

151281

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BULANIK PID TİPİ KONTROL ALGORİTMALARI

151281

YÜKSEK LİSANS TEZİ
Müh. Onur KARASAKAL
(504011134)

Tezin Enstitüye Verildiği Tarih : 22 Kasım 2003
Tezin Savunulduğu Tarih : 14 Ocak 2004

Tez Danışmanı : Prof.Dr. Müjde GÜZELKAYA
Diğer Jüri Üyeleri: Prof.Dr. İbrahim EKSİN (İTÜ)
Y.Doç.Dr. Neslihan ŞENGÖR (İTÜ)

M. Güzelkaya
İbrahim Eksin
N. Şengör

OCAK 2004

ÖNSÖZ

Öncelikle bana güvenip danışmanlığımı kabul etmesi ve tez çalışması boyunca her zaman, her konuda destek olması sebebiyle değerli hocam Doç. Dr. Müjde GÜZELKAYA' ya, her türlü yardımı esirgemeyen ve özellikle samimiyetinden dolayı hocam Prof. Dr. İbrahim EKSİN' e, bilgilerini her zaman benimle paylaşan Araştırma Görevlisi Engin YEŞİL ve Tolga HASDEMİR ile Dr. Murat AKIN' a tüm hayatım boyunca bana verdikleri destek ve bana karşı duydukları güvenden ötürü aileme, yüksek lisans çalışmam esnasında desteğinden, ilgisinden ve sabrından dolayı sevgili eşim Çağla KARASAKAL' a teşekkürü bir borç bilirim.

Kasım 2003

Onur KARASAKAL



İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	xi
ÖZET	xii
SUMMARY	xiv
1. GİRİŞ	1
2. SİSTEME GENEL BAKIŞ VE SİSTEM MODELLEME	4
2.1. Sisteme Genel Bakış	4
2.2. Sistem Modellerinin Basamak Yanıtı Yöntemi İle Elde Edilmesi	6
2.3. PCS 327 Proses Kontrol Simülatörünün Modellenmesi	8
2.3.1. PCS 327 Proses Simülatörünün Üç Parametre İle Modellenmesi	8
2.3.2. PCS 327 Proses Simülatörünün Dört Parametre İle Modellenmesi	11
3. PID KONTROLÖRÜ TASARIMI	13
3.1. PID Kontrol Algoritması	13
3.2. Üç Parametrelili Sistem Modeli İçin PID Kontrolörü Tasarımı	15
3.2.1. Ölü Zamansız Model İçin PID Kontrolörü Tasarımı	15
3.2.2. Ölü Zamanlı Model İçin PID Kontrolörü Tasarımı	17
3.3. Dört Parametrelili Sistem Modeli İçin PID Kontrolörü Tasarımı	19
3.4. Üç Ve Dört Parametrelili Sistem Modelleri İçin Tasarlanan PID Kontrolör Sonuçlarının İrdelenmesi	21
4. BULANIK KONTROLÖR TASARIMI	22
4.1. Bulanık PID Tipi Kontrolör Yapıları	22
4.1.1. Özayarlamaşız Bulanık PID Tipi Kontrolörler	24
4.1.2. Özayarlamaşlı Bulanık PID Tipi Kontrolörler	28
4.1.2.1. Aşım Gözleyicisi İle Parametre Uyarlamalı Yöntem	29
4.1.2.2. Fonksiyon Ayarlayıcısı İle Parametre Uyarlamalı Yöntem	32
4.1.2.3. Bağlı Hız Gözleyicisi İle Parametre Uyarlamalı Yöntem	34
4.2. Özayarlamaşız Bulanık PID Tipi Kontrolör Tasarımı	39

