for such systems and it may be costly.

The controller design with the aid of fuzzy logic theory can be implemented more simply even on the systems having most complex structure where the classical methods become inefficient. The fuzzy logic theory is very simple and similar to human reasoning and therefore the design methodologies do not require a mathematical modeling knowledge. For this reason, fuzzy logic controllers provide reasonable, effective alternatives to classical methods. However, the knowledge and experience of experts are needed in the design of fuzzy logic controllers. Experts determine membership functions and appropriate rules for control by using appropriate linguistic definitions to variables. Since the rules, the rule numbers and the linguistic definition numbers all depend on the experts, the performance of fuzzy logic controller naturally depends on determining the correct choices of the experts. Several researchers have proposed new methods for removing this dependence of experts in fuzzy logic controller design. Although the rules and the membership functions can be determined automatically by variety of algorithms, various difficulties emerge when the number of variables used is increased or when the parameters of the controlled plants are varied or when the conditions of the environment are changed. At this point, the design of an adaptive fuzzy logic controller with simple and small number of membership functions and rules is necessary.

In almost all of the studies done for adjustment of membership functions up to now, the membership functions are determined in offline manner and constant membership function structure is used by putting this optimal or near optimal values into controller structure. In this study, a new method is proposed to modify the membership functions of the fuzzy logic controllers in an online manner with linguistic hedges. The adjustments are done through a fuzzy inference mechanism. Inference mechanism modifies the shape of membership functions according to the error and error rate signals measured from the controlled system. Since the structural parameters of fuzzy logic controllers can be modified in online manner with this new method, a good performance can be obtained even if the system is too difficult to

control. This new method has been applied to the nonlinear and the time delay linear systems, and the effectiveness of the method is demonstrated by simulation results.

1. GİRİŞ

Klasik kontrol yaklaşımlarında büyük ölçüde kontrol edilecek sistemin doğrusallaştırılmış modeline ya da matematiksel modeline ihtiyaç duyulmaktadır. Oysa gerçek dünyada sistemler çoğu zaman doğrusal olmayan karmaşık bir yapıya sahip olduklarından dolayı genellikle bu sistemleri tanımlamak için yeterli matematiksel ifadeler mevcut değildir. Bu yüzden bu tür sistemlerin ilgili kontrol modellerinin oluşturulması oldukça zordur. Diğer yandan bazı sistemler için sistem yapısı belirlenebilse bile parametrelerinin belirlenememesi gibi sorunlar ortaya çıkmakta ya da modelleme doğru olarak yapılabilse bile bu zaman alıcı olmakta ve elde edilen bu modelin kontrolör tasarımında kullanımı yüksek maliyetlere neden olabilmektedir.

1965 yılında Zadeh tarafından ortaya atılan bulanık mantık kuramı [1] ile matematiksel modele gerek duyulmadan, daha basit ve insan düşünce yapısına dönük yaklaşık akıl yürütme yoluyla, klasik yöntemlerin etkisiz kaldığı en karmaşık yapıli sistemlerde bile kontrolör tasarımı gerçekleştirilebilmektedir. Bu yüzden klasik kontrolörlerle karşılaştırıldığında bulanık kontrolörler ile daha makul ve etkili sonuçlar elde edilebilmektedir.

Gerçek dünyada büyük ölçüde bulanıklık hakimdir. Bu yüzden 0 ve 1 (var/yok) ile temsil edilen klasik mantık bu işleyişi yeterince ifade edememektedir. İnsan akli olaylar karşısında klasik mantık gibi var-yok, sıcak-soğuk gibi kesin ifadeler yerine ılık, serin, biraz soğuk, oldukça sıcak gibi bulanık ifadeler kullanmaktadır. Bulanık mantık insana özgü bu yaklaşık akıl yürütmeyi temel alarak oluşturulan bir kuramdır.

Bulanık mantıkta giriş değerlerine SICAK, ILIK, SERİN gibi dilsel değişkenler atanır ve uzmanın belirlediği uygun kurallar kullanılarak bir çıkarım gerçekleştirilir. Bulanık mantığın güçlü yönlerinden biri bu sözel değişkenler yoluyla hesaplama yapabilmesidir.

Bulanık mantığın temel özellikleri şu şekilde ifade edilebilir [2]:

- Bulanık mantıkta, kesin değerlere dayanan düşünme yerine yaklaşık akıl yürütme kullanılır.
- Bulanık mantıkta her şey $[0,1]$ aralığında belirli bir derece ile gösterilir.
- Bulanık mantıkta bilgi BÜYÜK, KÜÇÜK, ÇOK AZ gibi dilsel ifadeler şeklindedir.
- Bulanık çıkarım işlemi dilsel ifadeler arasında tanımlanan kurallar ile yapılır.
- Her mantıksal sistem bulanık olarak ifade edilebilir.
- Bulanık mantık matematiksel modeli çok zor elde edilen sistemler için oldukça uygundur.
- Bulanık mantık tam olarak bilinmeyen ya da eksik girilen bilgilere göre işlem yapma yeteneğine sahiptir.

Yukarıda bahsedilen özelliklerinden ötürü klasik yöntemlere bir alternatif oluşturan bulanık mantık uygulamaları, klasik yöntemlerin üstesinden gelemediği durumlara basit ve daha etkili çözümler getirebilmektedirler. Bulanık mantığın ilk uygulaması, Mamdani tarafından 1974 yılında bir buhar makinesinin bulanık denetiminin gerçekleştirilmesi olmuştur [3]. O günden günümüze bulanık mantık uygulamaları hızlı bir şekilde gelişmiş ve bir çok modern uygulama alanında geniş yer edinmiştir.

Bulanık mantığın mühendislikteki uygulama alanları genel olarak aşağıdaki gibi gösterilebilir [4]:

- Uçaklar ve uzay araçları
- Otomobiller
- Raylı sistemler
- Üretim endüstrisi
- Güç endüstrisi
- Süreç kontrol
- Robotik
- Görüntü ve sinyal işleme

- Ev aletleri
- Sağlık endüstrisi

Bulanık kontrolörlerin tasarım aşamasında uzman bilgi ve deneyimlerine ihtiyaç duyulmaktadır. Uzman kişi değişkenlere uygun dilsel tanımlamaları yaparak üyelik fonksiyonlarını ve kontrol için uygun kuralları belirler. Kurallar, kural sayısı ve dilsel ifade sayısının uzman arzusuna bağlı olması nedeniyle bulanık kontrolörün performansı büyük ölçüde bu uzman kişinin en doğru seçimleri yapmasına bağlı olmaktadır. Bir çok araştırmacı bulanık kontrolör tasarımında uzman bilgisine olan bu bağılılığı ortadan kaldırmak için yeni metotlar önermişlerdir.

Genel olarak bulanık kontrolörlerin tasarım parametrelerini iki grupta toplayabiliriz [5]:

- Yapısal parametreler
- Ayarlama parametreleri.

Yapısal parametreler, bulanık kontrolörün giriş ve çıkış değişkenlerini, bulanık dilsel kümeleri, üyelik fonksiyonlarını, bulanık kuralları, bulanık çıkarım mekanizmasını ve durulama mekanizmasını içerir. Ayarlama parametreleri ise giriş/çıkış ölçekleme çarpanlarını ve üyelik fonksiyonu parametrelerini içerir. Genellikle yapısal parametreler çevrim dışı durumda belirlenir ve böylece kontrolörün temel yapısı tasarlanır. Örneğin üyelik fonksiyonu tipi, üyelik fonksiyonu sayısı, giriş ve çıkış değişkeni sayısı, kurallar ve kural sayısı, durulama yöntemi çevrim dışı olarak belirlenir ve süreç boyunca değişmez. Ayarlama parametreleri ise genellikle belirlenen kontrolörün performansını arttırmak, uyarlama ve kararlılığı sağlamak için çevrim içi olarak hesaplanır. Bu çalışmada ise önerilen yeni yöntem ile bulanık kontrolörün yapısal parametrelerinden olan üyelik fonksiyonlarının şekli çevrim içi olarak ayarlanabilmekte ve böylece basit yapıdaki bir bulanık kontrolörle bile sistem performansı büyük ölçüde arttırılabilmektedir.

Bu çalışmanın konusu olan üyelik fonksiyonlarının ayarlanmasına gelince, literatürde uygulanan üyelik fonksiyonlarını ayarlama yöntemlerini iki grupta toplayabiliriz [6]:

- Temel parametreleri değiştirmek
- Dilsel biçimlendiriciler kullanmak.

En yaygın yöntemlerden biri üyelik fonksiyonunun temel parametrelerini değiştirmektir. Örneğin üçgen yapıda bir üyelik fonksiyonu göz önüne alınırsa:

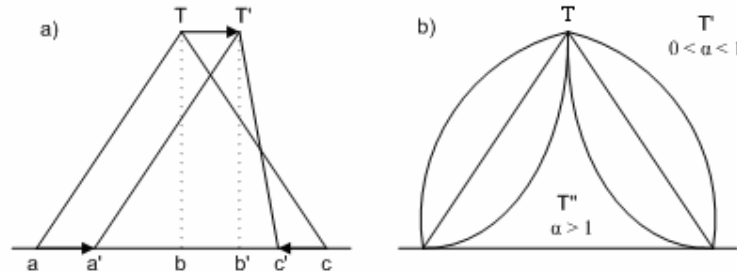
$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, & \text{if } a \leq x < b \\ \frac{c-x}{c-b}, & \text{if } b \leq x \leq c \\ 0, & \text{diğer durumda} \end{cases} \quad (1.1)$$

temel parametreler a , b ve c değiştirilerek yeni bir üyelik fonksiyonu oluşturulabilir ve böylece sistem performansı iyileştirilebilir. Bu yöntemde görüldüğü gibi ayarlanacak parametre sayısı fazladır. Fazla parametre ayarlamak hesap kolaylığını ortadan kaldırması ve uygulamalarda hafızada fazla yer kaplaması bakımından istenmeyen bir durumdur. Bu ayarlama yöntemi Şekil 1.1 (a)' da görülmektedir.

Üyelik fonksiyonlarının ayarlanması için uygulanan diğer bir yöntem ise üyelik fonksiyonunun temel parametrelerini değiştirmeden, daha esnek bir yaklaşımla dilsel biçimlendiricileri kullanarak sadece üyelik derecelerinin değiştirilmesidir. Örneğin sadece pozitif tanımlı bir α parametresi kullanılarak yeni üyelik fonksiyonu

$$\mu'(x) = (\mu(x))^\alpha, \quad \alpha \in R^+ \quad (1.2)$$

şeklinde oluşturulabilir. Bu yöntem Şekil 1.1 (b)' de görülmektedir.



Şekil 1.1: Üyelik fonksiyonlarını ayarlama yöntemleri a) Temel parametreleri değiştirilerek b) Dilsel biçimlendiriciler kullanarak

Bir çok araştırmacı yüksek performans sağlayacak üyelik fonksiyonlarının otomatik olarak seçimi için çeşitli yöntemler kullanarak temel parametreleri değiştirme yoluna gitmiştir;

- Genetik algoritmalar (GA) [7-11]
- Yapay sinir ağları [12-16]
- Diğer metotlar [17-24].

Bazı araştırmacılar ise Zadeh tarafından önerilmiş dilsel biçimlendiriciler [25] üzerine yoğunlaşmışlardır [26-32]. Ayrıca literatürde bu iki yöntemi aynı anda kullanan çalışmalar da mevcuttur [6].

Üyelik fonksiyonlarının ve kuralların otomatik olarak belirlenmesi için bir çok algoritma mevcut olmasına rağmen, kullanılan değişken sayısının artması, kontrol edilen sistemin parametrelerinin değişmesi ya da çevre koşullarının değişmesi gibi durumlarda çeşitli zorluklarla karşılaşmaktadır. Bu yüzden bulanık kontrolör tasarımı basit ve az sayıda üyelik fonksiyonu ve kural yapısı kullanma gereksinimi doğmaktadır.

Bu çalışmada, en basit yapıya sahip bir bulanık kontrolör üzerinde, üyelik fonksiyonlarını dilsel biçimlendiriciler kullanarak çevrim içi olarak ayarlayan yeni bir yöntem önerilmiştir. Bu yöntem sayesinde, her bir giriş değişkeni için sadece basit üçgen yapıda üç üyelik fonksiyonu ve iki giriş değişkeni için toplam dokuz kural kullanılarak oluşturulan basit yapıdaki bir bulanık kontrolörün performansı etkili bir biçimde artırılabilir.

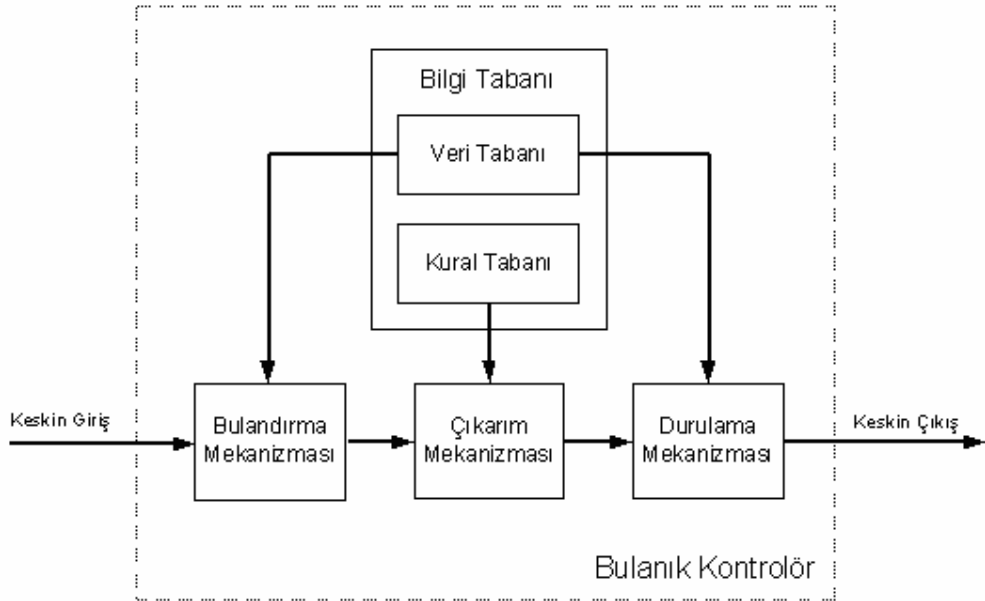
Tezin sonraki bölümleri şöyledir: 2. bölümde bulanık kontrolörün temel yapısı ve bulanık kontrolör çeşitleri anlatılmaktadır. 3. bölümde dilsel biçimlendiricilerden bahsedilmiştir. 4. bölümde önerilen yeni yöntemin yapısı gösterilmektedir. Yeni yöntemin çeşitli sistemler üzerinde denenmesiyle elde edilen simülasyon sonuçları 5. bölümde gösterilmektedir. Ve son olarak 6. bölümde çalışmalardan elde edilen sonuçlar özetlenmektedir.

2. BULANIK KONTROLÖRLER

2.1 Bulanık Kontrolörlerin Temel Yapısı

Bulanık mantık yaklaşımı ile tasarlanan bulanık kontrolörler, insan düşünce tarzını taklit ederek yaklaşık karar verme özelliğine sahip olmaları, kontrol edilecek sistemin matematiksel modeline ihtiyaç duymamaları, dilsel ifadelerle hesaplama yapabilmeleri ve sadece bir çözüm noktasında değil tüm işlem aralığında etkili olabilme özellikleri nedeniyle, klasik kontrolörlere bir alternatif oluşturmakta ve özellikle klasik kontrolörlerin üstesinden gelemediği bir çok karmaşık yapıdaki sistemlerin kontrolünde yaygın olarak kullanılmaktadır.

Genel bir bulanık kontrolör yapısı dört bölümden oluşur; bulandırma mekanizması, bulanık çıkarım mekanizması, bilgi tabanı ve durulama mekanizması. Şekil 2.1' de bulanık kontrolörün temel yapısı gösterilmiştir.



Şekil 2.1: Bulanık kontrolörün temel yapısı

Bulandırma mekanizması, kontrolöre gelen giriş değişkenlerinin kesin değerlerini bulanık değerlere çevirir. SICAK, SOĞUK, ILIK,... gibi dilsel olarak ilişkilendirilmiş uygun bulanık kümelerle çevrilerek bulandırılan kesin giriş değerleri böylece kural tabanındaki dilsel kurallarla yorumlanabilir hale gelmektedir. Bu bulanık kümeler genellikle üçgen şeklinde seçilmekle beraber yamuk ve çan eğrisi şeklinde ya da özel tanımlanmış fonksiyonlar şeklinde de kullanılabilir. Üyelik fonksiyonlarının şekli üyelik derecelerini, üyelik dereceleri de bir kuralın ne ölçüde ateşleneceğini belirleyerek, kural kümesi ve üyelik fonksiyonları arasındaki ilişkiyi belirtir.

Bilgi tabanı kuralların ve üyelik fonksiyonlarının tanımlandığı iki kısımdan oluşur. Veri tabanı giriş ve çıkış üyelik fonksiyonu bilgilerini içerir. Kural tabanı ise kontrol amacını ve stratejisini belirleyen kural kümesini içerir. Kural tabanı “EĞER - O HALDE” şeklinde tanımlanmış kurallarla kontrol edilecek sistemin sözel olarak modellenmiş hali olarak görülebilir. Bulanık kural tabanı oluşturulurken şu üç özellik göz önünde bulundurulmalıdır; ilk olarak kurallar tam olmalıdır, yani olası koşulları atlamamalıdır, ikinci olarak tutarlı olmalıdır, yani çelişkili sonuçlar bulunmamalıdır ve son olarak kurallar kesin olmalıdır, yani fazla ya da eksik olmamalıdır [33].

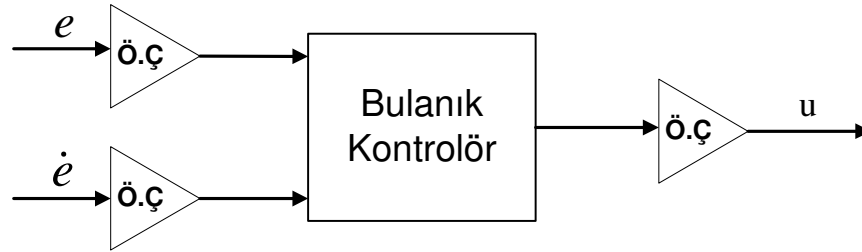
Çıkarım mekanizması bulandırılmış ölçümleri kullanarak o an için hangi kontrol kuralının uygun olduğunu değerlendirir ve kontrol çıkışının ne olması gerektiğine karar verir. Çıkarım mekanizmasının sonucu bir bulanık kümedir.

Durulama birimi çıkarım mekanizmasından gelen bulanık sonucu uygulamada kullanılacak gerçek değere dönüştürür. Durulayıcı mekanizmasının seçiminde şu üç özellik göz önünde bulundurulmalıdır; ilk olarak akla yatkın olmalı, yani durulayıcıdan elde edilen keskin çıkış değeri çıkarım mekanizmasından elde edilen bulanık sonucu sezgisel ve mantıksal açıdan iyi ifade etmelidir, ikinci olarak algoritma kısa sürede hesaplanabilecek kadar basit olmalıdır ve son olarak çıkarım mekanizmasından elde edilen bulanık değerdeki küçük değişimler, elde edilen keskin çıkışta çok büyük değişikliklere sebep olmamalıdır.

2.2 Bulanık Kontrolör Çeşitleri

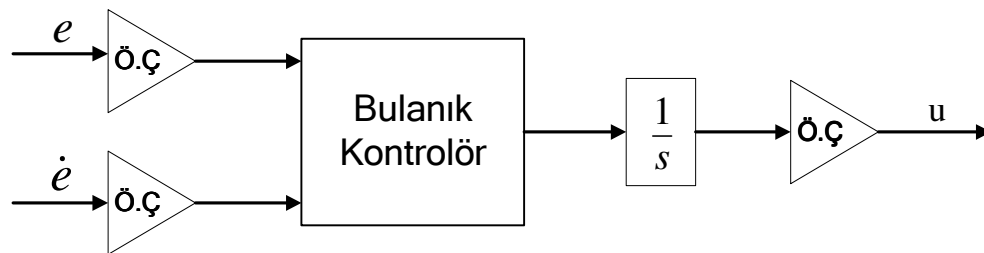
Literatürde çeşitli tiplerde bulanık kontrolörler mevcuttur. Bunlardan biri PD (orantı-türev) tip bulanık kontrolörlerdir. Bu kontrolörler sistemden gelen hata (e) ve hatanın

türevi (Δe) bilgisini kullanarak uygun kontrol işaretini (u) üretirler. Bulanık PD tip kontrolörler sistem yanıtında iyi bir sönüm sağlar ve maksimum aşım, yükselme zamanı ve yerleşme zamanı değerlerini azaltırlar. Ayrıca geri beslemeli sistemlerin kararlılığını arttıırırlar [34]. Fakat bu tip kontrolörlerde kararlı hal hatasını ortadan kaldırmak oldukça zordur. Şekil 2.2' de PD tip bulanık kontrolörün blok diyagramı görölmektedir.



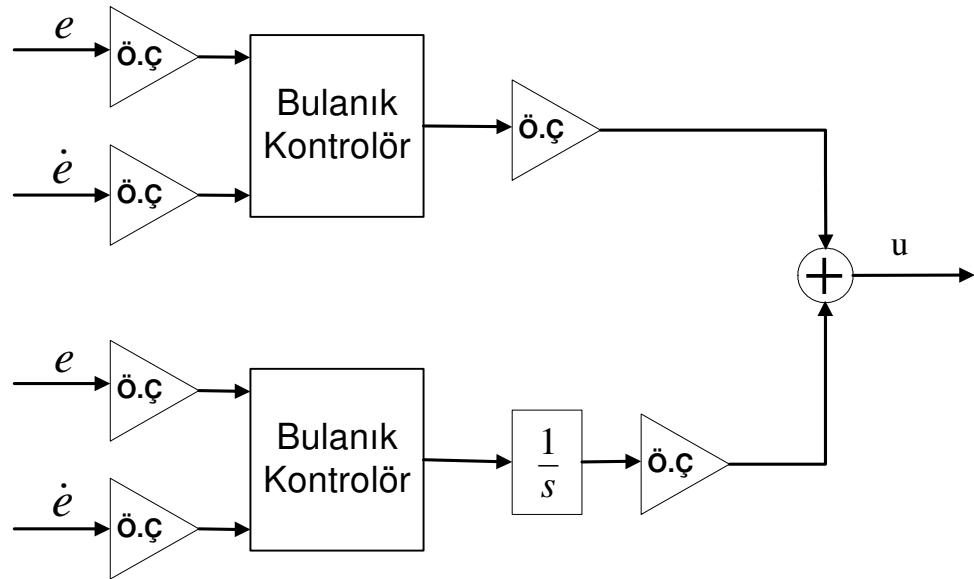
Şekil 2.2: PD tip bulanık kontrolörün blok diyagramı

PD tip bulanık kontrolörlerin kararlı hal hatasına sebep olmalarından ötürü PI (orantı-integral) tip bulanık kontrolörler uygulamalarda daha fazla tercih edilmektedir. Girişte hata ve hatanın integrali bilgisinin kullanılması kural çıkarımı açısından zor olduğundan, bunun yerine girişte hata ve hatanın türevi bilgisi kullanılıp kontrolörün çıkışına bir integratör konularak PI tip bulanık kontrolörler elde edilir. Böylece bulanık PI tip kontrolörler giriş olarak hata ve hatanın türevi bilgisini kullanarak kontrol işaretindeki değişim miktarını (Δu) belirlerler. Bu tip kontrolörlerde kararlı hal hatası oluşmamasına karşın, özellikle sistem derecesi ve sistemdeki ölü zaman değeri arttıkça yapılarındaki integral etkisinden ötürü geçici hal üzerinde etkinlikleri azalmaktadır. Şekil 2.3' de PI tip bulanık kontrolörün blok diyagramı görölmektedir.

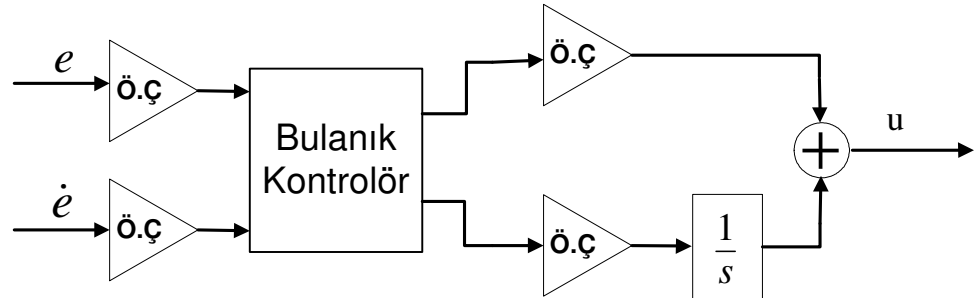


Şekil 2.3: PI tip bulanık kontrolörün blok diyagramı

Bulanık PID (orantı-integral-türev) tip kontrolörler performansı daha çok arttırırlar. Üç boyutlu kural çıkarımı zor olduğundan dolayı genellikle PID tip bulanık kontrolörler PI ve PD tip bulanık kontrolörlerin bir arada kullanılmasıyla oluşturulurlar [35-38]. PI ve PD tip bulanık kontrolörlerin kullanılmasıyla PID tip bulanık kontrolörün elde edilmesi iki yolla olmaktadır. İki ayrı kural tablosu kullanılarak PI ve PD tip bulanık kontrolörlerin çıkışları toplanır ve böylece kontrol işareti üretilir ya da tek kural tablosu kullanılarak PD tip bulanık kontrolörün çıkışına bir integratör ve toplama birimi eklenir ve bu şekilde kontrol işareti üretilir. Her iki yöntemin blok diyagramları Şekil 2.4 ve Şekil 2.5’ de sırasıyla gösterilmiştir.