4.2.1. Üç Parametrelili Ölü Zamansız Modele Uygun Olarak Tasarlanan Bulanık PID Tipi Kontrolörün Uygulanması	42
4.2.2. Üç Parametrelili Ölü Zamanlı Modele Uygun Olarak Tasarlanan Bulanık PID Tipi Kontrolörün Uygulanması	43
4.2.3. Dört Parametrelili Modele Uygun Olarak Tasarlanan Bulanık PID Tipi Kontrolörün Uygulanması	44
4.2.4. Üç Ve Dört Parametrelili Sistem Modelleri İçin Tasarlanan Bulanık PID Tipi Kontrolör Sonuçlarının İrdelenmesi	46
4.3. Özayarlama Bulanık PID Tipi Kontrolör Tasarımı	47
4.3.1. Aşım Gözleyicisi İle Parametre Uyarlamalı Yöntem	47
4.3.1.1. Aşım Gözleyicisi Yönteminin Ölü Zamansız Sisteme Uygulanması	47
4.3.1.2. Aşım Gözleyicisi Yönteminin Ölü Zamanlı Sisteme Uygulanması	51
4.3.2. Fonksiyon Ayarlayıcısı İle Parametre Uyarlamalı Yöntem	54
4.3.2.1. Fonksiyon Ayarlayıcısı Yönteminin Ölü Zamansız Sisteme Uygulanması	54
4.3.2.2. Fonksiyon Ayarlayıcısı Yönteminin Ölü Zamanlı Sisteme Uygulanması	57
4.3.3. Bağlı Hız Gözleyicisi İle Parametre Uyarlamalı Yöntem	60
4.3.3.1. Bağlı Hız Gözleyicisi Yönteminin Ölü Zamansız Sisteme Uygulanması	60
4.3.3.2. Bağlı Hız Gözleyicisi Yönteminin Ölü Zamanlı Sisteme Uygulanması	62
4.3.4. Dört Parametrelili Sistem Modeli İçin Tasarlanan Bulanık PID Tipi Kontrolör Sonuçlarının İrdelenmesi	65
4.3.4.1. Aşım Gözleyicisi Yöntemi İle Yapılan Özayarlamanın Etkileri	65
4.3.4.2. Fonksiyon Ayarlayıcısı Yöntemi İle Yapılan Özayarlamanın Etkileri	67
4.3.4.3. Bağlı Hız Gözleyicisi Yöntemi İle Yapılan Özayarlamanın Etkileri	70
4.3.4.4. Özayarlama Yöntemlerinin Karşılaştırılması	73
5. SONUÇLAR VE TARTIŞMA	76
KAYNAKLAR	79
EKLER	81
ÖZGEÇMİŞ	103

KISALTMALAR

PID	: Proportional, Integral and Derivative Controller
PCS	: Process Control Simulator
PLC	: Programmable Logic Controller
FLC	: Fuzzy Logic Controller
PD	: Proportional and Derivative Controller
PI	: Proportional and Integral Controller
ADC	: Analog to Digital Convertor
DAC	: Digital to Analog Convertor
SF	: Scaling Factor
MF	: Membership Function
BHG	: Bağlı hız gözleyicisi



TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1	De(k) ve dde(k) işaretleri arasındaki ilişki ve sistem yanıtı yapısı .. 36
Tablo 4.2	Çıkış değişkeni (δ) için kural tabanı 38
Tablo 4.3	Bulanık kontrol kural tablosu 40



ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : Deney setinin genel görünümü	4
Şekil 2.2 : Basamak cevaplı sistemlerin üç parametrelili modellerinin grafiksel olarak belirlenmesi	7
Şekil 2.3 : Ölü zamansız sistem için basamak işareti yanıtı	8
Şekil 2.4 : Ölü zamansız sistem için üç parametrelili modele ait parametrelerin belirlenmesi	9
Şekil 2.5 : Ölü zamanlı sistem için basamak işareti yanıtı	10
Şekil 2.6 : Ölü zamanlı sistem için üç parametrelili modele ait parametrelerin belirlenmesi	10
Şekil 3.1 : Standart bir PID kontrolörünün blok gösterimi	13
Şekil 3.2 : Ayrık PID kontrolörü blok gösterimi	14
Şekil 3.3 : Sayısal kontrolöre sahip bir kontrol sisteminin blok gösterimi	15
Şekil 3.4 : Ölü zamansız model ve gerçek sistem için tasarlanan PID kontrolörüne ilişkin sistem ve simülasyon cevapları	16
Şekil 3.5 : Ölü zamansız model ve gerçek sistem için tasarlanan PID kontrolörüne ait kontrol işaretleri	16
Şekil 3.6 : Ölü zamanlı model ve gerçek sistem için tasarlanan PID kontrolörüne ilişkin gerçek sistem ve simülasyon cevapları	18
Şekil 3.7 : Ölü zamanlı model ve gerçek sistem için tasarlanan PID kontrolörüne ait kontrol işaretleri	18
Şekil 3.8 : Dört parametrelili model ve gerçek sistem için tasarlanan PID kontrolörüne ilişkin sistem ve simülasyon cevapları	19
Şekil 3.9 : Dört parametrelili model ve gerçek sistem için tasarlanan PID kontrolörüne ait kontrol işaretleri	20
Şekil 4.1 : Bulanık PD tipi kontrolöre ait blok gösterimi.....	24
Şekil 4.2 : Girişlerin hata ve hatanın integrali olarak alınması durumunda elde edilen bulanık kontrolöre ait blok gösterimi.....	25
Şekil 4.3 : Bulanık PI tipi kontrolöre ait blok gösterimi.....	25
Şekil 4.4 : Girişlerin hata, hatanın değişimi ve hatanın integrali olarak alınması durumunda elde edilen bulanık kontrolöre ait blok gösterimi.....	26