Şekil 2.4: İki kural tablosu kullanılarak elde edilen PID tip bulanık kontrolör yapısı



Şekil 2.5: Tek kural tablosu kullanılarak elde edilen PID tip bulanık kontrolör yapısı

Bu çalışmada PI tip bulanık kontrolör kullanılmıştır. Önerilen yöntem ile PI tip bulanık kontrolörün yukarıda bahsedilen geçici haldeki kötü performansı ortadan

kaldırılmış ve büyük ölü zamanlı zor sistemlerde bile PI tip bulanık kontrolörle oldukça etkin bir kontrol sağlanmıştır.

3. DİLSEL BİÇİMLENDİRİCİLER

Dilsel biçimlendiriciler bulanık kontrolörün üyelik fonksiyonlarının şeklini değiştirmek için kullanılan operatörlerdir. Bu operatörler sayesinde değişkenler üyelik fonksiyonlarında o anki duruma göre daha düşük ya da daha yüksek kesinlikte tanımlanabilmekte ve böylece daha iyi bir çıkarım elde edilebilmektedir. Dilsel biçimlendiriciler “daraltıcı”, “genişletici” ve “zıt yönlü” dilsel biçimlendiriciler olmak üzere üç sınıfa ayrılabilir [25].

a) **Daraltıcı dilsel biçimlendiriciler:** Daraltıcı dilsel biçimlendiriciler şu şekilde ifade edilir

$$\text{DAR}(x) \triangleq x^\alpha, \quad \alpha > 1. \quad (3.1)$$

Daraltıcı dilsel biçimlendiriciler görüldüğü gibi uygulandıkları üyelik fonksiyonlarının üyelik derecelerini azaltırlar. Bu tür içinde “çok”, “biraz daha çok”, “daha da çok”, “çok daha çok”, ve “tamamen” gibi daraltıcı dilsel biçimlendirici kalıpları tanımlanabilir [26]

$$\begin{aligned} \text{tamamen } x &\triangleq x^4 \\ \text{çok daha çok } x &\triangleq x^2 \\ \text{daha da çok } x &\triangleq x^{1.75} \\ \text{biraz daha çok } x &\triangleq x^{1.5} \\ \text{çok } x &\triangleq x^{1.25}. \end{aligned} \quad (3.2)$$

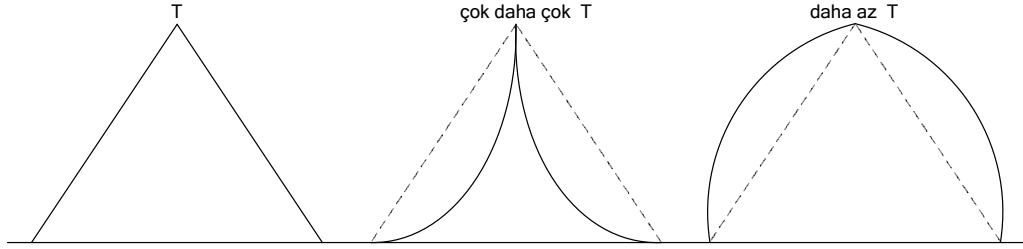
b) **Genişletici dilsel biçimlendiriciler:** Daraltıcı dilsel biçimlendiricilerin aksine genişletici dilsel biçimlendiriciler uygulandıkları üyelik fonksiyonlarının üyelik derecelerini arttıırırlar. Genişletici dilsel biçimlendiriciler Zadeh tarafından şu şekilde tanımlanmaktadır

$$\text{GEN}(x) \triangleq x^\alpha, \quad 0 < \alpha < 1. \quad (3.3)$$

Benzer şekilde genişletici dilsel biçimlendiriciler için de “az”, “daha az” ve “çok daha az” şeklinde kalıplar tanımlanabilir

$$\begin{aligned} az &\triangleq x^{0.75} \\ daha\ az &\triangleq x^{0.5} \\ çok\ daha\ az &\triangleq x^{0.25}. \end{aligned} \quad (3.4)$$

Şekil 3.1’ de daraltıcı dilsel biçimlendiricilerden “çok daha çok” un ve genişletici dilsel biçimlendiricilerden “daha az” ın üyelik fonksiyonları üzerindeki etkisi gösterilmektedir.



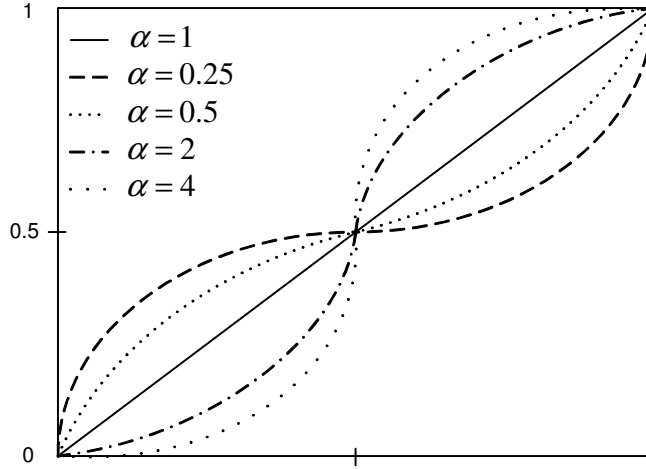
Şekil 3.1: “Çok daha çok” ve “daha az” dilsel biçimlendiricilerin üyelik fonksiyonları üzerindeki etkisi

c) **Zıt yönlü dilsel biçimlendiriciler:** Zıt yönlü dilsel biçimlendiriciler şu şekilde tanımlanmaktadır [32]

$$ZIT(x) \triangleq \begin{cases} (0.5)^{1-\alpha} \mu[x]^\alpha & \mu[x] < 0.5 \text{ için} \\ 1 - (0.5)^{1-\alpha} (1 - \mu[x])^\alpha & \mu[x] \geq 0.5 \text{ için} \end{cases} \quad \alpha > 0. \quad (3.5)$$

İfadeden de görüldüğü gibi eğer α parametresi $0 < \alpha < 1$ şeklinde seçilmişse zıt yönlü dilsel biçimlendiriciler, üyelik derecesinin 0.5 değerinden küçük olduğu durumda arttırıcı yönde, 0.5 değerinden büyük olduğu durumda azaltıcı yönde etki etmektedir. Eğer $\alpha > 1$ seçilmişse bir öncekinin aksine zıt yönlü dilsel biçimlendiriciler, üyelik derecesi 0.5 değerinden küçük olduğu durumda azaltıcı yönde, 0.5 değerinden büyük olduğu durumda arttırıcı yönde etki etmektedir. Zıt yönlü dilse biçimlendiricilerin üyelik fonksiyonları üzerindeki etkisi Şekil 3.2’ de gösterilmiştir. Diğer dilsel biçimlendiriciler için α parametresine sayısal değerler

verilerek yapılmış “az”, “daha az”, “çok”, “daha çok” gibi tanımlamalar burada da yapılabilmektedir.



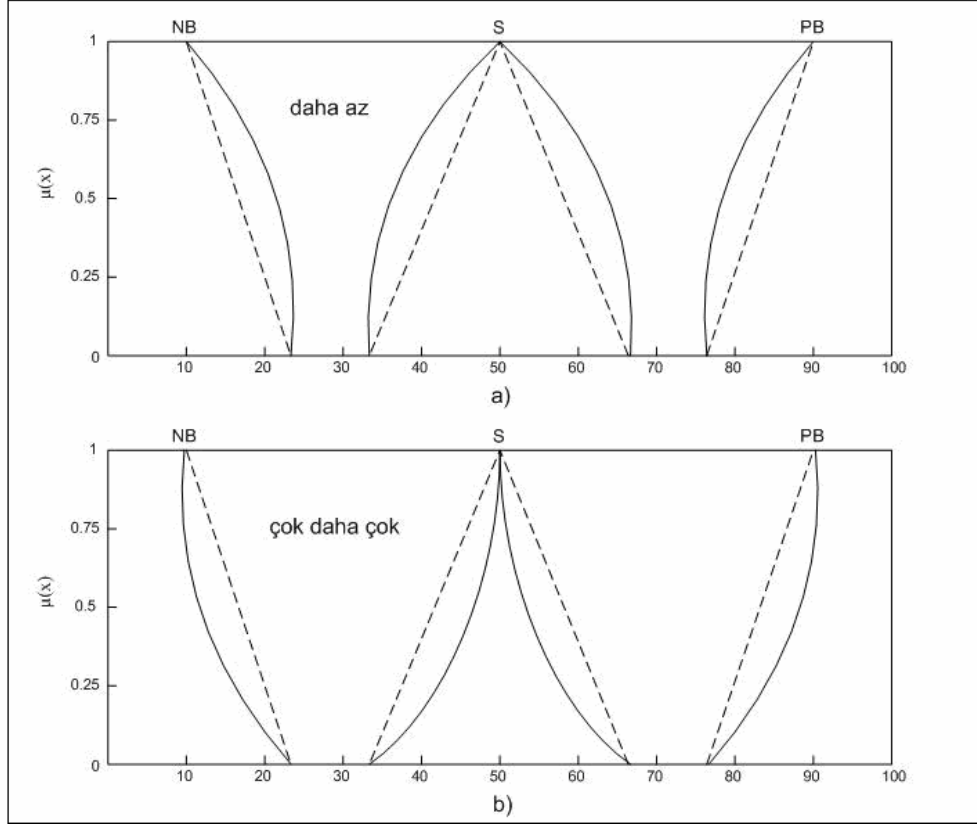
Şekil 3.2: Zıt yönlü dilsel biçimlendiricilerin üyelik fonksiyonu üzerindeki etkisi

Yukarıdaki ifadelerden kolayca görülebileceği gibi bütün h dilsel biçimlendiricileri aşağıdaki koşulları sağlamaktadır [39];

1. $h(0) = 0$ ve $h(1) = 1$.
2. h sürekli bir fonksiyondur.
3. Eğer h genişletici bir dilsel biçimlendirici ise h^{-1} daraltıcı bir dilsel biçimlendiricidir ve bunun tam aksi de doğrudur.
4. Verilen başka bir g dilsel biçimlendiricisinin h ile veya h dilsel biçimlendiricisinin g ile birlikte kullanılması yine bir dilsel biçimlendiricidir.

Bu çalışmada sadece daraltıcı ve genişletici dilsel biçimlendiriciler kullanılmıştır. Ayrıca çalışmada α parametresi için “çok daha çok” ($\alpha = 2$), “az” ($\alpha = 0.75$), “daha az” ($\alpha = 0.5$) gibi sayısal değerlerle belirlenmiş tanımları kullanmak yerine, daha hassas bir kontrol sağlamak için $(0, 4]$ aralığındaki tüm değerler kullanılmıştır.

Dilsel biçimlendiricilerin üyelik fonksiyonları üzerindeki etkisini basitçe anlatmak için bir giriş değişkeni için tanımlanmış NB (negatif büyük), S (sıfır) ve PB (pozitif büyük) üyelik fonksiyonlarını göz önüne alalım. Bu üyelik fonksiyonlarının yapısı Şekil 3.3’ de kesik çizgiler ile gösterilmiştir.



Şekil 3.3: NB, S ve PB üyelik fonksiyonları üzerinde a) “daha az” dilsel biçimlendiricisinin etkisi b) “çok daha çok” dilsel biçimlendiricisinin etkisi

Şekil 3.3(a)’ da genişletici dilsel biçimlendiricilerden “daha az” uygulanması sonucu oluşan yeni üyelik fonksiyonları görülmektedir. NB ve PB üyelik fonksiyonları üzerinde bakıldığında görüldüğü gibi “daha az” dilsel biçimlendiricisi üyelik derecelerini arttırmakta ve bulanık kümenin anlamını güçlendirmektedir. Bu durumda giriş değeri, bulanık kontrolör tarafından S hedef değerine normalde olduğundan daha da uzakta algılanmakta ve bulanık kontrolör hedefe daha çabuk ulaşılması için kontrol çıkışı güçlendirmektedir. S üyelik fonksiyonu üzerindeki etkiye bakıldığında, yine anlam güçlenmekte ve bu sefer de giriş değeri bulanık kontrolör tarafından S hedef değerine daha da yakın olarak algılanmakta ve böylece kontrol işareti daha hassas bir şekilde ayarlanmaktadır.

Diğer taraftan Şekil 3.3(b)’ de “çok daha çok” dilsel biçimlendiricinin uygulanması sonucu elde edilen yeni üyelik fonksiyonları görülmektedir. Bu dilsel biçimlendirici üyelik derecelerini azaltmakta ve bulanık kümenin anlamını zayıflatmaktadır. NB ve PB üyelik fonksiyonları üzerinde bakıldığında giriş değeri bulanık kontrolör

tarafından S hedef deęerine normalde olduęundan daha da yakında algılanmakta ve böylece kontrol işareti zayıflatılmaktadır. S üyelik fonksiyonu üzerinde bakıldığında ise yine üyelik derecesi azaldığından giriş deęeri bulanık kontrolör tarafından S hedef deęerine normalde olduęundan daha da uzakta algılanmakta ve böylece kontrol işareti olması gerektiğinden daha kaba bir şekilde ayarlanmaktadır.

Bulanık kontrolör tasarımında uygun üyelik fonksiyonlarının seçimi oldukça önemlidir. Hesap kolaylığını arttırmak ve hafızada kullanılan yeri azaltmak için az sayıda ve basit yapıda üyelik fonksiyonlarının seçilmesi gerekmektedir. Ayrıca seçilen üyelik fonksiyonları üzerinde temel parametreleri deęiştirmek yerine dilsel biçimlendiricileri kullanarak ayarlama yapmak daha basit ve mantıklıdır. Çünkü bu durumda çok sayıda temel parametre ayarlamak yerine her bir üyelik fonksiyonu için sadece bir parametre ayarlanmaktadır.

Bu çalışmada bu mantıktan yola çıkılarak bulanık kontrolörde her bir giriş deęişkeni için basit üçgen yapıda üç üyelik fonksiyonu kullanılmaktadır. Üyelik fonksiyonlarının ayarlanması için ise dilsel biçimlendiriciler kullanılmış ve tüm üyelik fonksiyonlarının ayarlanması yerine çıkarımda baskın olan hata PB ve hatanın türevi NB üyelik fonksiyonlarının ayarlanması ile toplamda sadece iki parametre ayarlanarak oldukça etkili bir kontrol sağlanmaktadır.

4. ÖNERİLEN YÖNTEMİN YAPISI

Değişkenlerin üyelik fonksiyonları üzerinde sahip oldukları üyelik derecesi kuralların hangi ölçüde ateşleneceğini belirlemektedir. Bu yüzden üyelik fonksiyonlarının seçimi ve ayarlanması bulanık kontrolörün performansı üzerinde önemli bir etkiye sahiptir. Daha önce de bahsedildiği gibi hesap kolaylığı ve hafızada ayrılan yer göz önünde bulundurularak, kontrolör tasarımı yapılırken fazla ve karmaşık yapıda üyelik fonksiyonları kullanmaktan kaçınılır. Ayrıca üyelik fonksiyonlarının ayarlanması sırasında temel parametrelerin değiştirilmesi yerine dilsel biçimlendiricilerin kullanılması daha az sayıda parametre belirlemek açısından daha uygundur. Bu yüzden bu çalışmada her bir giriş değişkeni için basit üçgen yapıda üç üyelik fonksiyonu ve bu üyelik fonksiyonlarının ayarlanması için de dilsel biçimlendiriciler kullanılmıştır.

Bugüne kadar literatürde üyelik fonksiyonlarının ayarlanması için yapılan çalışmalarda, dilsel biçimlendirici veya temel parametre değerleri çevrim dışı olarak belirlenmiş ve bulunan optimal ya da optimale yakın bu değerler kontrolörün yapısına işlenerek sabit bir üyelik fonksiyonu yapısı kullanılmıştır. Bu çalışmada ise süreç boyunca kontrolöre çevrim dışı olarak belirlenmiş sabit bir dilsel biçimlendirici değeri atamak yerine, her örnekleme adımında sistem davranışına göre değişen uygun dilsel biçimlendiricileri atayan yeni bir yöntem önerilmiştir. Bu işlemi gerçekleştirmek için bir çıkarım mekanizması kullanılmaktadır. Çıkarım mekanizması süreç boyunca hata ve hatanın türevinden bilgi alarak üyelik fonksiyonlarına uygulanacak uygun dilsel biçimlendirici değerlerini belirlemektedir. Bu yöntem sayesinde bulanık kontrolörün iç yapısına çevrim içi olarak müdahale edilerek, karmaşık yapıya ya da büyük ölü zamanlı zor sistemlerde bile etkin bir kontrol sağlanabilmektedir. Bu yöntem uygulamalarda sıklıkla kullanılan PI tipi bulanık kontrolör üzerinde gerçekleştirilmiştir.

4.1 Ayarlama Mekanizmasının Yapısı

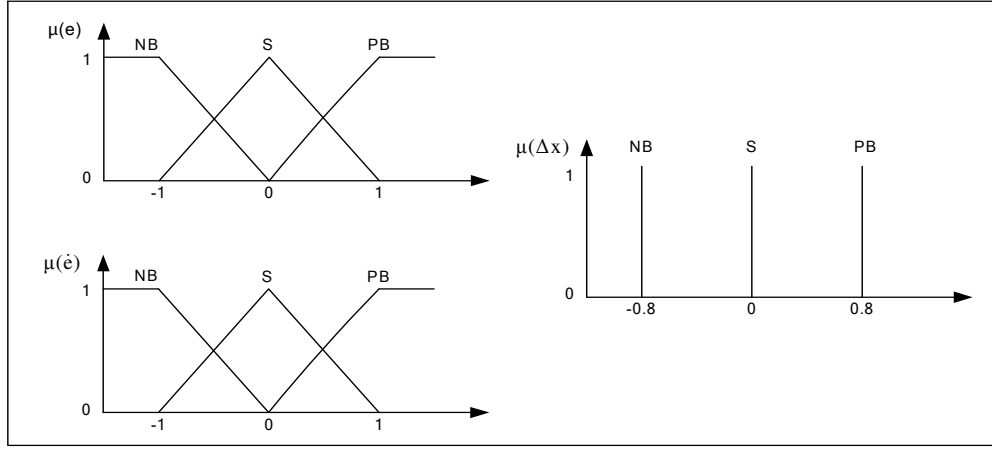
PI tipi bulanık kontrolörler kararlı hal hatasını ortadan kaldırdıklarından ötürü sıklıkla kullanılmalarına karşın bu kontrolörlerin en büyük dezavantajı sistem derecesi ve sistemdeki ölü zaman arttıkça yapılarındaki integral etkisi yüzünden geçici halde etkili bir kontrol sağlayamamalarıdır. Bu yüzden PI tipi bulanık kontrolörler genellikle düşük dereceden ve ölü zamanı az olan sistemlerin kontrolünde kullanılmaktadır.

Bu çalışmada önerilen yeni ayarlama yöntemin ana amacı, hatanın büyük olduğu kontrolün ilk evrelerinde kontrol işaretini hızlı bir şekilde arttırarak sistem yanıtında hızlı bir yükselme sağlamak ve referans değere yaklaşıldığında kontrol işaretini ani olarak azaltarak sistem yanıtının aşırı bir şekilde referans değere yerleşmesini sağlamaktır. Böylece PI tipi bir bulanık kontrolörle geçici halde de etkin olan hızlı bir kontrol elde etmektir.

Genel anlamda hızlı bir sistem yanıtı elde etmek için kontrol işaretinin aynı yönde kuvvetlendirilmesi istendiğinden, çıkarım mekanizmasında PI tipi bulanık kontrolörde kullanılan aynı kurallar ve kural yapısı kullanılmıştır. Kullanılan kural tablosu Tablo 4.1’ de gösterilmektedir. Basitlik açısından giriş ve çıkış üyelik fonksiyonları da Şekil 4.1’ de görüldüğü gibi doğrusal sistem uygulamasında kullanılan üyelik fonksiyonları ile aynı seçilmiştir. Ayrıca çıkarım mekanizmasının girişi olan hata ve hatanın türevine yeni ölçekleme çarpanları kullanmak yerine parametre sayısını azaltmak için çıkarım mekanizmasında bulanık kontrolörün ölçeklenmiş girişleri kullanılmıştır. Çıkarım için Takagi-Sugeno yöntemi seçilmiş ve durulama mekanizması için de yine ağırlık merkezi yöntemi kullanılmıştır.

Tablo 4.1: Çıkarım Mekanizmasında Kullanılan Kural Tablosu

$\Delta e/e$	NB	S	PB
NB	NB	NB	S
S	NB	S	PB
PB	S	PB	PB



Şekil 4.1: Çıkarım mekanizmasında kullanılan üyelik fonksiyonları

Çıkarım mekanizması, referans dilsel biçimlendirici değeri olan 1' e arttırım ve azaltımda bulunarak üyelik fonksiyonlarına uygun dilsel biçimlendirici değerlerinin atanmasını sağlayacak λ değerini belirlemektedir. Üyelik fonksiyonlarına atanacak dilsel biçimlendirici değeri α şu şekilde belirlenmektedir

$$\alpha_i = 1 + k_{ij} \cdot \lambda \quad i = 1, 2, \dots, j = 1, 2. \quad (4.1)$$

Burada λ çıkarım mekanizması tarafından belirlenen çıkış değerini, k dilsel biçimlendirici atanacak üyelik fonksiyonu için kullanılan çıkış ölçekleme çarpanını, i dilsel biçimlendirici atanacak üyelik fonksiyonunu ve j de anahtarlama kademesini belirtmektedir.

Kontrol boyunca çıkarım mekanizmasında bir anahtarlama yardımıyla dilsel biçimlendirici atanacak her bir üyelik fonksiyonu için iki farklı k çıkış ölçekleme çarpanı kullanılmıştır. Anahtarlamanın birinci kademesinde bulunan ölçekleme çarpanları ile üyelik fonksiyonlarına kontrol işaretini arttırıcı yönde uygun dilsel biçimlendirici değeri uygulanmakta ve ikinci kademe de bulunan ölçekleme çarpanlarıyla ise üyelik fonksiyonlarına kontrol işaretini azaltıcı yönde uygun dilsel biçimlendirici değeri uygulanmaktadır. Ayrıca $\alpha < 0$ olmasını engellemek için $k_{ij} \cdot \lambda$ değerine -0.9 değerinde saturasyon uygulanmaktadır.

Anahtarlama kuralı, ölü zamanlı sistemler için ölü zamanın bitiminde, ölü zamansız sistemler için ise ölçeklenmiş hata değeri, başlangıç değerinin yani kullanılan

ölçekleme çarpanı değerinin %15' ine ulaştığı anda gerçekleşecek şekilde belirlenmiştir.

Kontrol işaretinin artırılıp azaltılması için üyelik fonksiyonları üzerinde hangi tip dilsel biçimlendiricilerin atanması gerektiğini görmek için hata PB ve hatanın türevi NB üyelik fonksiyonlarını göz önüne alalım. Kural tablosundan ve durulama yönteminden anlaşılacağı gibi kontrol işaretini arttırmak için o anki hata PB üyelik derecesini artırmak ve hatanın türevi NB üyelik derecesini azaltmak gerekir. Kontrol işaretini azaltmak için ise hata PB üyelik derecesinin azaltılması ve hatanın türevi NB üyelik derecesi artırılması gerekir. Bu yüzden kontrol işaretinin hızlı bir şekilde artırılması istenen ilk kademede hata PB üyelik fonksiyonuna genişletici tip dilsel biçimlendirici ve hatanın türevi NB üyelik fonksiyonuna daraltıcı tip dilsel biçimlendirici atanması gerekmektedir. Kontrol işaretinin azaltılması istenen ikinci kademede ise hata PB üyelik fonksiyonuna daraltıcı tip dilsel biçimlendirici ve hatanın türevi NB üyelik fonksiyonuna genişletici tip dilsel biçimlendirici atanması gerekmektedir.

Kontrolün ilk evrelerinde ve büyük bölümünde λ değeri pozitif bir değer olduğundan, (4.1)' den kolayca görülebildiği gibi genişletici tip ($0 < \alpha < 1$) dilsel biçimlendirici elde etmek için k parametresinin işareti negatif, daraltıcı tip ($\alpha > 1$) dilsel biçimlendirici elde etmek için ise k parametresinin işareti pozitif olması gerekir.

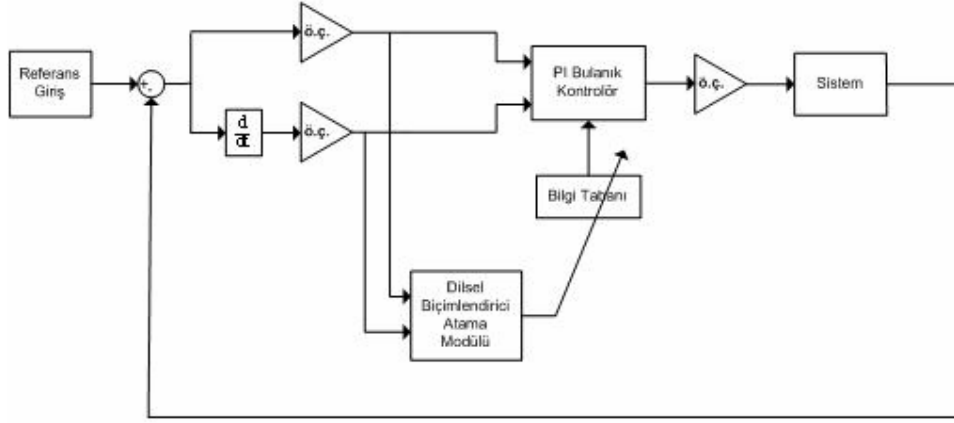
Yöntem geliştirilirken ilk olarak literatürde uygulanan çevrim dışı olarak sabit dilsel biçimlendirici atama yöntemi üzerinde incelemeler yapılmış ve Liu [26]' nun önerdiği bölgesel dilsel biçimlendirici atama yöntemi doğrusal olmayan ve ölü zamanlı doğrusal sistemler üzerinde denenmiştir. İncelemeler sonucu dilsel biçimlendiricilerin sistem performansı üzerindeki iyileştirici etkileri açıkça görülmüştür. Ayrıca doğrusal sistemler üzerinde Liu' nun önerdiği yöntemin bir geçerliliği olmadığı ve en iyi cevabın tek bölgesel dilsel biçimlendirici değeri atamakla elde edildiği ispatlanmıştır. Bunun nedeni bölge sayısı arttıkça uygulanan dilsel biçimlendiricilerin üyelik fonksiyonu üzerindeki etkisinin azalması ve üyelik fonksiyonu şeklinin giderek ilk halindeki üçgen duruma yaklaşmasıdır. Doğrusal olmayan sistemlerde ise doğrusal sistemlerin aksine yapılarındaki düzensizlikten ötürü bölgesel dilsel biçimlendirici atamanın iyileştirici bir etkisi olduğu ve

kullanılan sistem için üyelik fonksiyonları üzerinde dört bölge dilsele biçimlendirici atanması durumunda en iyi sistem yanıtının elde edildiđi görölmüştür. Fakat burada karşımıza çıkan sorun her bir doğrusal olmayan sistem için en uygun bölge sayısının nasıl bilineceğidir. Deneme yanılma süreci zaman alıcı ve masraflı olabilir.

Elde edilen sonuçlar ışığında, süreç boyunca kontrolöre çevrim dışı olarak belirlenmiş sabit bir dilsele biçimlendirici değeri atamak yerine, her örnekleme adımında sistem davranışına göre değışen uygun dilsele biçimlendiricileri çevrim içi olarak atayan yeni bir yöntem geliştirme fikri ortaya çıkmıştır.

Yeni yöntem için yapılan ilk denemelerde dilsele biçimlendiricilerin atanması sırasında üyelik fonksiyonları üzerinde simetri korunmuş, böylece her bir giriş değışkeni üzerinde PB ve NB üyelik fonksiyonlarına ortak bir dilsele biçimlendirici ve S üyelik fonksiyonuna bir dilsele biçimlendirici olmak üzere iki giriş değışkeni için toplam dört dilsele biçimlendirici değeri belirlenmiştir. Bu durumda oldukça etkili sonuçlar elde edilmiş ve savunulan düşünce simölasyon sonuçları ile ispatlanmıştır. Fakat bu durumda iki kademeli anahtarlama mekanizmasından ötürü k parametresi için toplam sekiz değeri belirlenmesi gerekmektedir. Bu yüzden parametre azaltma çalışmaları yapılmıştır. Bunun için yine simetri korunarak sadece, kontrol sürecinin büyük bölümünde baskın olan hata PB ve hatanın türevi NB üyelik fonksiyonlarına dilsele biçimlendirici atama yoluna gidilmiştir. Bu durumda anahtarlama mekanizmasından ötürü toplam dört k parametresi belirlenmektedir. Parametre sayısını daha da azaltmak için hata PB üyelik fonksiyonu için her iki anahtarlama kademesinde de işaretleri farklı tek k değeri kullanılmıştır. Ölü zaman boyunca hatanın değışimi sıfır olduğundan ölü zamanlı sistemlerde hatanın türevi NB üyelik fonksiyonu için anahtarlamanın ilk kademesinde dilsele biçimlendirici değeri atamanın bir etkisi olmamaktadır ($0^a = 0$). Bu yüzden k parametresi sadece ölü zamanın bitiminde anahtarlamanın gerçekleşmesi ile aktif olduğundan, ölü zamanlı sistemlerde hatanın türevi NB üyelik fonksiyonu için anahtarlamaya gerek kalmadan tek bir k parametresi kullanılmıştır. Ölü zamansız sistemlerde ise yine parametre sayısını düşürmek için ilk kademedeki $k = 0$, yani etkisiz seçilmiş ve böylece yine sadece ikinci kademedeki aktif olan bir k parametresi kullanılmıştır. Sadece iki parametre belirlenen bu durumda yapılan simölasyon sonuçları, sekiz parametre belirlendiđi ilk durumda elde edilen sonuçlarla karşılaştırıldığında sonucun

neredeys e aynı olduđu g r lm şt r. B ylece son halini alan bu yeni y ntemle sadece iki parametre belirlenerek kontrol r n performansı  nemli bir  l  de iyileştirilebilmektedir.  nerilen y ntemin blok diyagramı Őekil 4.2' de g sterilmiřtir.



Őekil 4.2:  nerilen y ntemin blok diyagramı

5. SİMÜLASYON SONUÇLARI

Bu bölümde önerilen yöntem doğrusal olmayan ve ölü zamanlı doğrusal iki sistem üzerinde uygulanmış ve simülasyon sonuçlarıyla geçerliliği ispatlanmıştır. İlk olarak dilsel biçimlendiricilerin etkilerini araştırmak için çevrim dışı yöntemlerle yapılan ön çalışmalar gösterilmiştir. Daha sonra elde edilen sonuçlar ışığında önerilen yeni yöntemde kullanılan parametre sayısının düşürülme çalışmaları gösterilmiş ve son olarak da iki parametreye kadar düşürülerek son halini alan bu yeni yöntem zaman sabitinin beş katı büyüklüğünde ölü zaman değerine sahip

$$H(s) = \frac{1}{s+1} e^{-5s} \quad (5.1)$$

şeklinde transfer fonksiyonlu doğrusal bir sistem ve

$$\begin{aligned} y(k+1) = & 0.2y^2(k) + 0.2y(k-1) \\ & + 0.4 \sin[0.5(y(k) + y(k-1))] \\ & \times \cos[0.5(y(k) + y(k-1))] + 1.2x(k) \end{aligned} \quad (5.2)$$

modeliyle verilen doğrusal olmayan bir sistem üzerinde uygulanmıştır. Burada x sisteme uygulanan giriş değerini ve y de sistem çıkışını ifade etmektedir. Tüm simülasyon çalışmaları Matlab Simulink simülasyon paket programı ile gerçekleştirilmiştir. Simülasyonlarda doğrusal sistem için örnekleme zamanı 0.2 saniye, doğrusal olmayan sistem için ise 0.1 saniye seçilmiş ve grafikler örnekleme sayıları ile çizdirilmiştir. Ayrıca tüm parametreler genetik algoritma (GA) kullanılarak belirlenmiş ve bunun için Matlab GA araç kutusu kullanılmıştır. Simülasyonlarda yerleşme zamanı kriteri sistem yanıtının referans değerin $\pm\%1$ ' lik dilimine girmesi olarak belirlenmiştir. Parametrelerin aratılması sırasında amaç ölçütü olarak doğrusal sistemlerde

$$f(h) = \frac{1000}{100mp+6tp+3ts+100ess} \quad (5.3)$$

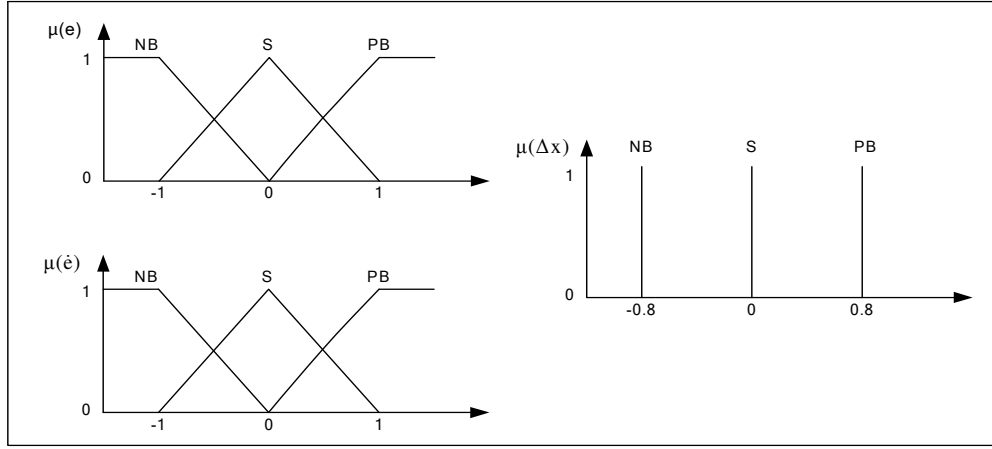
kullanılmıştır. Burada *mp*, *tp*, *ts* ve *ess* sırasıyla aşım, tepe zamanı, yerleşme zamanı ve kararlı hal hatası değerlerini belirtmektedir. Önlerindeki katsayılar ise her bir değer indeks üzerindeki ağırlıklarının eşit olması için bu şekilde belirlenmiştir. Doğrusal olmayan sistemlerde ise Liu' un [26] kullandığı

$$f(h) = \exp\left(-0.2 \cdot \sum_{i=1}^m e_i^2(h)\right) \quad (5.4)$$

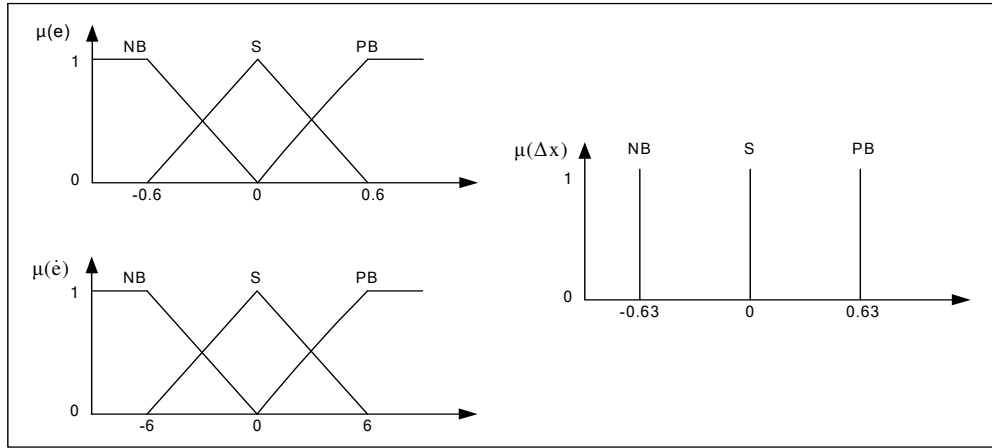
amaç ölçütü kullanılmıştır. Burada *e* hata değerini ve *m* de örnekleme sayısını göstermektedir. Elde edilen sonuçlar, Liu' nun önerdiği üyelik fonksiyonları üzerinde bölgesel olarak dilsel biçimlendirici değeri atama yöntemi ile karşılaştırılmıştır.

5.1 Uygulamalarda Kullanılan PI Tipi Bulanık Kontrolörün Yapısı

Doğrusal sistem uygulamaları için bulanık kontrolörün tasarımında tüm üyelik fonksiyonları için [-1, 1] aralığında normalize edilmiş tanım uzayı kullanılmıştır. Giriş ve çıkış değişkenleri için bu aralığa eşit olarak dağıtılmış NB, S ve PB olmak üzere üç bulanık küme belirlenmiştir. Hesap kolaylığı açısından Takagi-Sugeno tipi bir kontrolör yapısı seçilmiş ve giriş değişkenleri hata ve hatanın türevi için basit yapıda üçgen tip üyelik fonksiyonları kullanılmıştır. Doğrusal olmayan sistem uygulaması için ise seçilen üyelik fonksiyonları yine aynı tip ve sayıda olup sadece normalize edilmiş tanım uzayları farklıdır. Şekil 5.1 ve Şekil 5.2' de her iki uygulama için kontrolörde kullanılan üyelik fonksiyonları görülmektedir.



Şekil 5.1: Doğrusal sistem uygulaması için PI tipi bulanık kontrolörde kullanılan üyelik fonksiyonları



Şekil 5.2: Doğrusal olmayan sistem uygulaması için PI tipi bulanık kontrolörde kullanılan üyelik fonksiyonları

Kontrolörde kullanılan kural tablosu ise Tablo 5.1' den görüldüğü gibi basit ve en genel simetrik yapıda olup dokuz kural içermektedir.

Tablo 5.1: Bulanık Kontrolörde Kullanılan Kural Tablosu

$\Delta e/e$	NB	S	PB
NB	NB	NB	S
S	NB	S	PB
PB	S	PB	PB

Bulanık kontrolörde durulama yöntemi olarak uygulamalarda sıklıkla kullanılan ağırlık merkezi yöntemi seçilmiştir. Bu yöntemde keskin çıkış z_0

$$z_0 = \frac{\sum_{i=1}^n \mu_z(\omega_i) \cdot \omega_i}{\sum_{i=1}^n \mu_z(\omega_i)} \quad (5.5)$$

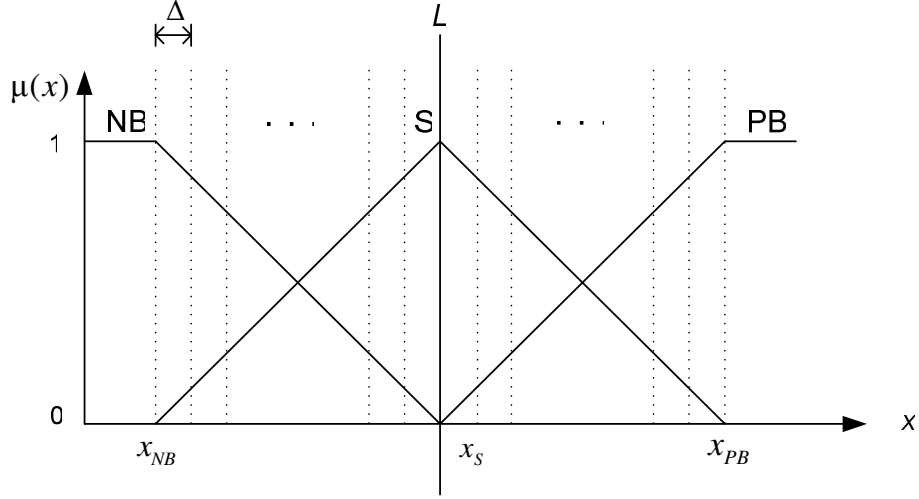
şeklinde elde edilir. Burada $\mu_z(\omega_i)$ bulanık çıkarım değeri, ω_i bu çıkarıma denk düşen tekil tip çıkış üyelik fonksiyonu değeri ve n çıktının niceleme sayısıdır.

Görüldüğü gibi bu yöntem en basit yapıda bir bulanık kontrolöre uygulanmakta ve böylece uygun üyelik fonksiyonları ve kural seçimi gibi özen ve zaman gerektiren tasarım aşamalarına duyulan ihtiyaç ortadan kaldırılmaktadır.

5.2 Çevrim Dışı Yöntemlerle Dilsel Biçimlendiricilerin İncelenmesi

Bu bölümde literatürde şimdiye kadar kullanılan üyelik fonksiyonları için sabit bir dilsel biçimlendirici değeri belirleme ve süreç boyunca bu sabit değeri kullanma yöntemi irdelenerek Liu' nun önerdiği yeni yaklaşım üzerinde durulmuştur.

Liu giriş değişkenleri için belirlenmiş NB, S ve PB şeklinde üç üyelik fonksiyonu üzerinde bölgesel dilsel biçimlendirici atamanın sistem performansını arttıracaklarını savunmuştur. Şekil 5.3' de görüldüğü gibi üyelik fonksiyonlarını belli aralıklarla bölmüş (çalışmasında sekiz aralık kullanma yoluna gitmiştir) ve L simetri eksenini olmak üzere birbirine denk düşen her bir aralık için bir dilsel biçimlendirici değeri belirlemiştir. Bu durumda NB ve PB üyelik fonksiyonları için ortak sekiz parametre ve S üyelik fonksiyonunun her iki bacağı için ortak sekiz parametre olmak üzere toplam on altı parametre aratmıştır. Ayrıca bulduğu her dilsel biçimlendirici değerini her iki giriş değişkeni için de ortak olarak kullanmıştır. Bu yöntemi doğrusal olmayan sistemler üzerinde denemiş ve etkili sonuçlar elde etmiştir.



Şekil 5.3: Bölgesel dilsel biçimlendirici belirleme yönteminin üyelik fonksiyonları üzerinde gösterimi

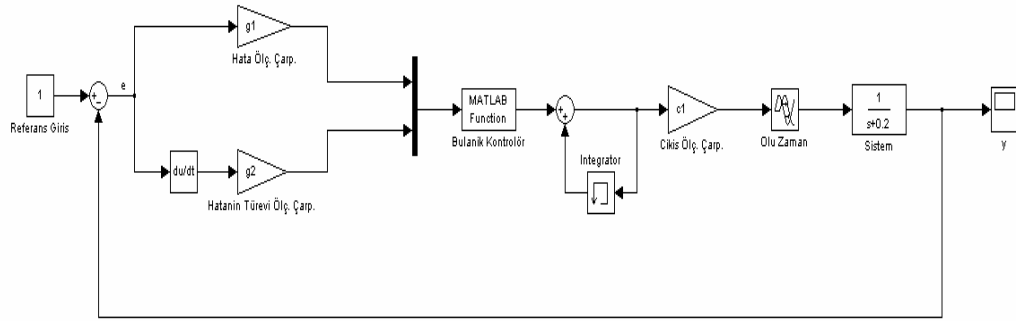
Çevrim dışı yöntemlerle dilsel biçimlendiricilerin etkilerini araştırmak için yapılan çalışmalar, doğrusal olmayan ve ölü zamanlı doğrusal sistemler üzerinde gerçekleştirilmiştir.

5.2.1 Zaman Sabitinin İki Katı Büyüklüğünde Ölü Zaman Değerine Sahip Doğrusal Sistem Üzerinde Yapılan Çalışmalar

Bu bölümde yapılan çalışmalar

$$H(s) = \frac{1}{s + 0.2} e^{-0.4s} \quad (5.6)$$

transfer fonksiyonuna ve zaman sabitinin iki katı büyüklüğünde ölü zaman değerine sahip doğrusal bir sistem üzerinde gerçekleştirilmiştir. Sistemin Simulink blok diyagramı Şekil 5.4' de gösterilmiştir.



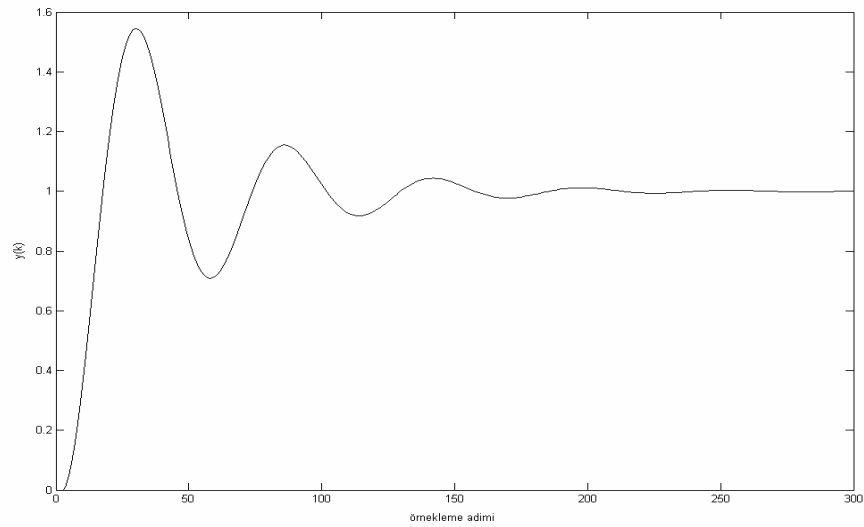
Şekil 5.4: Çevrim dışı yöntemlerle dilsel biçimlendiricilerin incelendiği ölü zamanlı doğrusal sistemin Simulink blok diyagramı

Dilsel biçimlendiricilerin sistem performansı üzerindeki iyileştirme etkilerinin ne ölçüde olduğunu görebilmek için ilk olarak bulanık kontrolörün ölçekleme çarpanları, düzgün olmayan bir sistem yanıtı elde edilecek şekilde seçilmiştir. Bu durum için seçilen ölçekleme çarpanları Tablo 5.2’ de gösterilmiştir.

Tablo 5.2: Doğrusal Sistem İçin Düzgün Seçilmemiş Ölçekleme Çarpanları

Giriş Ölçekleme Çarpanları	Çıkış Ölçekleme Çarpanı
$g1 = 0.8$ $g2 = 0.3$	$c1 = 0.1$

Bu ölçekleme çarpanlarının seçilmesi ile elde edilen sistem yanıtı Şekil 5.5’ de gösterilmiştir.



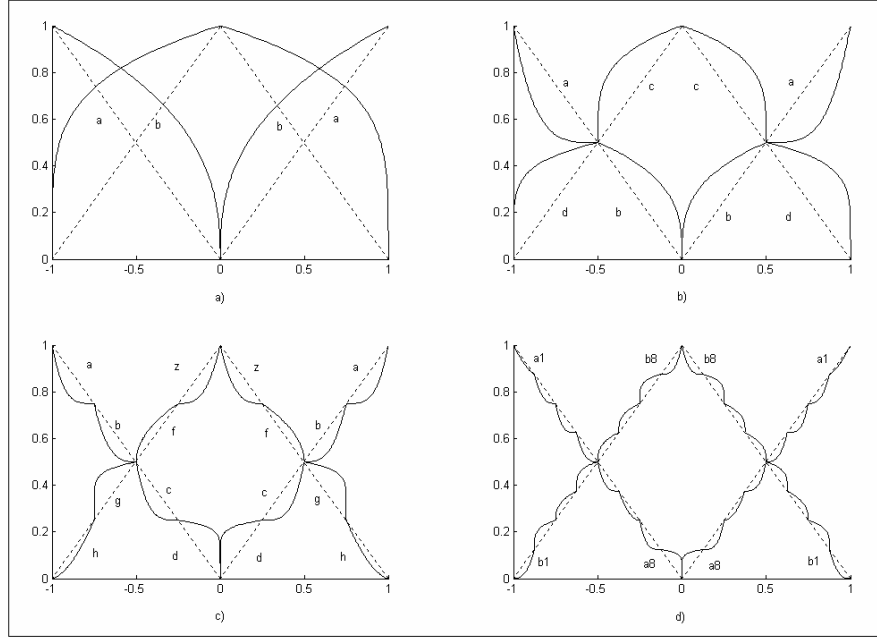
Şekil 5.5: Doğrusal sistem için düzgün seçilmemiş ölçekleme çarpanları ile elde edilen sistem yanıtı

Bu bölümde Liu' nun önerdiği bölgesel dilsel biçimlendirici atama yönteminin doğrusal sistemler üzerindeki geçerliliği araştırılmıştır. İlk olarak üyelik fonksiyonları çeşitli sayılarda bölgelere ayrılmış ve her iki giriş değişkeni için de ortak dilsel biçimlendirici değerleri kullanılmıştır. GA ile her bir bölge sayısı için elde edilen dilsel biçimlendirici değerleri Tablo 5.3' de gösterilmiştir.