Şekil 4.5	: Hata ve hatanın değişimi olmak üzere iki adet giriş ve iki ayrı kural tabanına sahip bulanık kontrolöre ait blok gösterimi.....	27
Şekil 4.6	: Bulanık PID tipi kontrolöre ait blok gösterimi.....	27
Şekil 4.7	: Bir kontrol sistemi basamak cevabının değişik evreleri	29
Şekil 4.8	: Aşım gözleyicisi ile parametre uyarlamalı bulanık PID tipi kontrolörün kapalı çevrim kontrol yapısı	31
Şekil 4.9	: Fonksiyon ayarlayıcısı ile parametre uyarlamalı bulanık PID tipi kontrolörün kapalı çevrim kontrol yapısı	33
Şekil 4.10	: Basamak girişi için sistem yanıtlarına ait bağlı hızların gösterimi .	35
Şekil 4.11	: Bağlı hız gözleyicisi ile parametre uyarlamalı bulanık PID tipi kontrolörün kapalı çevrim kontrol yapısı	37
Şekil 4.12	: Çıkış değişkeni (δ) için kural tabanı görünümü	39
Şekil 4.13	: Bulanık kontrol için üyelik fonksiyonları	39
Şekil 4.14	: Tasarlanan bulanık PID tipi kontrolöre ait sistem yanıtları	42
Şekil 4.15	: Tasarlanan bulanık PID tipi kontrolöre ait kontrol işaretleri	43
Şekil 4.16	: Tasarlanan bulanık PID tipi kontrolöre ait sistem yanıtları	44
Şekil 4.17	: Tasarlanan bulanık PID tipi kontrolöre ait kontrol işaretleri	44
Şekil 4.18	: Tasarlanan bulanık PID tipi kontrolöre ait sistem yanıtları	45
Şekil 4.19	: Tasarlanan bulanık PID tipi kontrolöre ait kontrol işaretleri	45
Şekil 4.20	: Ölü zamansız ikinci dereceden doğrusal sisteme tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait sistem yanıtları	48
Şekil 4.21	: Ölü zamansız ikinci dereceden doğrusal sisteme tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait kontrol işaretleri	48
Şekil 4.22	: Eşik değerine sahip olmayan kontrolöre ait sistem yanıtları	49
Şekil 4.23	: Eşik değerine sahip olmayan kontrolöre ait kontrol işaretleri	49
Şekil 4.24	: Eşik değerine sahip olan kontrolöre ait sistem yanıtları	50
Şekil 4.25	: Eşik değerine sahip olan kontrolöre ait kontrol işaretleri	50
Şekil 4.26	: Dört parametrelili model ve gerçek sistem için tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait sistem yanıtları	51
Şekil 4.27	: Dört parametrelili model ve gerçek sistem için tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait kontrol işaretleri	51
Şekil 4.28	: Eşik değerine sahip olmayan kontrolöre ait sistem yanıtları	52
Şekil 4.29	: Eşik değerine sahip olmayan kontrolöre ait kontrol işaretleri	52
Şekil 4.30	: Eşik değerine sahip olan kontrolöre ait sistem yanıtları	53
Şekil 4.31	: Eşik değerine sahip olan kontrolöre ait kontrol işaretleri	53
Şekil 4.32	: Ölü zamansız birinci dereceden doğrusal sisteme tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait sistem yanıtları	54