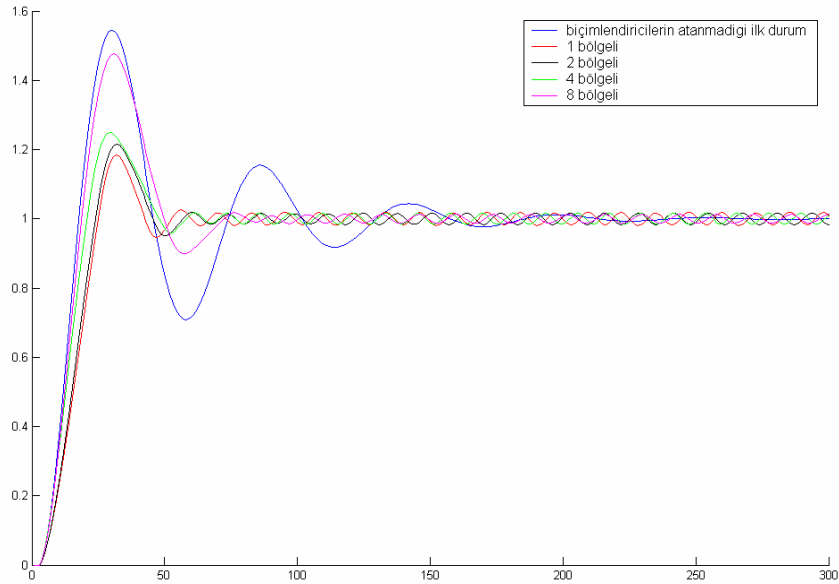
Tablo 5.3: Doğrusal Sistem İçin Ölçekleme Çarpanlarının Düzgün Seçilmemesi Durumunda Belirlenen Ortak Dilsel Biçimlendirici Değerleri

Bölge sayısı	Dilsel biçimlendirici değeri	Amaç ölçütü değeri
Ayarlamasız ilk durum	-	4.7227
Tek bölge	a = 0.3829, b = 0.2231	18.8034
İki bölge	a = 3.9692, b = 0.3182 c = 0.2604, d = 0.1714	17.7562
Dört bölge	a = 3.8539, b = 3.1082 c = 3.7568, d = 0.1038 z = 3.1379, f = 0.5070 g = 0.1367, h = 1.6327	7.5724
Sekiz bölge	a1 = 1.4481, a2 = 2.9206 a3 = 3.6170, a4 = 3.0181 a5 = 3.3439, a6 = 2.0136 a7 = 3.7044, a8 = 0.1118 b1 = 2.7212, b2 = 0.2372 b3 = 0.3769, b4 = 0.2354 b5 = 0.3543, b6 = 0.3318 b7 = 0.1997, b8 = 3.3780	6.4516

Bu dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen yeni üyelik fonksiyonları ve sistem yanıtları sırasıyla Şekil 5.6 ve Şekil 5.7' de gösterilmiştir.



Şekil 5.6: Doğrusal sistem için belirlenen ortak dilsel biçimlendirici değerleri ile elde edilen yeni üyelik fonksiyonları a) Tek bölgeli durum b) İki bölgeli durum c) Dört bölgeli durum d) Sekiz bölgeli durum



Şekil 5.7: Doğrusal sistem için belirlenen ortak dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları

Görüldüğü gibi en düzgün yanıt tek bölgeli yani bölgelendirmeden dilsel biçimlendirici atanması sonucu elde edilen yanıttır. Sekiz bölgeli dilsel biçimlendirici atanması sonucu elde edilen yanıttan görüldüğü gibi bölgelendirme

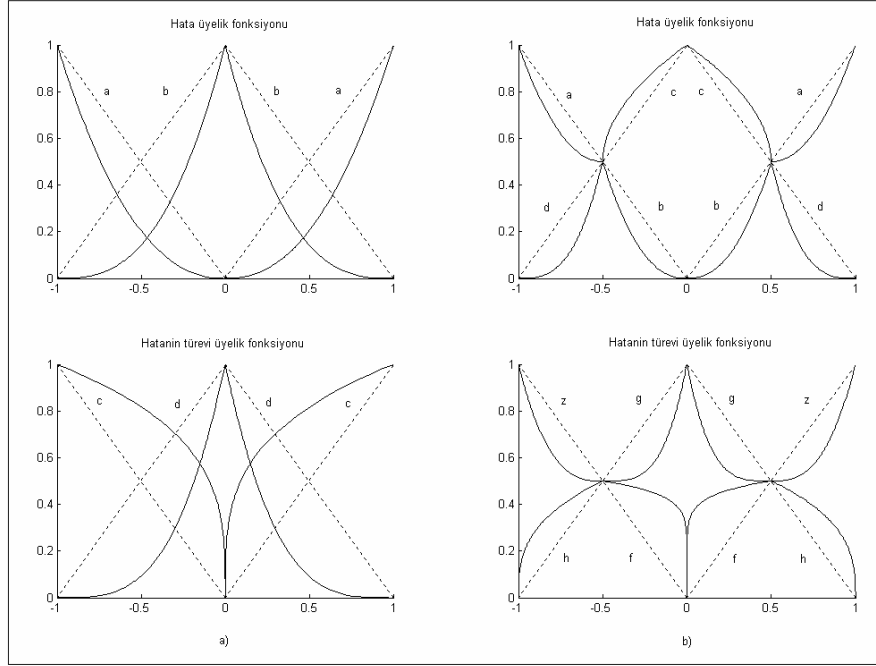
sayısı arttıkça sistem yanıtı dilsel biçimlendirici atanmamış durumdaki ilk yanıtı yaklaşımaktadır. Bunun nedeni bölge sayısı arttıkça uygulanan dilsel biçimlendiricilerin üyelik fonksiyonu üzerindeki etkisinin azalması ve üyelik fonksiyonu şeklinin ilk haldeki üçgen duruma yaklaşmasıdır. Bu durum sekiz bölgeli dilsel biçimlendirici atama sonucu elde edilen yeni üyelik fonksiyonlarında açıkça görülmektedir. Böylece doğrusal sistemlerde Liu' nun önerdiği bölgesel dilsel biçimlendirici atama yönteminin iyi bir sonuç vermediği, en iyi sonucun tek bölgeli dilsel biçimlendirici değeri atama durumunda elde edildiği görülmektedir. Çünkü tek bölgeli durumda üyelik fonksiyonlarının şekli çok bölgeli duruma göre daha fazla değişebilmekte ve dolayısıyla kontrol yüzeyini de daha fazla etkileyebilmektedir.

Bu sonuç elde edildikten sonra her bir giriş değişkeni için ortak dilsel biçimlendirici değerleri yerine farklı dilsel biçimlendirici değerleri kullanılması durumunda performansın nasıl etkileneceği araştırılmıştır. Bölgesel dilsel biçimlendiriciler atamanın bir getirisi olmadığı görüldüğünden burada sadece 2 bölgeye kadar bölgelendirme denenmiştir. Elde edilen dilsel biçimlendirici değerleri Tablo 5.4' de gösterilmiştir.

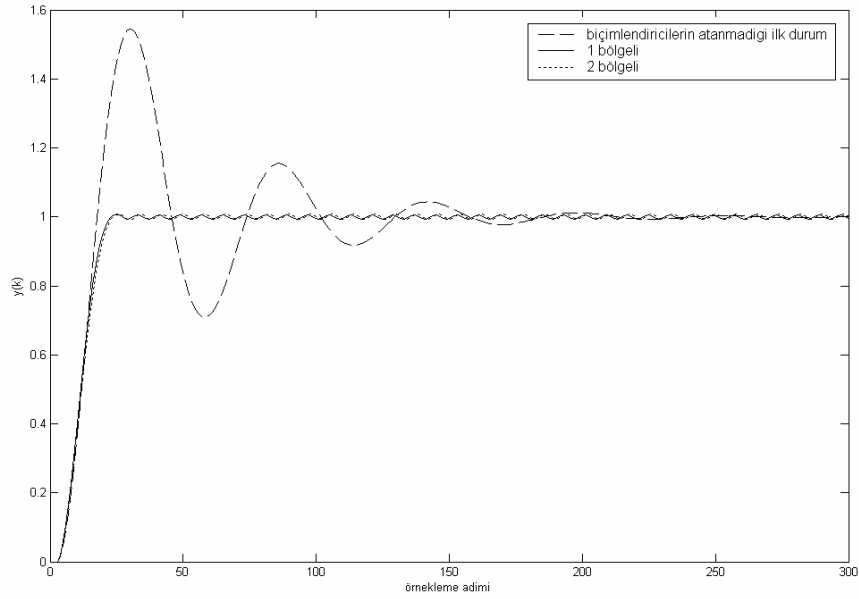
Tablo 5.4: Doğrusal Sistem İçin Ölçekleme Çarpanlarının Düzgün Seçilmemesi Durumunda Belirlenen Farklı Dilsel Biçimlendirici Değerleri

Bölge sayısı	Dilsel biçimlendirici değeri		Amaç ölçütü değeri
Ayarlamasız ilk durum	-		4.7227
Tek bölge	Hata üyelik fonksiyonları	a = 2.2912 b = 2.8128	47.8283
	Hatanın türevi üyelik fonksiyonları	c = 0.2923 d = 3.4354	
İki bölge	Hata üyelik fonksiyonları	a = 2.0540 b = 2.3119 c = 0.5190 d = 3.0190	45.4413
	Hatanın türevi üyelik fonksiyonları	z = 2.8451 f = 0.1149 g = 3.7131 h = 0.3470	

Bu dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen yeni üyelik fonksiyonları ve sistem yanıtları sırasıyla Şekil 5.8 ve Şekil 5.9' da gösterilmiştir.



Şekil 5.8: Doğrusal sistem için belirlenen farklı dilsel biçimlendirici değerleri ile elde edilen yeni üyelik fonksiyonları a) Tek bölgeli durum b) İki bölgeli durum



Şekil 5.9: Doğrusal sistem için belirlenen farklı dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları

Görüldüğü gibi her bir değişkene farklı dilsel biçimlendirici değeri atanması durumunda daha etkili sonuçlar elde edilmektedir. Ayrıca ortak dilsel biçimlendirici

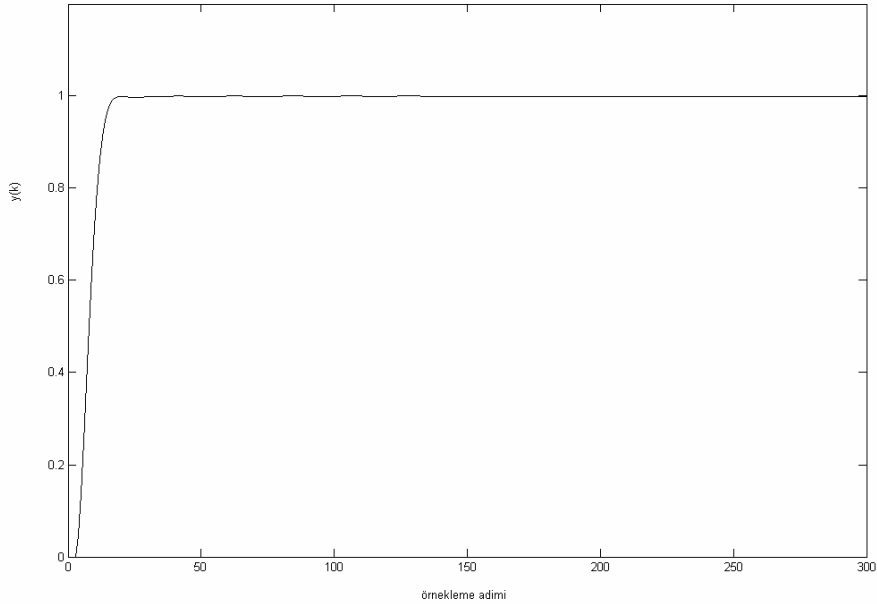
değeri atama durumunda görüldüğü gibi burada da bölgelendirme sayısını arttırmak sistem performansını kötü yönde etkilemektedir.

Düzgün seçilmemiş ölçekleme çarpanları kullanılması durumunda dilsel biçimlendiricilerin sistem performansı üzerinde önemli büyüklükte iyileştirici etkiye sahip oldukları görüldükten sonra düzgün seçilmiş ölçekleme çarpanları kullanılması durumunda dilsel biçimlendiricilerin sistem performansı üzerindeki etkileri incelenmiştir. Bu durum için GA tarafından belirlenen ölçekleme çarpanları Tablo 5.5’ de gösterilmektedir.

Tablo 5.5: Doğrusal Sistem İçin Düzgün Seçilmiş Ölçekleme Çarpanları

Giriş Ölçekleme Çarpanları	Çıkış Ölçekleme Çarpanı
$g1 = 0.41$ $g2 = 0.56$	$c1 = 0.56$

Bu ölçekleme çarpanlarının kullanılması ile elde edilen sistem yanıtı Şekil 5.10’ da gösterilmiştir.



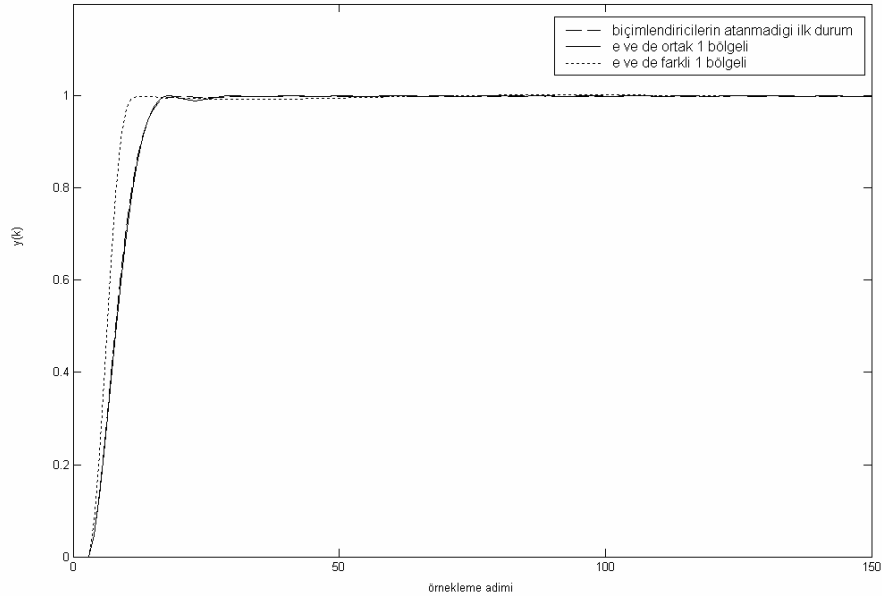
Şekil 5.10: Doğrusal sistem için düzgün seçilmiş ölçekleme çarpanları ile elde edilen sistem yanıtı

Daha önceki denemelerde ortaya çıkan sonuçların ışığında bu uygulamada sadece tek bölge dilsel biçimlendirici değeri atanarak sistem yanıtları incelenmiştir. GA ile elde edilen dilsel biçimlendirici değerleri Tablo 5.6’ da gösterilmiştir.

Tablo 5.6: Doğrusal Sistem İçin Ölçekleme Çarpanlarının Düzgün Seçilmesi Durumunda Belirlenen Dilsel Biçimlendirici Değerleri

Bölge sayısı	Dilsel biçimlendirici değeri		Amaç ölçütü değeri
Ayarlamasız ilk durum	-		32.6789
Ortak dilse biçimlendirici Tek bölge	a = 0.9124 b = 0.7703		34.6759
Farklı dilse biçimlendirici Tek bölge	Hata üyelik fonksiyonları	a = 1.3793 b = 3.6443	49.8386
	Hatanın türevi üyelik fonksiyonları	c = 1.6747 d = 2.5501	

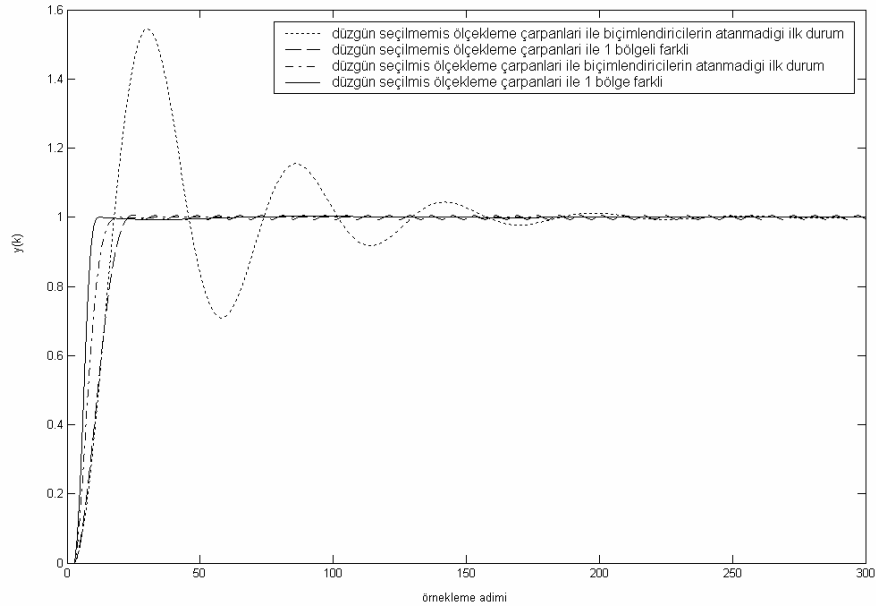
Bu dilse biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları Şekil 5.11’ de gösterilmiştir.



Şekil 5.11: Doğrusal sistem için belirlenen dilse biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları

Burada da açıkça görülmektedir ki üyelik fonksiyonlarına farklı dilsel biçimlendirici değeri atamak performansı daha çok iyileştirmektedir.

Son olarak Şekil 5.12’ de, düzgün seçilmiş ölçekleme çarpanları ve düzgün seçilmemiş ölçekleme çarpanları ile elde edilen en iyi sonuçlar karşılaştırılmıştır.



Şekil 5.12: Doğrusal sistem için düzgün seçilmiş ve düzgün seçilmemiş ölçekleme çarpanları ile elde edilen en iyi sonuçların karşılaştırılması

Görüldüğü gibi doğrusal sistemlerde en iyi cevap düzgün seçilmiş ölçekleme çarpanları ile her bir giriş değişkeni için farklı dilsel biçimlendirici değeri atanması ile elde edilmektedir. Böylece Liu’ nun önerdiği yöntemin doğrusal sistemler üzerinde bir geçerliliği olmadığı ispatlanmıştır.

5.2.2 Doğrusal Olmayan Sistem Üzerinde Yapılan Çalışmalar

Bu bölümdeki çalışmalar

$$\begin{aligned}
 y(k+1) = & 0.2y^2(k) + 0.2y(k-1) \\
 & + 0.4 \sin \left[0.5(y(k) + y(k-1)) \right] \\
 & \times \cos \left[0.5(y(k) + y(k-1)) \right] + 1.2x(k)
 \end{aligned} \tag{5.7}$$

modeline sahip doğrusal olmayan sistem üzerinde gerçekleştirilmiştir. Burada 0.1 örnekleme zamanı olmak üzere hatanın değişimi

$$\Delta e = \frac{e(k) - e(k-1)}{0.1} \quad (5.8)$$

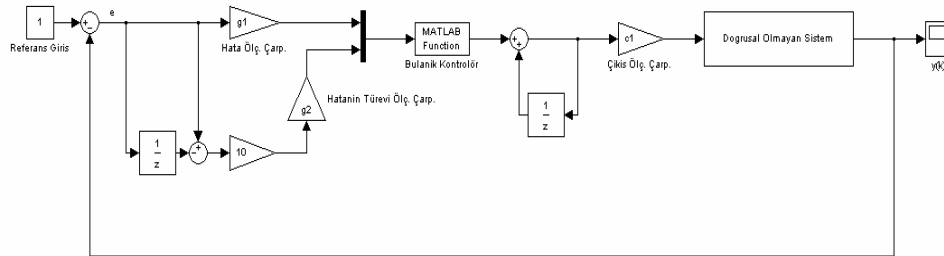
şeklinde ifade edilmiştir.

Burada da yine ilk olarak düzgün seçilmemiş ölçekleme çarpanları kullanılması durumunda dilsel biçimlendiricilerin etkisi incelenmiştir. Bu durum için seçilen ölçekleme çarpanları Tablo 5.7' de gösterilmektedir.

Tablo 5.7: Doğrusal Olmayan Sistem İçin Düzgün Seçilmemiş Ölçekleme Çarpanları

Giriş Ölçekleme Çarpanları	Çıkış Ölçekleme Çarpanı
$g1 = 0.5$ $g2 = 0.7$	$c1 = 0.9$

Sistemin Simulink blok diyagramı Şekil 5.13' de gösterilmiştir.



Şekil 5.13: Doğrusal olmayan sistemin kapalı çevrim Simulink blok diyagramı

Yine ilk olarak her iki giriş değişkeni için ortak dilsel biçimlendirici değeri atanmış, daha sonra farklı dilsel biçimlendiriciler atanarak sonuçlar incelenmiştir. Üyelik fonksiyonlarına ortak ve farklı bölgesel dilsel biçimlendiricilerin atanması durumları için GA ile belirlenen dilsel biçimlendirici değerleri sırasıyla Tablo 5.8 ve Tablo 5.9' da gösterilmiştir.

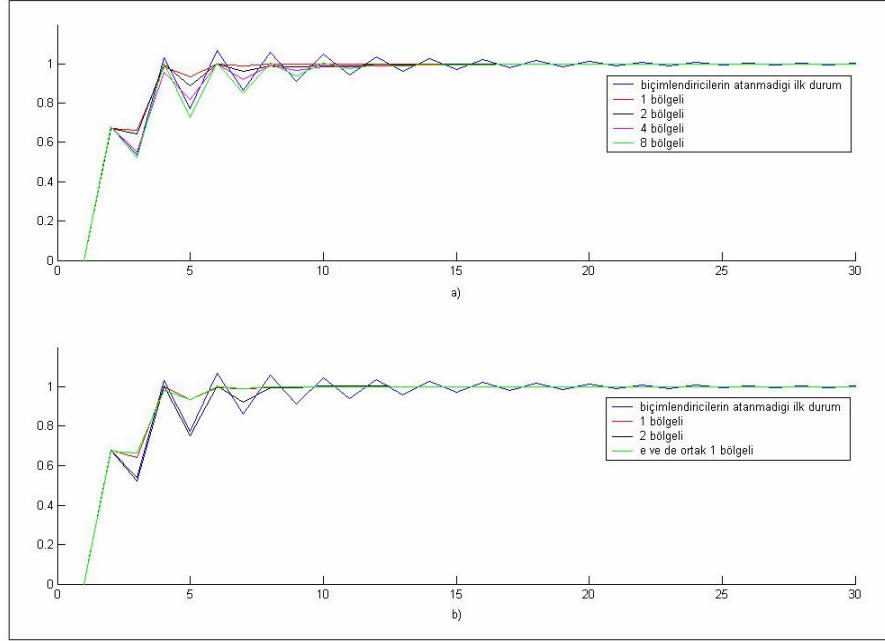
Tablo 5.8: Doğrusal Olmayan Sistem İçin Ölçekleme Çarpanlarının Düzgün Seçilmemesi Durumunda Belirlenen Ortak Dilsel Biçimlendirici Değerleri

Bölge sayısı	Dilsel biçimlendirici değeri	Amaç ölçütü değeri
Ayarlamasız ilk durum	-	0.7537
Tek bölge	a = 1.2888, b = 0.1988	0.7826
İki bölge	a = 3.1603, b = 1.2970 c = 0.3206, d = 0.1327	0.7788
Dört bölge	a = 2.5125, b = 0.9857 c = 3.4886, d = 1.3475 z = 0.3116, f = 0.1996 g = 0.1084, h = 0.1012	0.7639
Sekiz bölge	a1 = 2.8360, a2 = 0.9727 a3 = 1.4401, a4 = 0.2566 a5 = 1.7816, a6 = 3.4814 a7 = 3.4392, a8 = 1.3277 b1 = 1.7427, b2 = 1.2612, b3 = 0.8230, b4 = 2.3892, b5 = 1.4591, b6 = 0.6719 b7 = 0.1278, b8 = 0.2471	0.7505

Tablo 5.9: Doğrusal Olmayan Sistem İçin Ölçekleme Çarpanlarının Düzgün Seçilmemesi Durumunda Belirlenen Farklı Dilsel Biçimlendirici Değerleri

Bölge sayısı	Dilsel biçimlendirici değeri		Amaç ölçütü değeri
Ayarlamasız ilk durum	-		0.7537
Tek bölge	Hata üyelik fonksiyonları	a = 1.4912 b = 1.7276	0.7808
	Hatanın türevi üyelik fonksiyonları	c = 1.4102 d = 0.3931	
İki bölge	Hata üyelik fonksiyonları	a = 3.0704 b = 1.0284 c = 1.8533 d = 1.0003	0.7563
	Hatanın türevi üyelik fonksiyonları	z = 2.9172 f = 1.7255 g = 0.1028 h = 2.8575	

Bu dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları Şekil 5.14’ de gösterilmiştir.