Şekil 4.33	: Ölü zamansız birinci dereceden doğrusal sisteme tasarlanan bulanık PID tipi kontrolöre ait kontrol işaretleri	55
Şekil 4.34	: Katsayıların $a_1 = 2$, $a_2 = 8$, $b_1 = 0.5$, $b_2 = 4.5$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları	55
Şekil 4.35	: Katsayıların $a_1 = 2$, $a_2 = 8$, $b_1 = 0.5$, $b_2 = 4.5$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri	56
Şekil 4.36	: Katsayıların $a_1 = 2.3$, $a_2 = 7.5$, $b_1 = 0.3$, $b_2 = 5$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları	56
Şekil 4.37	: Katsayıların $a_1 = 2.3$, $a_2 = 7.5$, $b_1 = 0.3$, $b_2 = 5$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri	57
Şekil 4.38	: Katsayıların $a_1 = 0.25$, $a_2 = 0.4$, $b_1 = 0.05$, $b_2 = 0.8$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları	58
Şekil 4.39	: Katsayıların $a_1 = 0.25$, $a_2 = 0.4$, $b_1 = 0.05$, $b_2 = 0.8$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri	58
Şekil 4.40	: Katsayıların $a_1 = 0.28$, $a_2 = 0.45$, $b_1 = 0.14$ ve $b_2 = 2.2$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları	59
Şekil 4.41	: Katsayıların $a_1 = 0.28$, $a_2 = 0.45$, $b_1 = 0.14$ ve $b_2 = 2.2$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri	59
Şekil 4.42	: Katsayıların $K_f = 1.2$ ve $K_{fd} = 1$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları	60
Şekil 4.43	: Katsayıların $K_f = 1.2$ ve $K_{fd} = 1$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri	61
Şekil 4.44	: Katsayıların $K_f = 22$ ve $K_{fd} = 0.21$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları	61
Şekil 4.45	: Katsayıların $K_f = 22$ ve $K_{fd} = 0.21$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri	62
Şekil 4.46	: Katsayıların $K_f = 6.45$ ve $K_{fd} = 1$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları	63
Şekil 4.47	: Katsayıların $K_f = 6.45$ ve $K_{fd} = 1$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri	63
Şekil 4.48	: Katsayıların $K_f = 7.8$ ve $K_{fd} = 0.25$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları	64
Şekil 4.49	: Katsayıların $K_f = 7.8$ ve $K_{fd} = 0.25$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri	64
Şekil 4.50	: Ölü zamansız gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	65
Şekil 4.51	: Ölü zamansız gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	66

Şekil 4.52	: Ölü zamanlı gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	67
Şekil 4.53	: Ölü zamanlı gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	67
Şekil 4.54	: Ölü zamansız gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	68
Şekil 4.55	: Ölü zamansız gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	69
Şekil 4.56	: Ölü zamanlı gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	69
Şekil 4.57	: Ölü zamanlı gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	70
Şekil 4.58	: Ölü zamansız gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	71
Şekil 4.59	: Ölü zamansız gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	72
Şekil 4.60	: Ölü zamanlı gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	72
Şekil 4.61	: Ölü zamanlı gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları	73
Şekil 4.62	: Özayarlama yöntemlerinin karşılaştırılması	74

SEMBOL LİSTESİ

AIW0	: PLC analog girişı
AQW0	: PLC analog çıkışı
K	: Sistem kazancı
L	: Ölü zaman değeri
T_{1.2}	: Sistem zaman sabiti
u	: Kontrol işareti
e	: Hata işareti
ė	: Hatanın türevi
K_p	: PID' ye ait oran katsayısı
K_i	: PID' ye ait integral katsayısı
K_d	: PID' ye ait türev katsayısı
K_e	: Bulanık PID tipi kontrolör giriş ölçekleme faktörü
K_{de}	: Bulanık PID tipi kontrolör giriş ölçekleme faktörü
K_{ie}	: Bulanık PID tipi kontrolör giriş ölçekleme faktörü
β	: Bulanık PID tipi kontrolör çıkış ölçekleme faktörü
α	: Bulanık PID tipi kontrolör çıkış ölçekleme faktörü
δ	: Mutlak tepe değeri
r_v	: Bağlı hız değeri
de	: Hata değerindeki değişim
dde	: Hata değerindeki hızlanma
γ	: Bulanık parametre düzenleyicisi çıkış parametresi
K_f	: Bulanık parametre düzenleyicisi çıkış ölçekleme faktörü
K_{fd}	: Bulanık parametre düzenleyicisi çıkış ölçekleme faktörü
v	: Kontrol işaretindeki değişim
μ	: Üyelik fonksiyonu
U	: Bulanık PID tipi kontrolör çıkışı
λ	: Kural ateşleme şiddeti

BULANIK PID TİPİ KONTROL ALGORİTMALARI

ÖZET

Bulanık küme kuramı, ilk olarak 1965 yılında Zadeh (1965) tarafından ortaya konulduktan sonra ilk uygulamasını 1974 yılında kontrol alanında bulmuştur. Takip eden yıllarda bulanık kontrol birçok araştırmacı tarafından önemli bir çalışma konusu olarak ele alınmış ve son on yıl içerisinde bulanık kontrolörler endüstride geniş bir kullanım alanına sahip olmuştur. Bu kontrolörler arasında en yaygın olanı, yapıları itibariyle PID kontrolörleri ile denkliği ortaya konmuş olan bulanık PID tipi kontrolörlerdir. Literatürde özayarlamasız bulanık PID tipi kontrolörlerin yaygın olarak kullanılmasına karşın, son zamanlarda bu tip kontrolörlerin başarımını artırmak maksadıyla çeşitli uyarlama yöntemlerinin kullanıldığı özayarlamalı bulanık PID tipi kontrolör tasarımına önem verilmiştir.

Elde edilen kontrol algoritmalarının genellikle sadece bilgisayar ortamında gerçekleşmesinden dolayı, bu algoritmaların fiziksel sistemler üzerinde uygulanmaları neticesinde birtakım farklılıklar ortaya çıkabilmektedir. Bu tezde, PID ve bulanık PID tipi kontrol algoritmaları oluşabilecek bu farklılıkları, elde edilecek sistem yanıtlarını ve bulanık kontrolör tasarımında uygulanan özayarlama yöntemlerinin sonuçlarını karşılaştırmak amacıyla PCS 327 Proses Kontrol Simülatörü adı verilen deney seti üzerinde uygulanmış ve elde edilen sonuçlar bilgisayar ortamında simülasyon ile elde edilen sonuçlarla karşılaştırılmıştır. Yapılan deneyler neticesinde başarılı sonuçlara ulaşılmıştır.

Öncelikle, simülatör üzerinde ölü zamansız ve ölü zamanlı olmak üzere oluşturulan iki ayrı sistem basamak yanıtı modelleme yöntemlerinden üç ve dört parametreye sahip model yapıları ile modellenmiştir. Elde edilen modellerin gerçek sistemleri yeterli düzeyde temsil ettikleri kanısına varılmıştır. Sistemlere ait modellerin elde edilmesinin ardından kontrolör tasarımına geçilmiş, üç ve dört parametrelili modeller ile gerçek sisteme PID kontrolörü tasarlanmıştır. Daha sonra tasarlanan kontrolöre ait gerçek sistem yanıtları ile simülasyon yanıtları karşılaştırılmıştır. Kontrolör tasarımında ikinci aşamada ise bulanık kontrolör yapıları ele alınmış, yapıları itibariyle bir PID kontrolörüne benzeyen, bulanık PID tipi kontrolörler hakkında teorik bilgiler verilmiştir. Burada bulanık PID tipi kontrolörler temel olarak

1. Ölçekleme faktörleri, güncel olmayan şekilde (off-line) tasarım başlangıcında ayarlanarak bu değerler için sistem yanıtının elde edilebildiği özayarlamasız bulanık PID tipi kontrolörler
2. Kontrolöre ait iki adet ölçekleme faktörünün değişik algoritmalar ile güncel (on-line) olarak ayarlanan ve özayarlama neticesinde kontrolöre ait sistem yanıtının elde edildiği özayarlamalı bulanık PID tipi kontrolörler

şeklinde sınıflandırılmıştır.

Daha sonra elde edilen modeller ile gerçek sisteme özayarlamasız bulanık PID tipi kontrolör tasarlanmış ve dört parametrelili modelin üç parametrelili modele nazaran

gerçek sistem yanıtına çok daha yakın bir yanıtı sahip olduğu gözlenmiştir. Bu sebepten ötürü, bir sonraki aşamada sadece dört parametrelili model ve gerçek sisteme, sistem başarımını artırmak maksadıyla aşım gözleyicisi, fonksiyon ayarlayıcısı ve bağıl hız gözleyicisi yoluyla iki adet ölçekleme faktörü uyarlanan özayarlamalı bulanık PID tipi kontrolör tasarlanmıştır ve her bir uyarlama yöntemine ilişkin elde edilen sistem yanıtları karşılaştırılmıştır. Özellikle aşım gözleyicisi yönteminde sistem yanıtında daha iyi bir sonucun elde edilebilmesi için eşik değeri kullanma düşüncesi önerilmiştir. Elde edilen sonuçlarda, ölçekleme faktörleri düzgün olarak ayarlanmayan bir bulanık kontrolör yanıtının uygulanan bu yöntemler neticesinde iyileştirilebildiği görülmüştür. Kontrolör tasarımında amaç minimum sayıda kurala sahip bir kural tabanı oluşturmak olduğu için toplam $2 \times 2 = 4$ adet kural yazılmıştır.

Çalışmada sisteme ait modellerin elde edilmesinde ve kontrol aşamalarında Siemens S-7 200 CPU 214 modelli PLC cihazı kullanılmıştır. Gerçek sistemle PLC arasındaki bağlantı EM-235 Analog giriş-çıkış ünitesi ile sağlanmıştır. Ayrıca simülasyon yanıtları Matlab 5.3 ve Simulink programları vasıtasıyla elde edilmiştir.