Şekil 5.14: Doğrusal olmayan sistem için ölçekleme çarpanlarının düzgün seçilmemesi durumunda belirlenen dilsel biçimlendirici değerlerinin uygulanması sonucu elde edilen sistem yanıtları a) Ortak dilsel biçimlendiricilerin atandığı durum b) Farklı dilsel biçimlendiricilerin atandığı durum

Grafiklerden ve amaç ölçütü değerlerinden görüldüğü gibi elde edilen en iyi yanıt her iki giriş değişkeni üyelik fonksiyonlarına tek bölgeyi ortak dilsel biçimlendirici değeri atanması ile elde edilmiştir. Burada doğrusal sistemlerde olduğundan farklı bir sonuç ortaya çıkmakta ve farklı dilsel biçimlendiriciler atamak yerine ortak dilsel biçimlendiriciler atamak az bir farkla da olsa daha iyi sonuç vermektedir.

Düzgün seçilmiş ölçekleme çarpanlarının kullanılması sonucu dilsel biçimlendiricilerin etkisini görmek için bulanık kontrolörde Tablo 5.10' da gösterilen GA ile belirlenen ölçekleme çarpanları kullanılmıştır.

Tablo 5.10: Doğrusal Olmayan Sistem İçin Düzgün Seçilmiş Ölçekleme Çarpanları

Giriş Ölçekleme Çarpanları	Çıkış Ölçekleme Çarpanı
$g1 = 1.62$ $g2 = 1.64$	$c1 = 0.3$

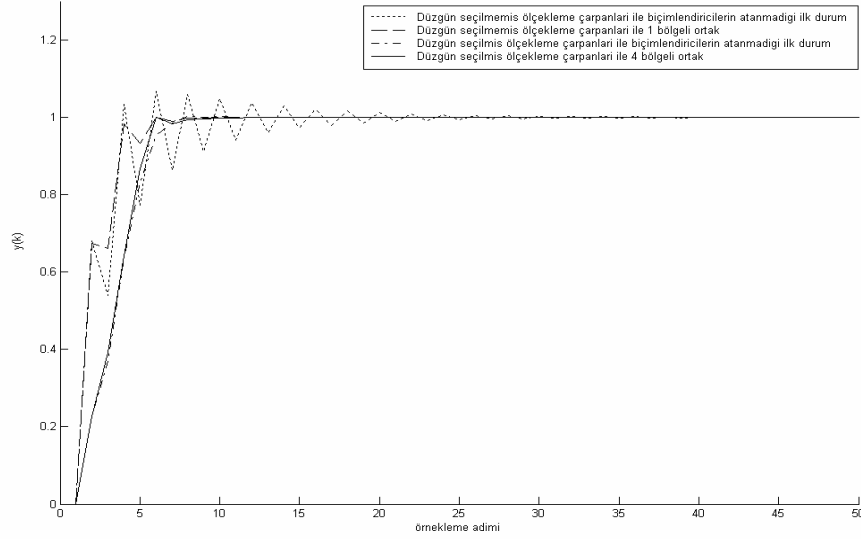
Üyelik fonksiyonlarına uygulanacak ortak ve farklı bölgesel dilsel biçimlendirici değerleri Tablo 5.11 ve Tablo 5.12' de sırasıyla gösterilmiştir

Tablo 5.11: Doğrusal Olmayan Sistem İçin Ölçekleme Çarpanlarının Düzgün Seçilmesi Durumunda Belirlenen Ortak Dilsel Biçimlendirici Değerleri

Bölge sayısı	Dilsel biçimlendirici değeri	Amaç ölçütü değeri
Ayarlamasız ilk durum	-	0.6442
Tek bölge	a = 2.7252, b = 2.5489	0.6456
İki bölge	a = 1.6229, b = 3.7066 c = 1.4406, d = 3.3637	0.6463
Dört bölge	a1 = 2.9574, a2 = 3.6821 a3 = 0.1429, a4 = 3.6220 b1 = 0.6711, b2 = 0.1084 b3 = 2.3027, b4 = 0.4344	0.6553
Sekiz bölge	a1 = 0.5644, a2 = 1.0954 a3 = 0.2614, a4 = 0.9997 a5 = 2.1142, a6 = 2.4125 a7 = 0.1557, a8 = 1.4275 b1 = 1.5818, b2 = 0.9946 b3 = 3.3487, b4 = 1.3567 b5 = 1.9700, b6 = 2.6362 b7 = 1.7338, b8 = 1.7611	0.6491

Tablo 5.12: Doğrusal Olmayan Sistem İçin Ölçekleme Çarpanlarının Düzgün Seçilmesi Durumunda Belirlenen Farklı Dilsel Biçimlendirici Değerleri

Bölge sayısı	Dilsel biçimlendirici değeri		Amaç ölçütü değeri
Ayarlamasız ilk durum	-		0.6442
Tek bölge	Hata üyelik fonksiyonları	a = 1.7335 b = 2.6507	0.6450
	Hatanın türevi üyelik fonksiyonları	c = 1.4591 d = 1.5894	
İki bölge	Hata üyelik fonksiyonları	a = 1.9431 b = 3.4459 c = 3.0501 d = 2.8457	0.6471
	Hatanın türevi üyelik fonksiyonları	z = 3.1067 f = 3.4056 g = 0.8424 h = 1.1913	



Şekil 5.16: Doğrusal olmayan sistem için düzgün seçilmiş ve düzgün seçilmemiş ölçekleme çarpanları ile elde edilen en iyi sonuçların karşılaştırılması

Doğrusal olmayan sistemlerde doğrusal sistemlerin aksine yapılarındaki düzensizlikten ötürü bölgesel dilsel biçimlendirici atamanın iyileştirici bir etkisi olduğu görülmektedir. Bu örnekte düzgün seçilmiş ölçekleme çarpanları kullanıldığında, üyelik fonksiyonları dört bölgeye ayrılıp her bir bölge için dilsel biçimlendirici değeri atanması durumunda en iyi sistem yanıtı elde edilmiştir. Fakat Şekil 5.16’ da dikkat çekici bir nokta, düzgün seçilmemiş ölçekleme çarpanları kullanılması durumunda tek bölge dilsel biçimlendirici değeri uygulamakla elde edilen yanıtın, düzgün seçilmiş ölçekleme çarpanları kullanılması ve dört bölge dilsel biçimlendirici değeri uygulanması sonucu elde edilen yanıtın daha iyi olmasıdır. Bunun nedeni ölçekleme çarpanları ve dilsel biçimlendiricilerin arasındaki ilişkide yatar. Ölçekleme çarpanları giriş değerlerini normalize edilmiş uzayda belli bir aralığa düşürmeyi amaçlamaktadır. Üyelik fonksiyonlarının şekli ise bu belirlenmiş aralıkta giriş değişkenlerinin üyelik derecelerini belirlemektedir. Görüldüğü gibi doğrusal sistemlerde en düzgün yanıt en uygun ölçekleme çarpanlarının seçilmesi ve en uygun dilsel biçimlendirici değerlerinin atanması ile elde edilmektedir. Fakat doğrusal olmayan sistemlerde yapılarındaki düzensizlikten ötürü ölçekleme çarpanları düzgün olarak seçildiğinde Şekil 5.16’ da görüldüğü gibi aşım kriterini sağlamaya çalışırken yükselme zamanı kriterinden ödün verilmektedir. Buna karşın düzgün seçilmemiş ölçekleme çarpanlarının kullanılması durumunda ise hızlı bir yükselme sağlanabilmesine karşın aşım ve yerleşme zamanı gibi kriterler

yerine getirilememektedir. Ölçekleme çarpanları dilsel biçimlendirici değerleri atanmadan üçgen üyelik fonksiyonlarının kullanıldığı ilk durum için belirlendiğinden, düzgün seçilmeleri durumunda yukarıda bahsedilen ilişkiden ötürü giriş değerlerini sadece normalize uzayda üçgen üyelik fonksiyonlarının kullanılması durumu için en uygun yere düşürürler. Bu yüzden dilsel biçimlendiricilerle yapılan üyelik fonksiyonunun üçgen şeklini bozan büyük değişiklikler sistem yanıtı üzerinde olumsuz etkilere neden olmaktadır. Grafiklerden görüldüğü gibi ölçekleme çarpanları ne kadar düzgün seçilirse kontrol performansı üçgen üyelik fonksiyonuna o kadar bağlı olduğundan dilsel biçimlendiricilerin düzeltme etkisi de o kadar az olmaktadır. Ölçekleme çarpanları düzgün seçilmediği durumda ise dilsel biçimlendiricilerin düzeltme etkisi daha büyük olmaktadır. Çünkü bu durumda giriş değerlerinin normalize uzayda düşürüleceği aralık ölçekleme çarpanları ile optimal olarak belirlenmediğinden kontrol performansının iyileştirilmesinde üyelik fonksiyonlarının şeklinin belirlenmesine daha büyük bir iş düşmektedir. Böylece esnek bırakılmış bu aralıkta dilsel biçimlendiriciler daha büyük değişiklikler yaparak kötü olan kontrol performansını daha büyük ölçüde iyileştirebilmektedir. Bu yüzden doğrusal olmayan sistemlerde düzgün seçilmemiş ölçekleme çarpanlarının kullanılması durumunda üyelik fonksiyonlarının şeklinin dilsel biçimlendiriciler kullanılarak ayarlanması düzgün ölçekleme çarpanlarının kullanılması durumuna göre daha iyi sonuç verebilmektedir.

Yukarıdaki çalışmalarda dilsel biçimlendiricilerin sistem yanıtı üzerinde iyileştirici etkileri açıkça görülmüştür. Ayrıca doğrusal sistemler için düzgün seçilmiş ölçekleme çarpanlarının kullanılmasının gerekli olduğu, bölgesel dilsel biçimlendirici atamanın bir etkisi olmadığı ve ortak dilsel biçimlendirici değeri atamak yerine her bir giriş değişkeni için farklı dilsel biçimlendirici değerleri atamanın daha etkili olduğu ispatlanmıştır. Doğrusal olmayan sistemler için ise bölgesel dilsel biçimlendirici değerleri belirlemenin iyi sonuçlar verebildiği ve düzgün seçilmiş ölçekleme çarpanları ile dilsel biçimlendirici atamak yerine düzgün seçilmemiş ölçekleme çarpanlarının kullanılması durumunda dilsel biçimlendirici atamanın daha etkili olabildiği görülmüştür.

Bu sonuçlar ışığında üyelik fonksiyonlarına süreç boyunca optimal belirlenmiş sabit dilsel biçimlendirici değerleri atamak yerine Bölüm 4’ de belirtilen, süreç boyunca

hata ve hatanın türevinden bilgi alarak çevrim içi durumda üyelik fonksiyonlarına uygun dilsel biçimlendirici değerlerini atayan yeni bir yöntem önerilmiştir.

5.3 Ayar Parametre Sayısının Azaltılması ve Yöntemin Zaman Sabitinin Beş Katı Büyüklüğünde Ölü Zaman Değerine Sahip Doğrusal Sistem Üzerinde Uygulanması

Bölüm 4’ de bahsedildiği gibi yöntem oluşturulurken ilk aşamalarda simetri korunarak tüm üyelik fonksiyonlarına dilsel biçimlendirici atama yoluna gidilmiştir. Böylece her bir giriş değişkeni üzerinde PB ve NB üyelik fonksiyonlarına ortak bir dilsel biçimlendirici ve S üyelik fonksiyonuna bir dilsel biçimlendirici olmak üzere iki giriş değişkeni için toplam dört dilsel biçimlendirici değeri belirlenmektedir. Bu durumda iki kademeli anahtarlama mekanizmasından ötürü k parametresi için toplam sekiz değer belirlenmesi gerekmektedir. Yöntemin bu ilk hali

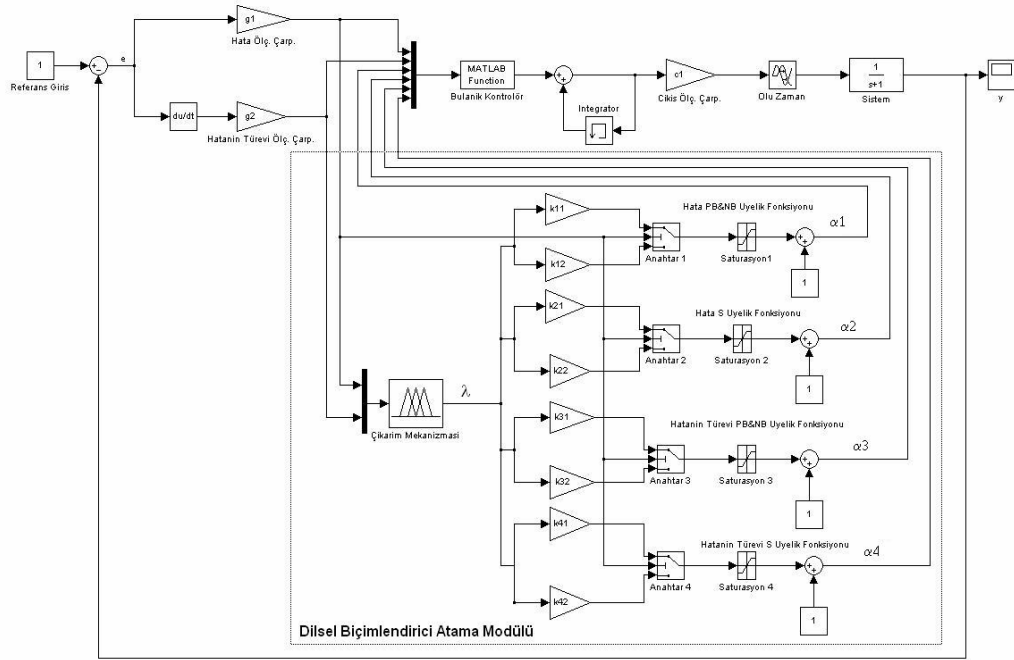
$$H(s) = \frac{1}{s+1} e^{-5s} \quad (5.9)$$

transfer fonksiyonuna ve zaman sabitinin beş katı büyüklüğünde ölü zaman değerine sahip doğrusal sistem üzerinde uygulanmıştır. Bölüm 5.2.1’ de yapılan çalışmalar sonucu doğrusal sistemlerde düzgün seçilmiş ölçekleme çarpanlarının kullanılmasının daha iyi sonuç verdiği görüldüğünden, bu uygulamada Tablo 5.13’ de gösterilen GA ile belirlenen ölçekleme çarpanları kullanılmıştır.

Tablo 5.13: Doğrusal Sistem İçin Düzgün Seçilmiş Ölçekleme Çarpanları

Giriş Ölçekleme Çarpanları	Çıkış Ölçekleme Çarpanı
$g1 = 0.31$ $g2 = 0.68$	$c1 = 0.1$

şeklinde GA ile belirlenen değerler kullanılmıştır. Sistemin Simulink blok diyagramı Şekil 5.17’ de gösterilmiştir.



Şekil 5.17: Sekiz parametrenin aratıldığı yöntemin ilk hali için ölü zamanlı doğrusal sistemin Simulink blok diyagramı

GA ile belirlenen uygun k parametreleri Tablo 5.14’ de gösterilmiştir. Hata değişkeni PB ve NB üyelik fonksiyonları için k_{11} ve k_{12} , S üyelik fonksiyonu için k_{21} ve k_{22} , hatanın türevi değişkeni PB ve NB üyelik fonksiyonları için k_{31} ve k_{32} , S üyelik fonksiyonu için k_{41} ve k_{42} parametreleri kullanılmıştır.

Tablo 5.14: Doğrusal Sistemde Sekiz Parametrenin Aratıldığı Yöntemin İlk Hali İçin Belirlenen k Değerleri

Anahtarlama	Belirlenen k parametreleri		Amaç ölçütü değeri
1. kademe (ölü zaman boyunca)	Hata	$k_{11} = -4$ $k_{21} = -1.5$	9,8274
	Hatanın türevi	$k_{31} = 4$ $k_{41} = 0$	
2. kademe (ölü zamandan sonra)	Hata	$k_{12} = 3.15$ $k_{22} = 8.5$	
	Hatanın türevi	$k_{32} = -5$ $k_{42} = 2$	

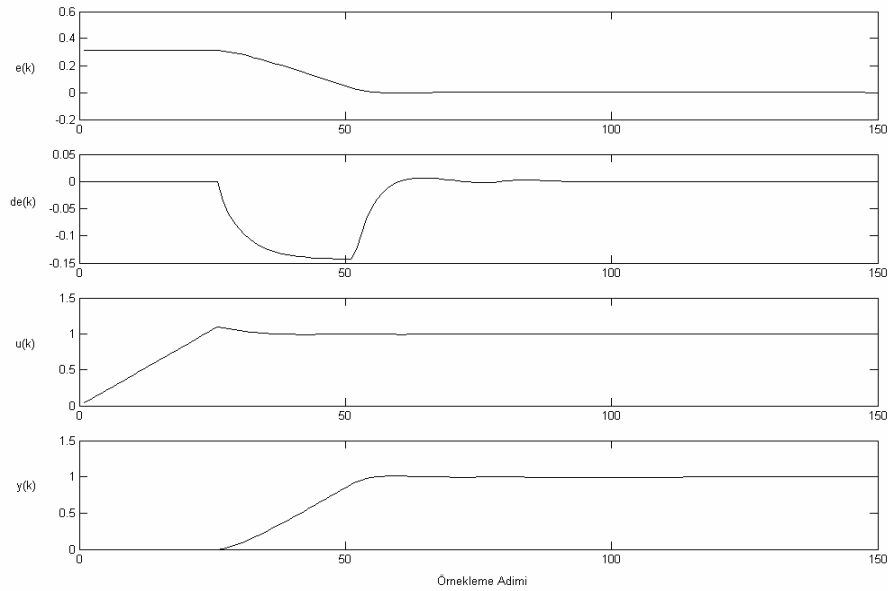
Yeni yöntem ile elde edilen sonuçlar, doğrusal sistemler için en uygun ayarlama yöntemi olan giriş değişkenleri üyelik fonksiyonlarına tek bölge farklı dilsel

biçimlendirici değerlerinin atanması durumunda elde edilen sonuçlar ile karşılaştırılmıştır. Bu durum için elde edilen tek bölge farklı dilsel biçimlendirici değerleri Tablo 5.15’ de gösterilmiştir.

Tablo 5.15: Doğrusal Sistem İçin Ölçekleme Çarpanlarının Düzgün Seçilmesi Durumunda Belirlenen Tek Bölge Farklı Dilsel Biçimlendirici Değerleri

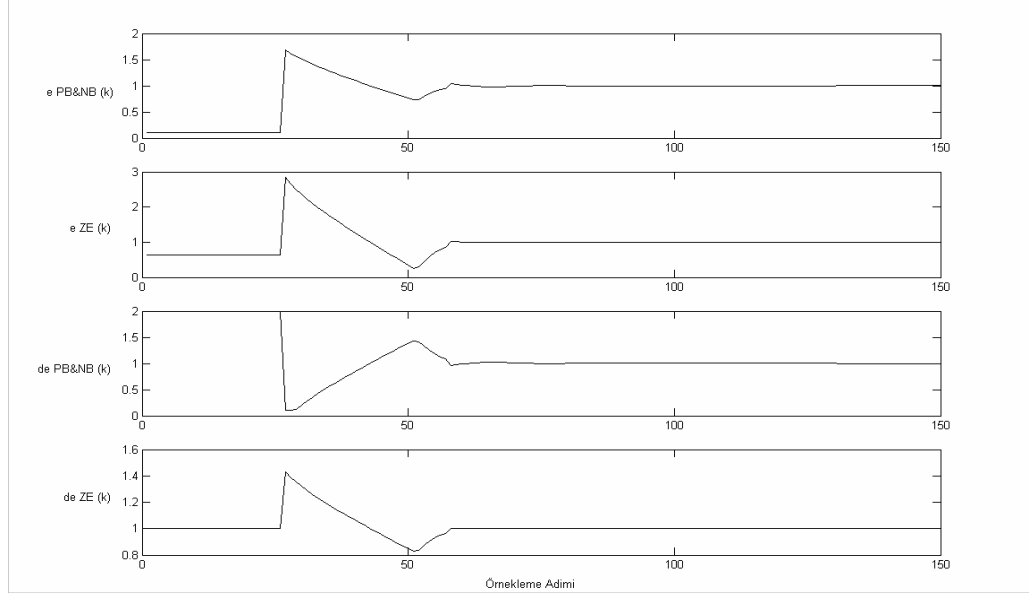
Bölge sayısı	Dilsel biçimlendirici değeri		Amaç ölçütü değeri
Ayarlamasız ilk durum	-		6.2976
Tek bölge	Hata üyelik fonksiyonları	a = 1.7563 b = 3.8411	7.4185
	Hatanın türevi üyelik fonksiyonları	c = 1.4622 d = 3.6355	

Yeni yöntemin uygulanması sonucu elde edilen hata, hatanın türevi, kontrol işareti ve sistem yanıtı Şekil 5.18’ de gösterilmiştir.



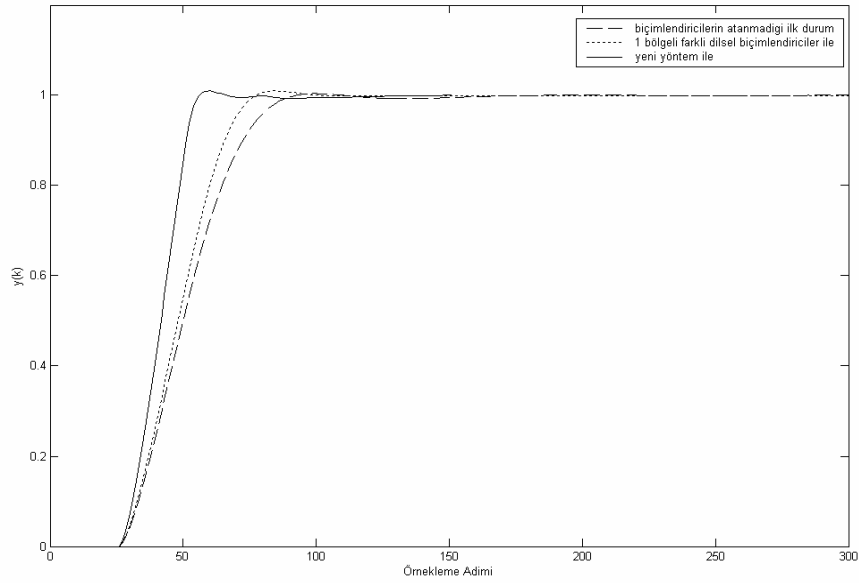
Şekil 5.18: Doğrusal sistemde sekiz parametrenin aratıldığı yöntemin ilk halinin uygulanması sonucu elde edilen hata, hatanın türevi, kontrol işareti ve sistem yanıtı

Şekil 5.19’ da çevrim içi olarak atanan dilsel biçimlendiricilerin grafiği görülmektedir.



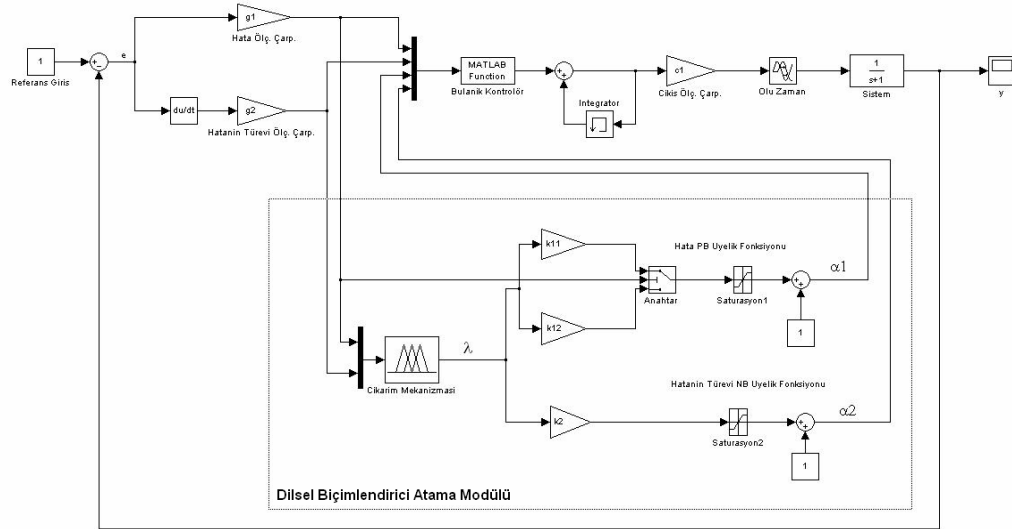
Şekil 5.19: Doğrusal sistemde sekiz parametrenin aratıldığı yöntemin ilk halinin uygulanması sonucu kontrolöre çevrim içi olarak atanan dilsel biçimlendiricilerin grafiği

Şekil 5.20’ de yeni yöntemin ilk halinin uygulanması ve tek bölge farklı dilsel biçimlendirici değerlerinin atanması ile elde edilen sonuçlar karşılaştırılmıştır.