FUZZY PID TYPE CONTROL ALGORITHMS

SUMMARY

After proposed by Zadeh (1965) in 1965, the fuzzy set theory had got its first application at the control research in 1974. In the following years, the fuzzy control was accepted as an important study by many researchers and in the last decade fuzzy controllers had found a wide usage area in the industry. Since their structures are almost as the same as PID controllers structures, the fuzzy PID type controllers had became the most widely used controllers. Although many fuzzy PID type controllers which has no self tuning mechanism are used in literature, in the last years self-tuned fuzzy PID type controllers in which many tuning algorithms are used had given importance in order to improve the performance of non self-tuned fuzzy PID type controllers.

Since the control algorithms are materialized in computer environment generally, there could be some differences when these algorithms are applied to the physical systems. In this thesis, the PID and fuzzy PID type control algorithms are applied to the so called PCS 327 Process Control Simulator experimental set in order to compare probable differences, obtained system responses and the results of self tuning algorithms which were used in designing the self-tuned fuzzy PID type controllers and lastly the results got from the physical system had been compared to the simulation results which were got from computer. After the experiments successful results were acquired.

First of all, two different systems with a dead time and without a dead time that are composed on the simulator were modelled by the step response modeling methods including three and four parametered model structures. The obtained models for each system had been thought to be sufficient. After getting the models for each system, the next stage became designing the controllers and a PID controller had been designed for three parametered model, four parametered model and real system, then the real system results and simulation results had been compared. At the second stage of designing controllers, some theoretical information was given about the fuzzy PID type controllers which have similar structures as PID controllers. The fuzzy PID type controllers were basically classified in two categories as follows;

1. The fuzzy PID type controllers that has no tuning mechanism in which the scaling factors of the controller are tuned at the beginning stage of controller design in an off line manner and the response of the system can be got for these parameter values
2. The fuzzy PID type controllers that has a tuning mechanism in which the two scaling factors of the controller are tuned by different algorithms in an on line manner and the response of the system can be got for self tuning parameter values.

After this, a fuzzy PID type controller that has no tuning mechanism was designed for the models and the real system and it is observed that the four parametered model

has more similar system response to real system compared with the three parametered model response. For this reason, the fuzzy PID type controller which has a tuning mechanism was designed for only four parametered model using the peak observer, function tuner and relative rate observer methods to increase the performance of the system by tuning the two scaling factors of the controller and the system responses concerning to each tuning method were compared. Especially, for the peak observer method, idea of using a threshold value was proposed to get a better system response. After getting the systems responses for this controller, it was seen that the system responses can be improved after applying these methods to a fuzzy PID type controller whose parameters were tuned improperly. Since the goal was to have a rule base which had minimum number of rules, while designing the controllers only $2 \times 2 = 4$ rules were written.

In the study, S-7 200 CPU 214 PLC device was used at the stages of getting the models and controlling the systems. The interconnection between the real system and PLC was established by using the EM-235 Analogue input-output unit. Also the simulation results were got by using the Matlab 5.3 and Simulink programs.



1. GİRİŞ

Birçok araştırmanın ve yüksek sayıda değişik çözüm yolunun önerilmesine karşın, çoğu endüstriyel kontrol sistemi hala PID kontrolüne dayanmaktadır. PID kontrolörünün endüstride kullanım oranının yaklaşık olarak %90 ile %99 arasında olduğu bilinmektedir. Bu yüksek kullanım oranının bazı sebepleri şu şekilde ifade edilebilir:

- a) PID kontrolörleri dayanıklı ve tasarımları basittir.
- b) PID ile sistem yanıtına ait parametreler arasında belirgin bir ilişki mevcuttur. Bir PID kontrolörünün sadece üç adet parametreye sahip olmasından dolayı, operatörler bu parametrelerin etkileri ve sistem yanıtı açısından birbirlerine karşı olan üstünlükleri hakkında geniş bir bilgiye sahiptir.
- c) Son yıllarda PID kontrolörüne ait parametrelerin ayarlanmasına ilişkin birçok değişik yöntem önerilmiştir.

Ancak, PID kontrolörleri bütün kontrol problemlerine karşı genel bir çözüm sağlayamamaktadır. Zaman gecikmesine ve doğrusal olmayan özelliklere sahip, karmaşık, zamanla değişen sistemler karşılaşılabilecek bu tür problemlere örnek olarak verilebilir. Bir proses analitik modellerle ifade edilemeyecek kadar karmaşık bir yapıya sahip ise o zaman bu sistemin klasik yaklaşımlar ile etkili biçimde kontrol edilebilmesi imkansız hale gelmektedir.

Yapay zeka ve kontrol mühendisliği arasında sınır olan bulanık kontrol bir çözüm olarak düşünülebilir. Bulanık mantık ile kontrol, uzman kişi bilgisini ve kontrol yöntemlerini sayısal algoritmalara başarılı şekilde aktarabilmektedir.