Şekil 5.20: Doğrusal sistem üzerinde sekiz parametrenin aratıldığı yöntemin ilk halinin uygulanması ve tek bölge farklı dilsel biçimlendirici değerlerinin atanması ile elde edilen sonuçların karşılaştırılması

Ayarlama yapmadan üçgen üyelik fonksiyonlarının kullanılması durumunda elde edilen sistem yanıtının yerleşme zamanı 17.4 saniye, tek bölge farklı dilsel biçimlendiricilerin atanması sonucu elde edilen yanıtın yerleşme zamanı 15 saniye ve yeni önerilen yöntemle elde edilen sistem yanıtının yerleşme zamanı 11 saniye şeklindedir. Görüldüğü gibi klasik dilsel biçimlendirici atama yöntemiyle sistem performansı üzerinde %13.8 kadar bir iyileşme sağlanırken önerilen yeni yöntemle sistem performansı üzerinde %36.8 değerinde bir iyileşme sağlanmaktadır. Yeni yöntemle elde edilen yanıt büyük ölü zamana karşın oldukça hızlı ve düzgündür. Görüldüğü gibi bu yöntem sayesinde PI tipi bulanık kontrolör geçici halde oldukça etkin bir kontrol sağlayabilmektedir. Bu çarpıcı sonuçlara karşın yöntemde belirlenecek parametre sayısının çokluğu bir dezavantajdır. Bu yüzden parametre azaltma yoluna gidilmiştir. Bunun için tüm üyelik fonksiyonlarına dilsel biçimlendirici atamak yerine sadece baskın olan hata PB ve hatanın türevi NB üyelik fonksiyonlarına dilsel biçimlendirici atama yoluna gidilerek parametre sayısı ikiye kadar indirilmiştir. Yöntemin son halinin Simulink blok diyagramı Şekil 5.21’ de gösterilmektedir.



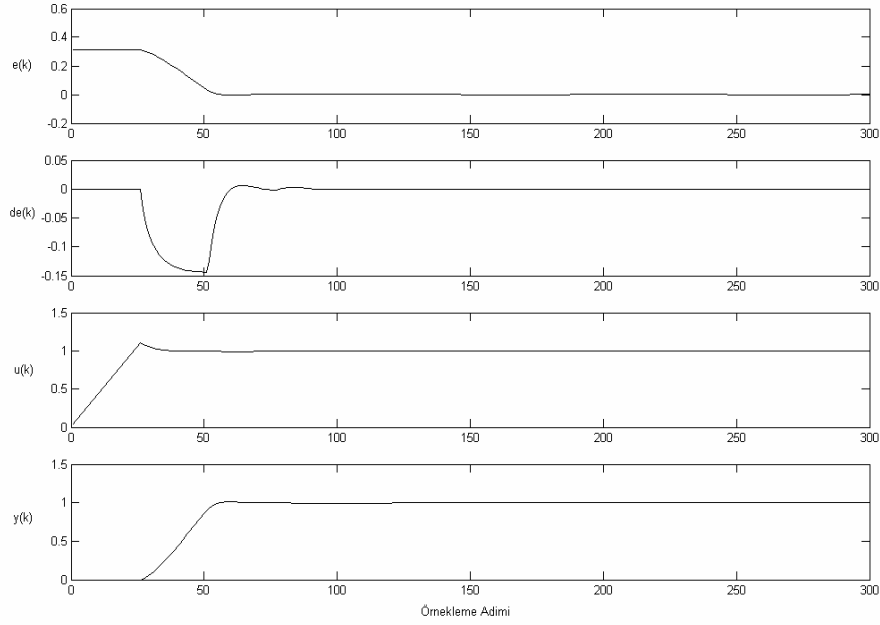
Şekil 5.21: İki parametrenin ayarlandığı yöntemin son halinin Simulink blok diyagramı

Tablo 5.16’ da yöntemin bu son hali için GA ile elde edilmiş k parametre değerleri görülmektedir.

Tablo 5.16: İki Parametrenin Aratıldığı Yöntemin Son Hali İçin Belirlenen k Değerleri

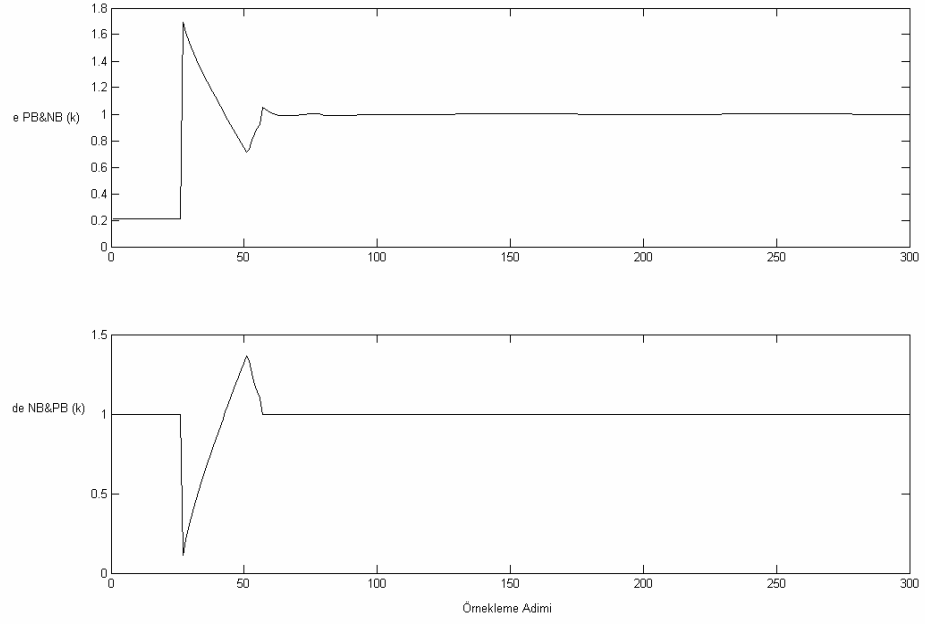
Anahtarlama	Belirlenen k parametreleri		Amaç ölçütü değeri
1. kademe (ölü zaman boyunca)	Hata PB&NB	$k_{11} = -3.2$	9,2211
2. kademe (ölü zamandan sonra)	Hata PB&NB	$k_{12} = 3.2$	
Tüm süreç boyunca	Hatanın türevi PB&NB	$k_2 = -4.1$	

Belirlenen bu değerlerin uygulanması sonucunda elde edilen hata, hatanın türevi, kontrol işareti ve sistem yanıtı Şekil 5.22’ de gösterilmiştir.



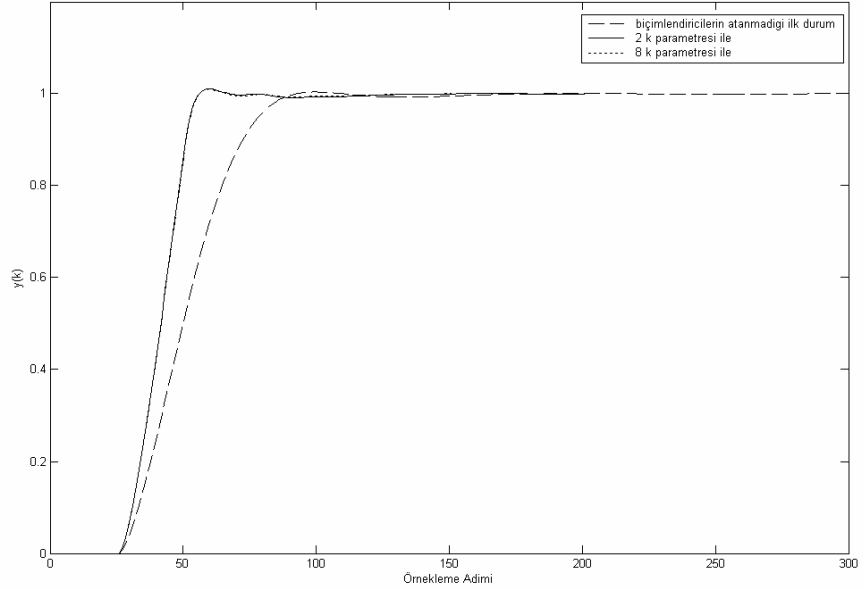
Şekil 5.22: İki parametrenin aratıldığı yöntemin son hali için belirlenen k değerlerinin uygulanması sonucu doğrusal sistemde elde edilen hata, hatanın türevi, kontrol işareti ve sistem yanıtı

Şekil 5.23’ de çevrim içi olarak atanan dilsel biçimlendirici değerleri görülmektedir.



Şekil 5.23: Yöntemin son halinin uygulanması ile doğrusal sistemde kontrolöre çevrim içi olarak atanan dilsel biçimlendirici değerleri

İki parametre ile elde edilen sistem yanıtı daha önce sekiz parametre kullanılarak elde edilen yanıtla karşılaştırılmış ve sonuç Şekil 5.24’ de gösterilmiştir.



Şekil 5.24: Önerilen yöntemin iki parametre atıldığı son hali ve sekiz parametre atıldığı ilk hali ile elde edilen sonuçların karşılaştırılması

denenmiştir. Kullanılan ölçekleme çarpanları bölüm 5.2.2’ deki değerlerle aynıdır. Tablo 5.17 ve Tablo 5.18’ de bu iki durum için belirlenmiş k parametreleri ve bölüm 5.2.2’ de elde edilen en iyi amaç ölçütü değerleri gösterilmektedir.

Tablo 5.17: Doğrusal Olmayan Sistem İçin Düzgün Seçilmemiş Ölçekleme Çarpanlarının Kullanılması Durumunda Belirlenen k Parametreleri

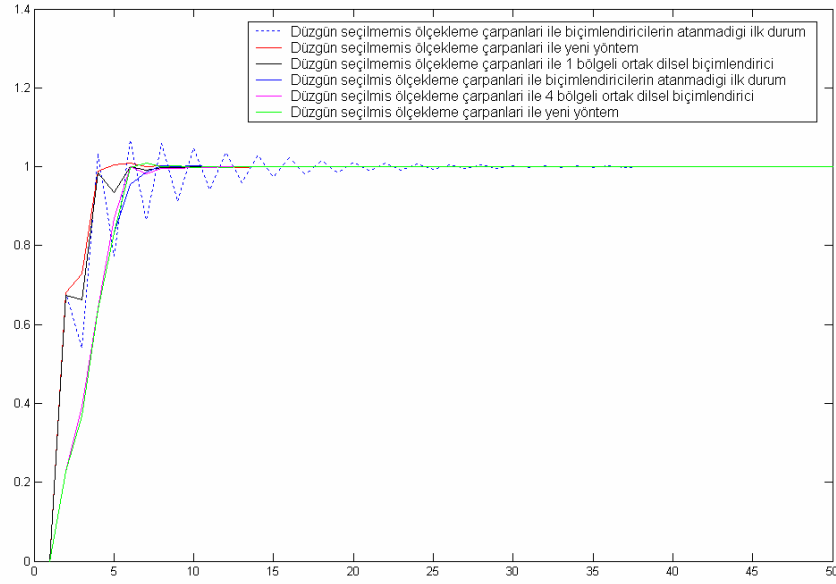
Anahtarlama	Belirlenen k parametreleri		Amaç ölçütü değeri
1. kademe	Hata	$k_{11} = -3.4$	0.7904
	Hatanın türevi	$k = 0$	
2. kademe	Hata	$k_{12} = 3.4$	
	Hatanın türevi	$k_2 = 1.9$	
Tek bölgesel ortak dilsel biçimlendirici			0.7826

Tablo 5.18: Doğrusal Olmayan Sistem İçin Düzgün Seçilmiş Ölçekleme Çarpanlarının Kullanılması Durumunda Belirlenen k Parametreleri

Anahtarlama	Belirlenen k parametreleri		Amaç ölçütü değeri
1. kademe	Hata	$k_{11} = -3$	0.6595
	Hatanın türevi	$k = 0$	
2. kademe	Hata	$k_{12} = 3$	
	Hatanın türevi	$k_2 = 0.9$	
Dört bölgesel ortak dilsel biçimlendirici			0.6553

Bölüm 4’ de bahsedildiği gibi ölü zamansız sistemlerde anahtarlama ölçeklenmiş hata değeri başlangıç değerinin %15’ ine ulaştığı zaman gerçekleşmektedir. Bu yüzden düzgün seçilmemiş ölçekleme çarpanları kullanılması durumunda 0.5 değerindeki hata ölçekleme çarpanından dolayı hata başlangıç değeri 0.5 olacağından anahtarlama bu değer %15’ i olan 0.075 değerinde gerçekleşecektir. Düzgün seçilmiş ölçekleme çarpanlarının kullanılması durumunda ise hata ölçekleme çarpanı değeri 1.62 olduğundan ve hata başlangıç değeri de bu değer olacağından anahtarlama bu değer %15’ i olan 0.243 değerinde gerçekleşecektir. Ayrıca yine bahsedildiği gibi parametre sayısını ikiye düşürmek için anahtarlamanın ilk kademesinde hatanın türevi NB üyelik fonksiyonu için k parametresi her zaman 0 olarak belirlenmiştir.

Belirlenen bu değerlerin uygulanması sonucu elde edilen sonuçlar ve bunların bölüm 5.1.2’ de bölgesel dilsel biçimlendiricilerin uygulanması sonucu elde edilen en iyi yanıtlarla karşılaştırılması Şekil 5.26’ da gösterilmiştir.



Şekil 5.26: Doğrusal olmayan sistemde yeni yöntemin ve bölgesel dilsel biçimlendiricilerin uygulanması sonucu elde edilen sistem yanıtları

Düzgün seçilmemiş ölçekleme çarpanlarının kullanılması durumunda sistem yanıtı 2.1 saniyede yerleşmektedir. Ayarlama sonucunda bölüm 5.1.2’ de belirlenmiş en iyi yanıt olan tek bölgesel ortak dilsel biçimlendirici değerlerinin kullanılmasıyla elde edilen yanıt 0.58 saniyede yerleşirken yeni yöntemin uygulanması sonucu elde edilen yanıt 0.41 saniyede yerleşmektedir. Görüldüğü gibi sabit dilsel biçimlendiricilerin kullanılması ile sistem performansı üzerinde %72.4 değerinde bir iyileşme sağlanırken önerilen yeni yöntemle %80.5 değerinde bir iyileşme sağlanmıştır.

Düzgün seçilmiş ölçekleme çarpanlarının kullanılması durumunda ise sistem yanıtı 0.73 saniyede yerleşmektedir. Ayarlama sonucunda yine bölüm 5.1.2’ de belirlenmiş en iyi yanıt olan dört bölgesel ortak dilsel biçimlendirici değerlerinin kullanılmasıyla elde edilen yanıt 0.68 saniyede yerleşirken önerilen yöntemin uygulanması sonucu elde edilen yanıt 0.59 saniyede yerleşmektedir. Böylece sabit dilsel biçimlendiricilerin kullanılması ile sistem performansı üzerinde %6.8 değerinde bir iyileşme sağlanırken önerilen yeni yöntemle %19.1 değerinde bir iyileşme sağlanmıştır.

Elde edilen sonuçlardan görüldüğü gibi önerilen yeni yöntemle en zor sistemlerde bile en basit kontrolör yapısı kullanarak ve sadece iki parametre ayarlayarak oldukça etkili sonuçlar elde edilmektedir. Böylece hem tasarım kolaylığı hem de bu kadar

basit bir yapıyla önerilen diğer yöntemlerden çok daha etkili bir kontrol performansı sağlanabilmektedir.

6. SONUÇLAR VE TARTIŞMALAR

Bu çalışmada bulanık kontrolör üyelik fonksiyonlarını dilsel biçimlendiriciler yardımıyla çevrim içi olarak ayarlayan yeni bir yöntem önerilmiştir. Bu yöntem sayesinde, giriş değişkenleri için üçgen yapıda basit üç üyelik fonksiyonu ve iki giriş değişkeni için toplamda sadece dokuz kural kullanılarak tasarlanan en basit yapıda bir bulanık kontrolörle oldukça etkili bir kontrol sağlanabilmektedir. Önerilen ayarlama yöntemi ile bulanık kontrolörün iç yapısına çevrim içi olarak etkide bulunulabilmekte ve kontrol süresince üyelik fonksiyonlarının şekli uygun biçimde değiştirilerek kontrol performansı büyük ölçüde artırılabilir.

Simülasyon sonuçlarından görüldüğü gibi doğrusal sistemlerde üyelik fonksiyonlarına bölgesel dilsel biçimlendiriciler atamanın bir faydası olmamaktadır. En iyi sonuç düzgün seçilmiş ölçekleme çarpanları kullanılması durumunda tek bölgesel dilsel biçimlendirici değerleri atanarak elde edilmektedir. Ayrıca giriş değişkenleri için ortak dilsel biçimlendiriciler atamak yerine farklı dilsel biçimlendiriciler belirlemek performansı daha da arttırmaktadır. Doğrusal sistemlerde üyelik fonksiyonlarına bu şekilde sabit dilsel biçimlendirici değerleri atandığında sistem performansında %13.8 kadar bir iyileşme görülürken, önerilen yeni yöntemle performans üzerinde %36.8 değerinde bir iyileşme sağlanmıştır. Görüldüğü gibi bu yöntem ile sabit dilsel biçimlendiricilerin kullanıldığı klasik ayarlama yönteminden yaklaşık üç kat daha iyi bir cevap elde edilmiştir.

Doğrusal olmayan sistemlerde ise bölgesel dilsel biçimlendiriciler atamanın performans üzerinde olumlu bir etkisi olduğu görülmüştür. Fakat bu durumda karşımıza çıkan sorun en uygun bölge sayısının nasıl belirleneceğidir. Deneme yanılma süreci zaman alıcı ve masraflı olabilir. Ayrıca bölge sayısını arttırmak aratılacak parametre sayısını da arttırmakta ve dolayısıyla parametrelerin uygun değerlerinin bulunması için GA ile arama sürecinde daha fazla zamana ihtiyaç olmaktadır. Örneğin sekiz bölge kullanıldığında 16 parametrenin belirlenmesi gerekmektedir. Bu da tasarım basitliği açısından istenen bir durum değildir. Buna

karşın önerilen yöntem oldukça basittir ve sadece iki parametre belirlenmesi gerekmektedir.

Doğrusal olmayan sistem üzerinde düzgün seçilmiş ölçekleme çarpanı ile yapılan simülasyonlarda en iyi yanıt dört bölgeci durumda giriş üyelik fonksiyonları için ortak dilsel biçimlendiricilerin kullanılması ile elde edilmiştir. Toplam sekiz parametrenin belirlendiği bu durumda sistem performansı üzerinde %6.8 değerinde bir iyileşme sağlanırken önerilen yeni yöntemle %19.1 değerinde bir iyileşme sağlanmıştır. Görüldüğü gibi yeni yöntemle sadece iki parametre belirlenerek sekiz parametrenin belirlendiği diğer yöntemden yaklaşık üç kat kadar daha etkili bir cevap elde edilmiştir.

Düzgün seçilmemiş ölçekleme çarpanlarının kullanıldığı durumda ise en iyi yanıt tek bölgeci ortak dilsel biçimlendirici değerleri atanması ile elde edilmiştir. Bu durumda sistem performansı üzerinde %72.4 değerinde bir iyileşme sağlanırken önerilen yeni yöntemle %80.5 değerinde bir iyileşme sağlanmıştır.

Sonuçlardan görüldüğü gibi yeni yöntem, oldukça karmaşık yapıli sistemlerde ve büyük ölü zamanlı doğrusal sistemlerde bile en basit kontrolör yapısının kullanılmasına karşın oldukça etkili bir kontrol sağlamaktadır.

Bu yöntem geliştirilmeye oldukça yatkındır. Her tür bulanık kontrolöre kolayca uygulanabilir. Her bir üyelik fonksiyonu için dilsel biçimlendiriciler atayarak daha çok parametre ile daha hassas sonuçlar elde edilebilir. Ayrıca değişik bir yaklaşımla çıkış üyelik fonksiyonları için de dilsel biçimlendiriciler belirlenebilir. Çalışmada basitlik açısından çıkarım mekanizmasında bulanık kontrolörün basit yapıdaki kural tablosu kullanılmaktadır. Bunun yerine daha özel kurallar seçilerek sistem performansı ve dayanıklılığı artırılabilir. Son olarak doğrusal olmayan sistem örneğinde görüldüğü gibi ölçekleme çarpanları ve dilsel biçimlendiriciler arasındaki özel ilişki göz önünde bulundurularak, ölçekleme çarpanları önceden belirlenmiş bir bulanık kontrolörün üyelik fonksiyonları için uygun k parametreleri belirlemek yerine, tasarımda ölçekleme çarpanları ve kullanılacak k parametrelerinin aynı anda aratılması ile elde edilecek değerlerin kullanılması daha etkili sonuçlar ortaya çıkaracaktır.

KAYNAKLAR

- [1] **Zadeh L. A.**, 1965. Fuzzy sets, *Information and Control*, **8**, 338-353.
- [2] **Elmas, Ç.**, 2003. Bulanık Mantık Denetleyiciler, Seçkin Yayıncılık, Ankara.
- [3] **Mamdani, E. H.**, 1974. Application of fuzzy logic algorithms for control of simple dynamic plant, *Proceedings of the Institute of Electrical Engineering*, **121**, 1585-1588.
- [4] **Passino K. M. and Yurkovich, S.**, 1998. Fuzzy Control, Addison Wesley Longman, Inc., California.
- [5] **Hu, B., Mann, G. K. I., Gosine, R. G.**, 1999. A new methodology for analytical and optimal design of fuzzy PID controllers, *IEEE Transactions on Fuzzy Systems*, **7** (5), 521-539.
- [6] **Casillas, J., Cordon, O., Jesus, M. J., and Herrera, F.**, 2005. Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction, *IEEE Transactions on Fuzzy Systems*, **13**, 13-29.
- [7] **Karr C. L., Gentry E. J.**, 1993. Fuzzy control of pH using genetic algorithms, *IEEE Transactions on Fuzzy Systems*, **1**, 46-53.
- [8] **C. H. Chang and Y. C. Wu**, 1995. The genetic algorithm based tuning method for symmetric membership functions of fuzzy logic control systems, *Proc. IEEE Conf. Industrial Automatic Control Emerging Technological Applications*, Taipei, R.O.C., 22-27 May, pp. 421-428.
- [9] **A. Homaifar and V. E. McCormick**, 1995. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, *IEEE Transactions on Fuzzy Systems*, **3**, 129-139.
- [10] **Liska J., Melsheimer S. S.**, 1994. Complete design of fuzzy logic systems using genetic algorithms, *IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on Fuzzy Systems*, Orlando, Florida, 26-29 June, **2**, 1377-1382.
- [11] **Rojas I., Merelo J. J., Bernier J. L., Prieto A.**, 1997. A new approach to fuzzy controller designing and coding via genetic algorithms, *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, Barcelona, Spain, 1-5 July, **3**, 1505-1510.
- [12] **Zhou Y., Li S., Jin R.**, 2002. A new fuzzy neural network with fast learning algorithm and guaranteed stability for manufacturing process control, *Fuzzy Sets and Systems*, **132**, 201-216.
- [13] **Lin C.T., Lee C.S.G.**, 1991. Neural network based fuzzy logic control and decision systems, *IEEE Trans. on Computers*, **40** (12), 1320-1336.

- [14] **Wu C-H. J., Li C. C.**, 1995. Automatically generated rules and membership functions for a neural fuzzy-based fault classifier, *Proceedings of the 37th Midwest Symposium on Circuits and Systems*, Lafayette, Louisiana, 3-5 Aug., **2**, 1377-1380.
- [15] **Cleave D. V., Rattan K. S.**, 2000. Tuning of fuzzy logic controller using neural network, *Proceeding of the IEEE National Aerospace and Electronics Conference (NAECON)*, Dayton, Ohio, 10-12 Oct, 305-312.
- [16] **Liu J., Liu D., Bai H-Y., Wu P-S.**, 2003. The optimal design of neural fuzzy controller, *IEEE Proceedings of the Second International Conference on Machine Learning and Cybernetics*, Xi'an, 2-5 Nov., **7**, 544-547.
- [17] **Iokibe T.**, 1994. A method for automatic rule and membership function generation by discretionary fuzzy performance function and its application to a practical system, *Proc. 1st Int. Joint Conf. North American Fuzzy Information Processing Society Biannual Conf., Ind. Fuzzy Control Intelligent Systems Conf. NASA Joint Technology Workshop Neural Networks Fuzzy Logic*, San Antonio, TX, 18-21 Dec., 363-364.
- [18] **Krishnapuram R.**, 1994. Generation of membership functions via possibilistic clustering, *Proc. 3th IEEE Int. Conf. Fuzzy Systems*, Orlando, FL, 26-29 June, **2**, 902-908.
- [19] **Kim C. J., Russell B. D.**, 1993. Automatic generation of membership function and fuzzy rule using inductive reasoning, *Proc. 3rd IEEE Int. Conf. Industrial Fuzzy Control Intelligent Systems*, Houston, TX, 1-3 Dec., 93-96.
- [20] **Cano J. C., Nava P. A.**, 2002. A fuzzy method for automatic generation of membership function using fuzzy relations from training examples, *the Annual Meeting of the North American Fuzzy Information Processing Society Proceedings (NAFIPS)*, New Orleans, LA, 27-29 June, 158-162.
- [21] **Sebag M., Schoenauer M.**, 1993. Learning membership functions from examples, *Proceedings of Second International Symposium on Uncertainty Modeling and Analysis*, Maryland, USA, 25-28 April, 169-173.
- [22] **Cerrada M., Aguilar J., Colina E., Titli A.**, 2005. Dynamical membership functions: an approach for adaptive fuzzy modelling, *Fuzzy Sets and Systems*, **152**, 513-533.
- [23] **Woodard S. E., Garg D. P.**, 1999. A numerical optimization approach for tuning fuzzy logic controllers, *IEEE Transactions on systems, Man, and Cybernetics—Part B: Cybernetics*, **29**, 565-569.
- [24] **Sánchez A., Álvarez R., Moctezuma1 J.C., Sánchez S.**, 2006. Clustering and artificial neural networks as a tool to generate membership functions, *16th International Conference on Electronics, Communications and Computers (CONIELECOMP)*, Puebla, Mexico, 27-01 Feb., 32-32.

- [25] **Zadeh, L. A.**, 1973. Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Syst., Man, Cybern.*, SMC-3, 28-44.
- [26] **Liu, B. D., Chen, C.Y., and Tsao, J.Y.**, 2001. Design of adaptive fuzzy logic controller based on linguistic hedge concepts and genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, **31**, 32-53.
- [27] **Novák V.**, 1996. A horizon shifting model of linguistic hedges for approximate reasoning, *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, LA, 8-11 Sept., **1**, 423-427.
- [28] **Huynh V.N., Ho T.B., Nakamori Y.**, 2002. A parametric representation of linguistic hedges in Zadeh's fuzzy logic, *International Journal of Approximate Reasoning*, **30**, 203-223.
- [29] **Cock M. D., Kerre E. E.**, 2004. Fuzzy modifiers based on fuzzy relations, *Information Science*, **160**, 173-199.
- [30] **Marin-Blazquez J.G., Shen Q.**, 2001. Linguistic hedges on trapezoidal fuzzy set: Reviset, *The 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, 2-5 Dec., **1**, 412-415.
- [31] **De S. K., Biswas R., Roy A. R.**, 2000. Some operations on intuitionistic fuzzy sets, *Fuzzy Sets and Systems*, **114**, 477-484.
- [32] **Hongjian Shi, Rabab Ward**, 2001. Expanding the definitions of linguistic hedges, *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, 25-28 July, **5**, 2591-2595.
- [33] **Baykal, N. and Beyan, T.**, 2004. Bulanık Mantık Uzman Sistemler ve Denetleyiciler, Bıçaklar Kitabevi, Ankara.
- [34] **Mohan B. M., Patel A. V.**, 2002. Analytical structures and analysis of the simplest fuzzy PD controllers, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, **32 (2)**, 239-248.
- [35] **Kwok, D.P., Tam, P., Li, C.K., Wang, P.**, 1990. Linguistic PID controllers, *Proceedings of the 11th IFAC World Congress*, Tallinn, Estonia, 13-17 Aug., **7**, 192-197.
- [36] **Li, H.X., Gatland, H.B.**, 1996. Conventional fuzzy control and its enhancement, *IEEE Transactions on Systems Man and Cybernetics - Part B*, **26 (5)**, 791-797.
- [37] **Güzelkaya, M., Eksin, İ., Gürleyen, F.**, 2001. A new methodology for designing a fuzzy logic controller and PI, PD blending mechanism, *Journal of Intelligent and Fuzzy Systems*, **11**, 85-98.
- [38] **Qiao, W.Z., Mizumoto, M.**, 1996. PID type fuzzy controller and parameters adaptive method, *Fuzzy Sets and Systems*, **78**, 23-35.
- [39] **Klir G. J., Yuan B.**, 1995. Fuzzy sets and fuzzy logic, theory and applications, Prentice Hall, New Jersey.

EK 1: Uygulamalarda Kullanılan Bulanık Kontrolör MATLAB Yazılımları

```
%%%%%%%%%%  
% Doğrusal sistem uygulamalarında üyelik fonksiyonları için bir bölgeyi ortak %  
% dilsel biçimlendirici değerlerinin atandığı bulanık kontrolör yazılımı %  
%%%%%%%%%
```

```
function output = fcontroller1(e,de)  
% GA tarafından belirlenen dilsel biçimlendiriciler a ve b değerleri  
load hedge1  
% giriş değişkenleri için normalize uzayın tanımlanması  
if e < -1  
    e = -1;  
elseif e > 1  
    e = 1;  
end  
if de < -1  
    de = -1;  
elseif de > 1  
    de = 1;  
end  
% üyelik fonksiyonlarının tanımlanması  
% giriş değişkenleri üyelik fonksiyonları  
% hata üyelik fonksiyonu  
yNBe = (-e)^a;  
yS1e = (e+1)^b; % S' nin sıfırdan küçük taraftaki bacağı  
yS2e = (-e+1)^b; % S' nin sıfırdan büyük taraftaki bacağı  
yPBe = (e)^a;  
% hatanın türevi üyelik fonksiyonu  
yNBde = (-de)^a;  
yS1de = (de+1)^b;
```

```

yS2de = (-de+1)^b;
yPBde = (de)^a;
% çıkış değişkeni üyelik fonksiyonları
outputNB = -0.8;
outputS = 0;
outputPB = 0.8;
% giriş değerlerine göre aktif olacak bölgelerin tanımlanması
% hata üyelik fonksiyonları için
if e >= -1 & e <= 0
    e1 = yNBde;
    e2 = yS1e;
else
    e1 = yS2e;
    e2 = yPBde;
end
% hatanın türevi üyelik fonksiyonları için
if de >= -1 & de <= 0
    de1 = yNBde;
    de2 = yS1de;
else
    de1 = yS2de;
    de2 = yPBde;
end
% kuralların tanımlanması
if e >= -1 & e <= 0
    if de >= -1 & de <= 0
        k1 = outputNB;
        k2 = outputNB;
        k3 = outputNB;
        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
end

```



```

    end
else
    if de >= -1 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end

% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

%% Doğrusal sistem uygulamalarında üyelik fonksiyonları için iki bölgeyi ortak
 %% dilsel biçimlendirici değerlerinin atandığı bulanık kontrolör yazılımı

```
function output = fcontroller2(e,de)
% GA tarafından belirlenen dilsel biçilendiriciler a, b, c ve d değerleri
load hedge2
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -1
    e = -1;
elseif e > 1
    e = 1;
end
if de < -1
    de = -1;
elseif de > 1
    de = 1;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonlarının iki bölgeyi olarak tanımlanması
% hata üyelik fonksiyonları
yNB = [0.5*(-2*e-1)^a+0.5;0.5*(-2*e)^b];
yS1 = [0.5*(2*e+1)^c+0.5; 0.5*(2*e+2)^d];
yS2 = [0.5*(-2*e+1)^c+0.5; 0.5*(-2*e+2)^d];
yPB = [0.5*(2*e-1)^a+0.5; 0.5*(2*e)^b];
% hatanın türevi üyelik fonksiyonları
yNBde = [0.5*(-2*de-1)^a+0.5;0.5*(-2*de)^b];
yS1de = [0.5*(2*de+1)^c+0.5; 0.5*(2*de+2)^d];
yS2de = [0.5*(-2*de+1)^c+0.5; 0.5*(-2*de+2)^d];
yPBde = [0.5*(2*de-1)^a+0.5; 0.5*(2*de)^b];
% çıkış değişkeni üyelik fonksiyonlarının tanımlanması
outputNB = -0.8;
outputS = 0;
outputPB = 0.8;
```

% giriş değerlerine göre aktif olacak bölgelerin tanımlanması

% hata üyelik fonksiyonları için

if $e \geq -1$ & $e \leq -0.5$

$e1 = yNB_e(1);$

$e2 = yS1_e(2);$

elseif $e > -0.5$ & $e \leq 0$

$e1 = yNB_e(2);$

$e2 = yS1_e(1);$

elseif $e > 0$ & $e \leq 0.5$

$e1 = yS2_e(1);$

$e2 = yPB_e(2);$

else

$e1 = yS2_e(2);$

$e2 = yPB_e(1);$

end

% hatanın türevi üyelik fonksiyonları için

if $de \geq -1$ & $de \leq -0.5$

$de1 = yNB_{de}(1);$

$de2 = yS1_{de}(2);$

elseif $de > -0.5$ & $de \leq 0$

$de1 = yNB_{de}(2);$

$de2 = yS1_{de}(1);$

elseif $de > 0$ & $de \leq 0.5$

$de1 = yS2_{de}(1);$

$de2 = yPB_{de}(2);$

else

$de1 = yS2_{de}(2);$

$de2 = yPB_{de}(1);$

end

% kuralların tanımlanması

if $e \geq -1$ & $e \leq 0$

 if $de \geq -1$ & $de \leq 0$

$k1 = \text{outputNB};$

$k2 = \text{outputNB};$

$k3 = \text{outputNB};$

```

        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
else
    if de >= -1 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

```

%% Doğrusal sistem uygulamalarında üyelik fonksiyonları için dört bölge
% ortak dilsel biçimlendirici değerlerinin atandığı bulanık kontrolör yazılımı
%%

```

```

function output = fcontroller4(e,de)

```

```

% GA tarafından belirlenen dilsel biçimlendiriciler a, b, c, d, z, f, g ve h değerleri

```

```

load hedge4

```

```

% giriş değişkenleri için normalize uzayın tanımlanması

```

```

if e < -1

```

```

    e = -1;

```

```

elseif e > 1

```

```

    e = 1;

```

```

end

```

```

if de < -1

```

```

    de = -1;

```

```

elseif de > 1

```

```

    de = 1;

```

```

end

```

```

% giriş değişkenleri üyelik fonksiyonlarının dört bölge olarak tanımlanması

```

```

% hata üyelik fonksiyonları

```

```

kk = 0.25; % kk = 1/n , n = bölge sayısı

```

```

yNBe = [ kk*(-4*e-3)^a+3/4;

```

```

    kk*(-4*e-2)^b+1/2;

```

```

    kk*(-4*e-1)^c+1/4;

```

```

    kk*(-4*e)^d];

```

```

yS1e = [ kk*(4*e+4)^h;

```

```

    kk*(4*e+3)^g+1/4;

```

```

    kk*(4*e+2)^f+1/2;

```

```

    kk*(4*e+1)^z+3/4];

```

```

yS2e = [ kk*(-4*e+4)^h;

```

```

    kk*(-4*e+3)^g+1/4;

```

```

    kk*(-4*e+2)^f+1/2;

```

```

    kk*(-4*e+1)^z+3/4];

```

```

yPBe = [ kk*(4*e-3)^a+3/4;

```

```

kk*(4*e-2)^b+1/2;
kk*(4*e-1)^c+1/4;
kk*(4*e)^d];
% hatanın türevi üyelik fonksiyonları
yNBde = [ kk*(-4*de-3)^a+3/4;
          kk*(-4*de-2)^b+1/2;
          kk*(-4*de-1)^c+1/4;
          kk*(-4*de)^d];
yS1de = [ kk*(4*de+4)^h;
          kk*(4*de+3)^g+1/4;
          kk*(4*de+2)^f+1/2;
          kk*(4*de+1)^z+3/4];
yS2de = [ kk*(-4*de+4)^h;
          kk*(-4*de+3)^g+1/4;
          kk*(-4*de+2)^f+1/2;
          kk*(-4*de+1)^z+3/4];
yPBde = [ kk*(4*de-3)^a+3/4;
          kk*(4*de-2)^b+1/2;
          kk*(4*de-1)^c+1/4;
          kk*(4*de)^d];
% çıkış değişkeni üyelik fonksiyonlarının tanımlanması
outputNB = -0.8;
outputS = 0;
outputPB = 0.8;
% giriş değerlerine göre aktif olacak bölgelerin tanımlanması
% hata üyelik fonksiyonları için
if e >= -1 & e <= -0.75
    t1e = 1;
elseif e > -0.75 & e <= -0.5
    t1e = 2;
elseif e > -0.5 & e <= -0.25
    t1e = 3;
elseif e > -0.25 & e <= 0
    t1e = 4;
elseif e > 0 & e <= 0.25

```

```

    t2e = 4;
elseif e > 0.25 & e <= 0.5
    t2e = 3;
elseif e > 0.5 & e <= 0.75
    t2e = 2;
else
    t2e = 1;
end
if e >= -1 & e <= 0
    e1 = yNBe(t1e);
    e2 = yS1e(t1e);
else
    e1 = yS2e(t2e);
    e2 = yPBe(t2e);
end
% hatanın türevi üyelik fonksiyonları için
if de >= -1 & de <= -0.75
    t1de = 1;
elseif de > -0.75 & de <= -0.5
    t1de = 2;
elseif de > -0.5 & de <= -0.25
    t1de = 3;
elseif de > -0.25 & de <= 0
    t1de = 4;
elseif de > 0 & de <= 0.25
    t2de = 4;
elseif de > 0.25 & de <= 0.5
    t2de = 3;
elseif de > 0.5 & de <= 0.75
    t2de = 2;
else
    t2de = 1;
end
if de >= -1 & de <= 0
    de1 = yNBde(t1de);

```

```

    de2 = yS1de(t1de);
else
    de1 = yS2de(t2de);
    de2 = yPBde(t2de);
end
% kuralların tanımlanması
if e >= -1 & e <= 0
    if de >= -1 & de <= 0
        k1 = outputNB;
        k2 = outputNB;
        k3 = outputNB;
        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
else
    if de >= -1 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;

```



```
output4 = e2*de2*k4;
```

```
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);
```

```

%% Doğrusal sistem uygulamalarında üyelik fonksiyonları için sekiz bölge
% ortak dilsel biçimlendirici değerlerinin atandığı bulanık kontrolör yazılımı
%%

```

```

function output = fcontroller8(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a1, a2 ..., ve b1, b2 ..., değerleri
load hedge8
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -1
    e = -1;
elseif e > 1
    e = 1;
end
if de < -1
    de = -1;
elseif de > 1
    de = 1;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonlarının sekiz bölge olarak tanımlanması
% hata üyelik fonksiyonları
kk = 0.1250; % kk = 1/n , n = bölge sayısı
yNBe = [ kk*(-8*e-7)^a1+7/8;
    kk*(-8*e-6)^a2+3/4;
    kk*(-8*e-5)^a3+5/8;
    kk*(-8*e-4)^a4+1/2;
    kk*(-8*e-3)^a5+3/8;
    kk*(-8*e-2)^a6+1/4;
    kk*(-8*e-1)^a7+1/8;
    kk*(-8*e)^a8];
yPBe = [ kk*(8*e-7)^a1+7/8;
    kk*(8*e-6)^a2+3/4;
    kk*(8*e-5)^a3+5/8;
    kk*(8*e-4)^a4+1/2;

```

$$\begin{aligned}
&kk*(8*e-3)^{a5+3/8}; \\
&kk*(8*e-2)^{a6+1/4}; \\
&kk*(8*e-1)^{a7+1/8}; \\
&kk*(8*e)^{a8};
\end{aligned}$$

$$yS1e = [kk*(8*e+8)^{b1};$$

$$\begin{aligned}
&kk*(8*e+7)^{b2+1/8}; \\
&kk*(8*e+6)^{b3+1/4}; \\
&kk*(8*e+5)^{b4+3/8}; \\
&kk*(8*e+4)^{b5+1/2}; \\
&kk*(8*e+3)^{b6+5/8}; \\
&kk*(8*e+2)^{b7+3/4}; \\
&kk*(8*e+1)^{b8+7/8}];
\end{aligned}$$

$$yS2e = [kk*(-8*e+8)^{b1};$$

$$\begin{aligned}
&kk*(-8*e+7)^{b2+1/8}; \\
&kk*(-8*e+6)^{b3+1/4}; \\
&kk*(-8*e+5)^{b4+3/8}; \\
&kk*(-8*e+4)^{b5+1/2}; \\
&kk*(-8*e+3)^{b6+5/8}; \\
&kk*(-8*e+2)^{b7+3/4}; \\
&kk*(-8*e+1)^{b8+7/8}];
\end{aligned}$$

% Hatanın türevi üyelik fonksiyonları

$$yNBde = [kk*(-8*de-7)^{a1+7/8};$$

$$\begin{aligned}
&kk*(-8*de-6)^{a2+3/4}; \\
&kk*(-8*de-5)^{a3+5/8}; \\
&kk*(-8*de-4)^{a4+1/2}; \\
&kk*(-8*de-3)^{a5+3/8}; \\
&kk*(-8*de-2)^{a6+1/4}; \\
&kk*(-8*de-1)^{a7+1/8}; \\
&kk*(-8*de)^{a8}];
\end{aligned}$$

$$yPBde = [kk*(8*de-7)^{a1+7/8};$$

$$\begin{aligned}
&kk*(8*de-6)^{a2+3/4}; \\
&kk*(8*de-5)^{a3+5/8}; \\
&kk*(8*de-4)^{a4+1/2}; \\
&kk*(8*de-3)^{a5+3/8}; \\
&kk*(8*de-2)^{a6+1/4};
\end{aligned}$$

```

kk*(8*de-1)^a7+1/8;
kk*(8*de)^a8];
yS1de = [kk*(8*de+8)^b1;
kk*(8*de+7)^b2+1/8;
kk*(8*de+6)^b3+1/4;
kk*(8*de+5)^b4+3/8;
kk*(8*de+4)^b5+1/2;
kk*(8*de+3)^b6+5/8;
kk*(8*de+2)^b7+3/4;
kk*(8*de+1)^b8+7/8];
yS2de = [kk*(-8*de+8)^b1;
kk*(-8*de+7)^b2+1/8;
kk*(-8*de+6)^b3+1/4;
kk*(-8*de+5)^b4+3/8;
kk*(-8*de+4)^b5+1/2;
kk*(-8*de+3)^b6+5/8;
kk*(-8*de+2)^b7+3/4;
kk*(-8*de+1)^b8+7/8];
% çıkış değişkeni üyelik fonksiyonlarının tanımlanması
outputNB = -0.8;
outputS = 0;
outputPB = 0.8;
% giriş değerlerine göre aktif olacak bölgelerin tanımlanması
% hata üyelik fonksiyonları için
if e >= -1 & e <= -0.875
    t1e = 1;
elseif e > -0.875 & e <= -0.75
    t1e = 2;
elseif e > -0.75 & e <= -0.625
    t1e = 3;
elseif e > -0.625 & e <= -0.5
    t1e = 4;
elseif e > -0.5 & e <= -0.375
    t1e = 5;
elseif e > -0.375 & e <= -0.25

```

```

    t1e = 6;
elseif e > -0.25 & e <= -0.125
    t1e = 7;
elseif e > -0.125 & e <= 0
    t1e = 8;
elseif e <= 1 & e >= 0.875
    t2e = 1;
elseif e < 0.875 & e >= 0.75
    t2e = 2;
elseif e < 0.75 & e >= 0.625
    t2e = 3;
elseif e < 0.625 & e >= 0.5
    t2e = 4;
elseif e < 0.5 & e >= 0.375
    t2e = 5;
elseif e < 0.375 & e >= 0.25
    t2e = 6;
elseif e < 0.25 & e >= 0.125
    t2e = 7;
elseif e < 0.125 & e > 0
    t2e = 8;
end
if e >= -1 & e <= 0
    e1 = yNBe(t1e);
    e2 = yS1e(t1e);
else
    e1 = yS2e(t2e);
    e2 = yPBe(t2e);
end
% hatanın türevi üyelik fonksiyonları için
if de >= -1 & de <= -0.875
    t1de = 1;
elseif de > -0.875 & de <= -0.75
    t1de = 2;
elseif de > -0.75 & de <= -0.625

```

```

    t1de= 3;
elseif de > -0.625 & de <= -0.5
    t1de = 4;
elseif de > -0.5 & de <= -0.375
    t1de = 5;
elseif de > -0.375 & de <= -0.25
    t1de = 6;
elseif de > -0.25 & de <= -0.125
    t1de = 7;
elseif de > -0.125 & de <= 0
    t1de = 8;
elseif de <= 1 & de >= 0.875
    t2de = 1;
elseif de < 0.875 & de >= 0.75
    t2de = 2;
elseif de < 0.75 & de >= 0.625
    t2de = 3;
elseif de < 0.625 & de >= 0.5
    t2de = 4;
elseif de < 0.5 & de >= 0.375
    t2de = 5;
elseif de < 0.375 & de >= 0.25
    t2de = 6;
elseif de < 0.25 & de >= 0.125
    t2de = 7;
elseif de < 0.125 & de >0
    t2de = 8;
end
if de >= -1 & de <= 0
    de1 = yNBde(t1de);
    de2 = yS1de(t1de);
else
    de1 = yS2de(t2de);
    de2 = yPBde(t2de);
end

```

```

% kuralların tanımlanması
if e >= -1 & e <= 0
    if de >= -1 & de <= 0
        k1 = outputNB;
        k2 = outputNB;
        k3 = outputNB;
        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
else
    if de >= -1 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end

% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

```

%% Doğrusal sistem uygulamalarında üyelik fonksiyonları için bir bölgeyi farklı
%% dilsel biçimlendirici değerlerinin atandığı bulanık kontrolör yazılımı

```

```

function output = fcontroller1(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a, b, c ve d değerleri
load hedge1
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -1
    e = -1;
elseif e > 1
    e = 1;
end
if de < -1
    de = -1;
elseif de > 1
    de = 1;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonları
% hata üyelik fonksiyonu
yNBe = (-e)^a;
yS1e = (e+1)^b;    % S' nin sıfırdan küçük taraftaki bacağı
yS2e = (-e+1)^b;   % S' nin sıfırdan büyük taraftaki bacağı
yPBe = (e)^a;
% hatanın türevi üyelik fonksiyonu
yNBde = (-de)^c;
yS1de = (de+1)^d;
yS2de = (-de+1)^d;
yPBde = (de)^c;
% çıkış değişkeni üyelik fonksiyonları
outputNB = -0.8;
outputS = 0;
outputPB = 0.8;

```


% giriş değerlerine göre aktif olacak bölgelerin tanımlanması

% hata üyelik fonksiyonları için

if $e \geq -1$ & $e \leq 0$

$e1 = yNB_e$;

$e2 = yS1_e$;

else

$e1 = yS2_e$;

$e2 = yPB_e$;

end

% hatanın türevi üyelik fonksiyonları için

if $de \geq -1$ & $de \leq 0$

$de1 = yNB_{de}$;

$de2 = yS1_{de}$;

else

$de1 = yS2_{de}$;

$de2 = yPB_{de}$;

end

% kuralların tanımlanması

if $e \geq -1$ & $e \leq 0$

 if $de \geq -1$ & $de \leq 0$

$k1 = outputNB$;

$k2 = outputNB$;

$k3 = outputNB$;

$k4 = outputS$;

 else

$k1 = outputNB$;

$k2 = outputS$;

$k3 = outputS$;

$k4 = outputPB$;

 end

else

 if $de \geq -1$ & $de \leq 0$

$k1 = outputNB$;

$k2 = outputS$;

$k3 = outputS$;

```

        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

```

%% Doğrusal sistem uygulamalarında üyelik fonksiyonları için iki bölge farklı
%% dilsel biçimlendirici değerlerinin atandığı bulanık kontrolör yazılımı

```

```

function output = fcontroller2(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a, b, c, d, z, f, g ve h değerleri
load hedge2
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -1
    e = -1;
elseif e > 1
    e = 1;
end
if de < -1
    de = -1;
elseif de > 1
    de = 1;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonlarının iki bölge olarak tanımlanması
% hata üyelik fonksiyonları
yNB = [0.5*(-2*e-1)^a+0.5;0.5*(-2*e)^b];
yS1 = [0.5*(2*e+1)^c+0.5; 0.5*(2*e+2)^d];
yS2 = [0.5*(-2*e+1)^c+0.5; 0.5*(-2*e+2)^d];
yPB = [0.5*(2*e-1)^a+0.5; 0.5*(2*e)^b];
% hatanın türevi üyelik fonksiyonları
yNBde = [0.5*(-2*de-1)^z+0.5;0.5*(-2*de)^f];
yS1de = [0.5*(2*de+1)^g+0.5; 0.5*(2*de+2)^h];
yS2de = [0.5*(-2*de+1)^g+0.5; 0.5*(-2*de+2)^h];
yPBde = [0.5*(2*de-1)^z+0.5; 0.5*(2*de)^f];
% çıkış değişkeni üyelik fonksiyonlarının tanımlanması
outputNB = -0.8;
outputS = 0;
outputPB = 0.8;