Zadeh, 1965 yılında Zadeh (1965) ile bulanık küme kuramını önerdikten sonra 1968' de bulanık algoritmalar konusu üzerinde çalışmış, 1970' de ise Bellman ile beraber bulanık kontrol kavramını açıklayan bir makale yayımlamıştır (Bellman ve Zadeh, 1970). Bu makalede, bulanık kontrolde insan bilgisinin "EĞER - O HALDE" kurallarıyla ifade edilmesi için dilsel değişkenler kavramı ortaya konulmuştur. Son 30 yıl içerisinde, bulanık kontrolör uygulamaları çeşitli endüstriyel kontrol

uygulamalarını ve önemli araştırma çalışmalarını kapsayacak şekilde genişlemiştir. 1974 yılında Mamdani, Zadeh (1973) ile önerilen çıkarım mekanizmasını dinamik bir sistemi kontrol etmekte kullanmıştır (Mamdani, 1974). Bir yıl sonra Mamdani ve Assilian ilk bulanık mantık kontrolörünü (FLC) geliştirmişler ve bu kontrolörü bir buhar makinasını kontrol etmek amacıyla kullanmışlardır (Mamdani ve Assilian, 1975). Ancak bu algoritma büyük ölçüde uzman operatörün bilgi ve tecrübesine dayanmaktadır. Mac Vicar-Whelan bu olumsuzlukları ortadan kaldırabilmek amacıyla birtakım kurallar önermiştir ve tasarlanan kontrolörler PD ve PI kontrolörlere benzer yapıda elde edilmiştir (Mac Vicar-Whelan, 1976).

Bulanık kontrolörlerde yaygın olarak Mamdani (1974) ile önerilen min-max çıkarım mekanizması ve Mizumoto (1990) ile önerilen çarpım-toplam çıkarım mekanizması kullanılmaktadır. Durulaştırmanın zaman alıcı bir işlem olması nedeniyle, özellikle Takagi ve Sugeno (1985)' de önerilen bulanık kontrol kurallarının sonuç kısmında bulanık kümeler yerine gerçel sayılar kullanılmasıyla elde edilen kural yapısı ile çarpım-toplam çıkarım mekanizmasının beraber olarak kullanılması, oldukça iyi bir başarımla ve kolay bir algoritma üretmektedir.

Mizumoto ve Qiao (1996)' da, bu tip çıkarım mekanizmasına sahip bir bulanık kontrolörü PID'ye ilişkilendiren, PID kontrolörün yapısına benzer bir kontrolör yapısı önerilmiştir. Elde edilen bu kontrolöre özayarlamasız bulanık PID tipi kontrolör adı verilmektedir. Burada kontrol işareti bilgi tabanı ve çıkarım mekanizmasından üretilmekte ve kontrolöre ait ölçekleme faktörleri tasarım başlangıcında ayarlanarak kontrolöre ait sistem yanıtı elde edilmektedir.

Bulanık kontrolör tasarımı üç önemli safhadan meydana gelmektedir. Bunlar; bilgi tabanı tasarımı, ayarlama parametreleri ve üyelik fonksiyonlarıdır. Tasarlanan kontrolörün istenilen başarımla sağlayabilmesi için bu üç safhanın ayarlanması gerekmektedir. Ancak sadece ölçekleme faktörlerinin ayarlanması ile gerçek zaman kontrolünün gereksinimleri sağlanabilmektedir (Woo ve diğ., 1998). Sadece üyelik fonksiyonlarının ayarlanması ile ancak geçici hal yanıtında iyileştirme yapılabilir, sürekli hal yanıtında bir değişiklik sağlanamaz. Dolayısıyla, özayarlamalı ölçekleme faktörlerine sahip bulanık kontrolör tasarımı, istenilen başarımla elde etmek için önemli bir ihtiyaçtır. Bu sebepten ötürü, özayarlamasız bulanık PID tipi kontrolörlerin ölçekleme faktörleri çeşitli algoritmaların kullanımı ile uyarlanmış ve

sistem başarımı önemli ölçüde artırılmıştır. Elde edilen bu tip kontrolörlere özayarlamalı bulanık PID tipi kontrolörler adı verilmektedir.