```

% giriş değerlerine göre aktif olacak bölgelerin tanımlanması

% hata üyelik fonksiyonları için

if $e \geq -1$ & $e \leq -0.5$

$e1 = yNB_e(1);$

$e2 = yS1_e(2);$

elseif $e > -0.5$ & $e \leq 0$

$e1 = yNB_e(2);$

$e2 = yS1_e(1);$

elseif $e > 0$ & $e \leq 0.5$

$e1 = yS2_e(1);$

$e2 = yPB_e(2);$

else

$e1 = yS2_e(2);$

$e2 = yPB_e(1);$

end

% hatanın türevi üyelik fonksiyonları için

if $de \geq -1$ & $de \leq -0.5$

$de1 = yNB_{de}(1);$

$de2 = yS1_{de}(2);$

elseif $de > -0.5$ & $de \leq 0$

$de1 = yNB_{de}(2);$

$de2 = yS1_{de}(1);$

elseif $de > 0$ & $de \leq 0.5$

$de1 = yS2_{de}(1);$

$de2 = yPB_{de}(2);$

else

$de1 = yS2_{de}(2);$

$de2 = yPB_{de}(1);$

end

% kuralların tanımlanması

if $e \geq -1$ & $e \leq 0$

 if $de \geq -1$ & $de \leq 0$

$k1 = \text{outputNB};$

$k2 = \text{outputNB};$

$k3 = \text{outputNB};$

```

        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
else
    if de >= -1 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

%%
 % Doğrusal olmayan sistem uygulamalarında üyelik fonksiyonları için bir %
 % bölgesi ortak dilsel biçimlendiricilerin atandığı bulanık kontrolör yazılımı %
 %%

```
function output = fcontroller1(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a ve b değerleri
load hedge1
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -0.6
    e = -0.6;
elseif e > 0.6
    e = 0.6;
end
if de < -6
    de = -6;
elseif de > 6
    de = 6;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonları
% hata üyelik fonksiyonları
yNB = (-1.66666666*e)^a;
yS1 = (1.66666666*e+1)^b; %S' nin sıfırdan küçük taraftaki bacağı
yS2 = (-1.66666666*e+1)^b; %S' nin sıfırdan büyük taraftaki bacağı
yPB = (1.66666666*e)^a;
% hatanın türevi üyelik fonksiyonları
yNBde = (-0.166666666*de)^a;
yS1de = (0.166666666*de+1)^b;
yS2de = (-0.166666666*de+1)^b;
yPBde = (0.166666666*de)^a;
% çıkış değişkeni üyelik fonksiyonları
outputNB = -0.63;
outputS = 0;
outputPB = 0.63;
```

% giriş değerlerine göre aktif olacak bölgelerin tanımlanması

% hata üyelik fonksiyonları için

if $e \geq -0.6$ & $e \leq 0$

$e1 = yNB_e$;

$e2 = yS1_e$;

else

$e1 = yS2_e$;

$e2 = yPB_e$;

end

% hatanın türevi üyelik fonksiyonları için

if $de \geq -6$ & $de \leq 0$

$de1 = yNB_{de}$;

$de2 = yS1_{de}$;

else

$de1 = yS2_{de}$;

$de2 = yPB_{de}$;

end

% kuralların tanımlanması

if $e \geq -0.6$ & $e \leq 0$

 if $de \geq -6$ & $de \leq 0$

$k1 = outputNB$;

$k2 = outputNB$;

$k3 = outputNB$;

$k4 = outputS$;

 else

$k1 = outputNB$;

$k2 = outputS$;

$k3 = outputS$;

$k4 = outputPB$;

 end

else

 if $de \geq -6$ & $de \leq 0$

$k1 = outputNB$;

$k2 = outputS$;

$k3 = outputS$;

```

        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```


%%
 % Doğrusal olmayan sistem uygulamalarında üyelik fonksiyonları için iki %
 % bölgesi ortak dilsel biçimlendiricilerin atandığı bulanık kontrolör yazılımı %
 %%

```
function output = fcontroller2(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a, b, c ve d değerleri
load hedge2
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -0.6
    e = -0.6;
elseif e > 0.6
    e = 0.6;
end
if de < -6
    de = -6;
elseif de > 6
    de = 6;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonlarının iki bölgesi olarak tanımlanması
% hata üyelik fonksiyonları
yNB=[0.5*(-3.3333333*e-1)^a+0.5;0.5*(-3.3333333*e)^b];
yS1=[0.5*(3.3333333*e+1)^c+0.5; 0.5*(3.3333333*e+2)^d];
yS2=[0.5*(-3.3333333*e+1)^c+0.5; 0.5*(-3.3333333*e+2)^d];
yPB=[0.5*(3.3333333*e-1)^a+0.5; 0.5*(3.3333333*e)^b];
% hatanın türevi üyelik fonksiyonları
yNBde=[0.5*(-0.3333333*de-1)^a+0.5;0.5*(-0.3333333*de)^b];
yS1de=[0.5*(0.3333333*de+1)^c+0.5; 0.5*(0.3333333*de+2)^d];
yS2de=[0.5*(-0.3333333*de+1)^c+0.5; 0.5*(-0.3333333*de+2)^d];
yPBde=[0.5*(0.3333333*de-1)^a+0.5; 0.5*(0.3333333333333333*de)^b];
% çıkış değişkeni üyelik fonksiyonları
outputNB = -0.63;
outputS = 0;
outputPB = 0.63;
```

% giriş değerlerine göre aktif olacak bölgelerin tanımlanması

% hata üyelik fonksiyonları için

if $e \geq -0.6$ & $e \leq -0.3$

$e1 = yNBe(1);$

$e2 = yS1e(2);$

elseif $e > -0.3$ & $e \leq 0$

$e1 = yNBe(2);$

$e2 = yS1e(1);$

elseif $e > 0$ & $e \leq 0.3$

$e1 = yS2e(1);$

$e2 = yPBe(2);$

else

$e1 = yS2e(2);$

$e2 = yPBe(1);$

end

% hatanın türevi üyelik fonksiyonları için

if $de \geq -6$ & $de \leq -3$

$de1 = yNBde(1);$

$de2 = yS1de(2);$

elseif $de > -3$ & $de \leq 0$

$de1 = yNBde(2);$

$de2 = yS1de(1);$

elseif $de > 0$ & $de \leq 3$

$de1 = yS2de(1);$

$de2 = yPBde(2);$

else

$de1 = yS2de(2);$

$de2 = yPBde(1);$

end

% kuralların tanımlanması

if $e \geq -0.6$ & $e \leq 0$

 if $de \geq -6$ & $de \leq 0$

$k1 = outputNB;$

$k2 = outputNB;$

$k3 = outputNB;$

```

        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
else
    if de >= -6 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

```

%% Doğrusal olmayan sistem uygulamalarında üyelik fonksiyonları için dört
%% bölgeyi ortak dilsel biçimlendiricilerin atandığı bulanık kontrolör yazılımı

```

```

function output = fcontroller4(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a1, a2 ..., ve b1, b2 ..., değerleri
load hedge4
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -0.6
    e = -0.6;
elseif e > 0.6
    e = 0.6;
end
if de < -6
    de = -6;
elseif de > 6
    de = 6;
end
% giriş değişkenleri üyelik fonksiyonlarının dört bölgeyi olarak tanımlanması
% hata üyelik fonksiyonları
kk = 0.25; % kk = 1/n , n = bölge sayısı
yNBe = [ kk*(-6.6666667*e-3)^a1+3/4;
    kk*(-6.6666667*e-2)^a2+1/2;
    kk*(-6.6666667*e-1)^a3+1/4;
    kk*(-6.6666667*e)^a4];
yS1e = [ kk*(6.6666667*e+4)^b1;
    kk*(6.6666667*e+3)^b2+1/4;
    kk*(6.6666667*e+2)^b3+1/2;
    kk*(6.6666667*e+1)^b4+3/4];
yS2e = [ kk*(-6.6666667*e+4)^b1;
    kk*(-6.6666667*e+3)^b2+1/4;
    kk*(-6.6666667*e+2)^b3+1/2;
    kk*(-6.6666667*e+1)^b4+3/4];
yPBe = [ kk*(6.6666667*e-3)^a1+3/4;

```

```

kk*(6.66666667*e-2)^a2+1/2;
kk*(6.66666667*e-1)^a3+1/4;
kk*(6.66666667*e)^a4];
% hata türevi üyelik fonksiyonları
yNBde = [ kk*(-0.66666667*de-3)^a1+3/4;
          kk*(-0.66666667*de-2)^a2+1/2;
          kk*(-0.66666667*de-1)^a3+1/4;
          kk*(-0.66666667*de)^a4];
yS1de = [ kk*(0.66666667*de+4)^b1;
          kk*(0.66666667*de+3)^b2+1/4;
          kk*(0.66666667*de+2)^b3+1/2;
          kk*(0.66666667*de+1)^b4+3/4];
yS2de = [ kk*(-0.66666667*de+4)^b1;
          kk*(-0.66666667*de+3)^b2+1/4;
          kk*(-0.66666667*de+2)^b3+1/2;
          kk*(-0.66666667*de+1)^b4+3/4];
yPBde = [ kk*(0.66666667*de-3)^a1+3/4;
          kk*(0.66666667*de-2)^a2+1/2;
          kk*(0.66666667*de-1)^a3+1/4;
          kk*(0.66666667*de)^a4];
% çıkış değişkeni üyelik fonksiyonları
outputNB = -0.63;
outputS = 0;
outputPB = 0.63;
% giriş değerlerine göre aktif olacak bölgelerin tanımlanması
% hata üyelik fonksiyonları için
if e >= -0.6 & e <= -0.45
    t1e = 1;
elseif e > -0.45 & e <= -0.3
    t1e = 2;
elseif e > -0.3 & e <= -0.15
    t1e = 3;
elseif e > -0.15 & e <= 0
    t1e = 4;
elseif e > 0 & e <= 0.15

```

```

    t2e = 4;
elseif e > 0.15 & e <= 0.3
    t2e = 3;
elseif e > 0.3 & e <= 0.45
    t2e = 2;
else
    t2e = 1;
end
if e >= -0.6 & e <= 0
    e1 = yNBe(t1e);
    e2 = yS1e(t1e);
else
    e1 = yS2e(t2e);
    e2 = yPBe(t2e);
end
% hatanın türevi üyelik fonksiyonları için
if de >= -6 & de <= -4.5
    t1de = 1;
elseif de > -4.5 & de <= -3
    t1de = 2;
elseif de > -3 & de <= -1.5
    t1de = 3;
elseif de > -1.5 & de <= 0
    t1de = 4;
elseif de > 0 & de <= 1.5
    t2de = 4;
elseif de > 1.5 & de <= 3
    t2de = 3;
elseif de > 3 & de <= 4.5
    t2de = 2;
else
    t2de = 1;
end
if de >= -6 & de <= 0
    de1 = yNBde(t1de);

```

```

    de2 = yS1de(t1de);
else
    de1 = yS2de(t2de);
    de2 = yPBde(t2de);
end
% kuralların tanımlanması
if e >= -0.6 & e <= 0
    if de >= -6 & de <= 0
        k1 = outputNB;
        k2 = outputNB;
        k3 = outputNB;
        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
else
    if de >= -6 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;

```

```
output4 = e2*de2*k4;
```

```
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);
```



```

%% Doğrusal olmayan sistem uygulamalarında üyelik fonksiyonları için sekiz
%% bölgesi ortak dilsel biçimlendiricilerin atandığı bulanık kontrolör yazılımı

```

```

function output = fcontroller8(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a1, a2 ..., ve b1, b2 ..., değerleri
load hedge8
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -0.6
    e = -0.6;
elseif e > 0.6
    e = 0.6;
end
if de < -6
    de = -6;
elseif de > 6
    de = 6;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonlarının sekiz bölgesi olarak tanımlanması
% hata üyelik fonksiyonları
kk = 0.1250; % kk = 1/n , n = bölge sayısı
yNBe = [ kk*(-13.33333*e-7)^a1+7/8;
    kk*(-13.33333*e-6)^a2+3/4;
    kk*(-13.33333*e-5)^a3+5/8;
    kk*(-13.33333*e-4)^a4+1/2;
    kk*(-13.33333*e-3)^a5+3/8;
    kk*(-13.33333*e-2)^a6+1/4;
    kk*(-13.33333*e-1)^a7+1/8;
    kk*(-13.33333*e)^a8];
yPBe = [ kk*(13.33333*e-7)^a1+7/8;
    kk*(13.33333*e-6)^a2+3/4;
    kk*(13.33333*e-5)^a3+5/8;
    kk*(13.33333*e-4)^a4+1/2;

```

```

kk*(13.333333*e-3)^a5+3/8;
kk*(13.333333*e-2)^a6+1/4;
kk*(13.333333*e-1)^a7+1/8;
kk*(13.333333*e)^a8];
yS1e = [kk*(13.333333*e+8)^b1;
kk*(13.333333*e+7)^b2+1/8;
kk*(13.333333*e+6)^b3+1/4;
kk*(13.333333*e+5)^b4+3/8;
kk*(13.333333*e+4)^b5+1/2;
kk*(13.333333*e+3)^b6+5/8;
kk*(13.333333*e+2)^b7+3/4;
kk*(13.333333*e+1)^b8+7/8];
yS2e = [kk*(-13.333333*e+8)^b1;
kk*(-13.333333*e+7)^b2+1/8;
kk*(-13.333333*e+6)^b3+1/4;
kk*(-13.333333*e+5)^b4+3/8;
kk*(-13.333333*e+4)^b5+1/2;
kk*(-13.333333*e+3)^b6+5/8;
kk*(-13.333333*e+2)^b7+3/4;
kk*(-13.333333*e+1)^b8+7/8];
% hata türevi üyelik fonksiyonları
yNBde = [ kk*(-1.3333333*de-7)^a1+7/8;
kk*(-1.3333333*de-6)^a2+3/4;
kk*(-1.3333333*de-5)^a3+5/8;
kk*(-1.3333333*de-4)^a4+1/2;
kk*(-1.3333333*de-3)^a5+3/8;
kk*(-1.3333333*de-2)^a6+1/4;
kk*(-1.3333333*de-1)^a7+1/8;
kk*(-1.3333333*de)^a8];
yPBde = [ kk*(1.3333333*de-7)^a1+7/8;
kk*(1.3333333*de-6)^a2+3/4;
kk*(1.3333333*de-5)^a3+5/8;
kk*(1.3333333*de-4)^a4+1/2;
kk*(1.3333333*de-3)^a5+3/8;
kk*(1.3333333*de-2)^a6+1/4;

```

```

kk*(1.3333333*de-1)^a7+1/8;
kk*(1.3333333*de)^a8];
yS1de = [kk*(1.3333333*de+8)^b1;
kk*(1.3333333*de+7)^b2+1/8;
kk*(1.3333333*de+6)^b3+1/4;
kk*(1.3333333*de+5)^b4+3/8;
kk*(1.3333333*de+4)^b5+1/2;
kk*(1.3333333*de+3)^b6+5/8;
kk*(1.3333333*de+2)^b7+3/4;
kk*(1.3333333*de+1)^b8+7/8];
yS2de = [kk*(-1.3333333*de+8)^b1;
kk*(-1.3333333*de+7)^b2+1/8;
kk*(-1.3333333*de+6)^b3+1/4;
kk*(-1.3333333*de+5)^b4+3/8;
kk*(-1.3333333*de+4)^b5+1/2;
kk*(-1.3333333*de+3)^b6+5/8;
kk*(-1.3333333*de+2)^b7+3/4;
kk*(-1.3333333*de+1)^b8+7/8];
% çıkış değişkeni üyelik fonksiyonları
outputNB = -0.63;
outputS = 0;
outputPB = 0.63;
% giriş değerlerine göre aktif olacak bölgelerin tanımlanması
% hata üyelik fonksiyonları için
if e >= -0.6 & e <= -0.525
    t1e = 1;
elseif e > -0.525 & e <= -0.45
    t1e = 2;
elseif e > -0.45 & e <= -0.375
    t1e = 3;
elseif e > -0.375 & e <= -0.3
    t1e = 4;
elseif e > -0.3 & e <= -0.225
    t1e = 5;
elseif e > -0.225 & e <= -0.15

```

```

    t1e = 6;
elseif e > -0.15 & e <= -0.075
    t1e = 7;
elseif e > -0.075 & e <= 0
    t1e = 8;
elseif e <= 0.6 & e >= 0.525
    t2e = 1;
elseif e < 0.525 & e >= 0.45
    t2e = 2;
elseif e < 0.45 & e >= 0.375
    t2e = 3;
elseif e < 0.375 & e >= 0.3
    t2e = 4;
elseif e < 0.3 & e >= 0.225
    t2e = 5;
elseif e < 0.225 & e >= 0.15
    t2e = 6;
elseif e < 0.15 & e >= 0.075
    t2e = 7;
elseif e < 0.075 & e > 0
    t2e = 8;
end
if e >= -0.6 & e <= 0
    e1 = yNBe(t1e);
    e2 = yS1e(t1e);
else
    e1 = yS2e(t2e);
    e2 = yPBe(t2e);
end
% hatanın türevi üyelik fonksiyonları için
if de >= -6 & de <= -5.25
    t1de = 1;
elseif de > -5.25 & de <= -4.5
    t1de = 2;
elseif de > -4.5 & de <= -3.75

```

```

    t1de = 3;
elseif de > -3.75 & de <= -3
    t1de = 4;
elseif de > -3 & de <= -2.25
    t1de = 5;
elseif de > -2.25 & de <= -1.5
    t1de = 6;
elseif de > -1.5 & de <= -0.75
    t1de = 7;
elseif de > -0.75 & de <= 0
    t1de = 8;
elseif de <= 6 & de >= 5.25
    t2de = 1;
elseif de < 5.25 & de >= 4.5
    t2de = 2;
elseif de < 4.5 & de >= 3.75
    t2de = 3;
elseif de < 3.75 & de >= 3
    t2de = 4;
elseif de < 3 & de >= 2.25
    t2de = 5;
elseif de < 2.25 & de >= 1.5
    t2de = 6;
elseif de < 1.5 & de >= 0.75
    t2de = 7;
elseif de < 0.75 & de > 0
    t2de = 8;
end
if de >= -6 & de <= 0
    de1 = yNBde(t1de);
    de2 = yS1de(t1de);
else
    de1 = yS2de(t2de);
    de2 = yPBde(t2de);
end

```

```

% kuralların tanımlanması
if e >= -0.6 & e <= 0
    if de >= -6 & de <= 0
        k1 = outputNB;
        k2 = outputNB;
        k3 = outputNB;
        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
else
    if de >= -6 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end

% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

%%
 % Doğrusal olmayan sistem uygulamalarında üyelik fonksiyonları için bir %
 % bölgeyi farklı dilsel biçimlendiricilerin atandığı bulanık kontrolör yazılımı %
 %%

```
function output = fcontroller1(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a, b, c ve d değerleri
load hedge1
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -0.6
    e = -0.6;
elseif e > 0.6
    e = 0.6;
end
if de < -6
    de = -6;
elseif de > 6
    de = 6;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonları
% hata üyelik fonksiyonları
yNBe = (-1.66666666*e)^a;
yS1e = (1.66666666*e+1)^b; %S' nin sıfırdan küçük taraftaki bacağı
yS2e = (-1.66666666*e+1)^b; %S' nin sıfırdan büyük taraftaki bacağı
yPBe = (1.66666666*e)^a;
% hatanın türevi üyelik fonksiyonları
yNBde = (-0.166666666*de)^c;
yS1de = (0.166666666*de+1)^d;
yS2de = (-0.166666666*de+1)^d;
yPBde = (0.166666666*de)^c;
% çıkış değişkeni üyelik fonksiyonları
outputNB = -0.63;
outputS = 0;
outputPB = 0.63;
```

% giriş değerlerine göre aktif olacak bölgelerin tanımlanması

% hata üyelik fonksiyonları için

if $e \geq -0.6$ & $e \leq 0$

$e1 = yNB_e$;

$e2 = yS1_e$;

else

$e1 = yS2_e$;

$e2 = yPB_e$;

end

% hatanın türevi üyelik fonksiyonları için

if $de \geq -6$ & $de \leq 0$

$de1 = yNB_{de}$;

$de2 = yS1_{de}$;

else

$de1 = yS2_{de}$;

$de2 = yPB_{de}$;

end

% kuralların tanımlanması

if $e \geq -0.6$ & $e \leq 0$

 if $de \geq -6$ & $de \leq 0$

$k1 = outputNB$;

$k2 = outputNB$;

$k3 = outputNB$;

$k4 = outputS$;

 else

$k1 = outputNB$;

$k2 = outputS$;

$k3 = outputS$;

$k4 = outputPB$;

 end

else

 if $de \geq -6$ & $de \leq 0$

$k1 = outputNB$;

$k2 = outputS$;

$k3 = outputS$;


```

        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış değeri
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

%%
 % Doğrusal olmayan sistem uygulamalarında üyelik fonksiyonları için iki %
 % bölge farklı dilsel biçimlendiricilerin atandığı bulanık kontrolör yazılımı %
 %%

```
function output = fcontroller2(e,de)
% GA tarafından belirlenen dilsel biçimlendiriciler a, b, c, d, z, f, g ve h değerleri
load hedge2
% giriş değişkenleri için normalize uzayın tanımlanması
if e < -0.6
    e = -0.6;
elseif e > 0.6
    e = 0.6;
end
if de < -6
    de = -6;
elseif de > 6
    de = 6;
end
% üyelik fonksiyonlarının tanımlanması
% giriş değişkenleri üyelik fonksiyonlarının iki bölgesi olarak tanımlanması
% hata üyelik fonksiyonları
yNB=[0.5*(-3.3333333*e-1)^a+0.5;0.5*(-3.3333333*e)^b];
yS1=[0.5*(3.3333333*e+1)^c+0.5; 0.5*(3.3333333*e+2)^d];
yS2=[0.5*(-3.3333333*e+1)^c+0.5; 0.5*(-3.3333333*e+2)^d];
yPB=[0.5*(3.3333333*e-1)^a+0.5; 0.5*(3.3333333*e)^b];
% hatanın türevi üyelik fonksiyonları
yNBde=[0.5*(-0.3333333*de-1)^z+0.5;0.5*(-0.3333333*de)^f];
yS1de=[0.5*(0.3333333*de+1)^g+0.5; 0.5*(0.3333333*de+2)^h];
yS2de=[0.5*(-0.3333333*de+1)^g+0.5; 0.5*(-0.3333333*de+2)^h];
yPBde=[0.5*(0.3333333*de-1)^z+0.5; 0.5*(0.333333333*de)^f];
% çıkış değişkeni üyelik fonksiyonları
outputNB = -0.63;
outputS = 0;
outputPB = 0.63;
```

% giriş değerlerine göre aktif olacak bölgelerin tanımlanması

% hata üyelik fonksiyonları için

if $e \geq -0.6$ & $e \leq -0.3$

$e1 = yNB_e(1);$

$e2 = yS1_e(2);$

elseif $e > -0.3$ & $e \leq 0$

$e1 = yNB_e(2);$

$e2 = yS1_e(1);$

elseif $e > 0$ & $e \leq 0.3$

$e1 = yS2_e(1);$

$e2 = yPB_e(2);$

else

$e1 = yS2_e(2);$

$e2 = yPB_e(1);$

end

% hatanın türevi üyelik fonksiyonları için

if $de \geq -6$ & $de \leq -3$

$de1 = yNB_{de}(1);$

$de2 = yS1_{de}(2);$

elseif $de > -3$ & $de \leq 0$

$de1 = yNB_{de}(2);$

$de2 = yS1_{de}(1);$

elseif $de > 0$ & $de \leq 3$

$de1 = yS2_{de}(1);$

$de2 = yPB_{de}(2);$

else

$de1 = yS2_{de}(2);$

$de2 = yPB_{de}(1);$

end

% kuralların tanımlanması

if $e \geq -0.6$ & $e \leq 0$

 if $de \geq -6$ & $de \leq 0$

$k1 = \text{outputNB};$

$k2 = \text{outputNB};$

$k3 = \text{outputNB};$

```

        k4 = outputS;
    else
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    end
else
    if de >= -6 & de <= 0
        k1 = outputNB;
        k2 = outputS;
        k3 = outputS;
        k4 = outputPB;
    else
        k1 = outputS;
        k2 = outputPB;
        k3 = outputPB;
        k4 = outputPB;
    end
end
end
% ağırlık merkezi durulama yöntemi ile elde edilen keskin çıkış
output1 = e1*de1*k1;
output2 = e1*de2*k2;
output3 = e2*de1*k3;
output4 = e2*de2*k4;
output = (output1+output2+output3+output4)/(e1*de1+e1*de2+e2*de1+e2*de2);

```

ÖZGEÇMİŞ

Cenk Ulu 1981 yılında Yozgat'ta doğdu. Lise öğrenimini 1999 yılında Orhan Cemal Fersoy Lisesinde tamamladı ve aynı yıl Yıldız Teknik Üniversitesi Elektrik Mühendisliği bölümüne kayıt oldu. Bu üniversiteden 2004 yılında mezun olarak İstanbul Teknik Üniversitesi Kontrol ve Otomasyon Mühendisliği Yüksek Lisans Programında öğrenimine devam etti. Halen bu bölümde öğrenciliğini sürdürmektedir. İlgi duyduğu alanlar bulanık mantık kontrolörler, uyarlamalı kontrol teknikleri ve genetik algoritmalarıdır.