Bu tez kapsamında modelleme, PID kontrolörü ve bulanık PID tipi kontrolörü tasarımı ele alınmıştır. 2. Bölüm’ de deney setinde oluşturulan iki ayrı sistem basamak yanıtı yöntemleriyle üç ve dört parametrelili olarak modellenmiş, 3. Bölüm’ de elde edilen modeller ile gerçek sisteme PID kontrolörü tasarlanmış ve kontrolöre ait sistem yanıtları irdelenmiştir. 4. Bölüm’ de sırasıyla özayarlamasız ve özayarlamalı bulanık PID tipi kontrolörler hakkında detaylı olarak teorik bilgiler verilmiş ve ilk olarak özayarlamasız bulanık PID tipi kontrolör tasarlanmış, daha sonra ölçekleme faktörleri düzgün olarak ayarlanamayan bulanık kontrolöre ait başarımı artırmak maksadıyla, bu kontrolöre sırasıyla aşım gözleyicisi, fonksiyon ayarlayıcısı ve bağıl hız gözleyicisi ile parametre uyarlama yöntemleri uygulanarak özayarlamalı bulanık PID tipi kontrolör tasarlanmış ve elde edilen sonuçlar irdelenmiştir. Ayrıca aşım gözleyicisi yönteminde eşik değeri kullanma düşüncesi önerilmiştir. Yapılan bu çalışmanın, bu konularda çalışacak diğer kontrol mühendislerine rehberlik edeceği düşünülmektedir.

2. SİSTEMLER GENEL BAKIŞ VE SİSTEM MODELLEME

2.1 Sistem Genel Bakış

Tez çalışmasının yapıldığı deney seti genel olarak Şekil 2.1’ de görülmektedir. Deney seti temel olarak PCS 327 Proses Kontrol Simülatörü, kontrolör amaçlı kullanılan PLC, osiloskop ve bilgisayardan meydana gelmektedir.

Proses Kontrol Simülatörü PCS 327, tümleşik devre elemanlarından oluşturulmuş özel amaçlı analog bir simülatördür. Doğrusal olmayan fonksiyonlar ile beraber birçok değişken bağlantısı simülatörü, doğrusal ve doğrusal olmayan proseslerin dinamik yanıtları ile oransal, integral, türev ve sistem başarımını artıran diğer birçok kontrol özellikleri üzerinde detaylı olarak çalışma yapılabilmesi açısından çok yönlü kılmalıdır.



Şekil 2.1 Deney setinin genel görünümü

Proses temel olarak transfer fonksiyonu $\frac{1}{s+1}$ olan bir sistemden, zaman sabitleri bir düğme vasıtasıyla 1 sn yada 10 ms konumuna getirilebilen ve transfer fonksiyonları sırasıyla $\frac{1}{s+1}$ ve $\frac{1}{s}$ olan iki adet sistemden, yine gecikme zamanı değeri düğme vasıtasıyla 1 sn yada 10 ms konumuna getirilebilen bir adet ölü zaman elemanından ve bir adet dönüştürücü elemanından oluşmaktadır. İki adet sistem elemanı bir anahtar yardımıyla istenen transfer fonksiyonu elde edilmek amacıyla sistem yada integratör konumuna getirilebilmektedir. Dönüştürücü, girişini "-1" değeri ile çarparak işaret değişimini sağlar. Bu elemanlar kullanılarak sistem tipi 0, 1, 2 olmak üzere 12 değişik doğrusal sistem transfer fonksiyonu elde edilebilir.

Bu elemanlara ilave olarak proses çıkışını algılayan ve ölçen elemanlar ile sistem çıkışını ve referans değerini sürekli olarak takip eden bir gösterge, PID kontrolörüne ait oran, integral ve türev bileşenlerinin değerlerinin ayarlandığı düğmeler mevcuttur. Ancak bu çalışmada kontrolör olarak PLC kullanıldığı için PID kontrolörüne ait olan bu elemanlar kullanılmamıştır.

Programlanabilir lojik kontrolör (Programmable Logic Controller, PLC) endüstriyel otomasyon sistemlerinin kumanda ve kontrol devrelerini gerçeklemeye uygun yapıda giriş-çıkış birimleri ve iletişim arabirimleri ile donatılmış, kontrol yapısına uygun bir sistem programı altında çalışan bir endüstriyel bilgisayardır (Kurtulan, 2001).

Günümüzde üretilen PLC'ler, lojik temelli işlemlere ek olarak aritmetik ve özel matematiksel işlemlerin yapılmasını sağlayan komutları da içerirler. Komut kümesinin genişlemesi sonucu daha karmaşık kumanda ve kontrol işlevleri gerçekleştirilmektedir. PLC'lerin en yaygın kullanıldığı alanlar, endüstriyel otomasyon sistemlerinin kumanda devreleridir. Bilindiği gibi kumanda devreleri, yardımcı röle yada kontaktör, zaman rölesi ve sayıcı gibi elemanlarla gerçekleştirilen devrelerdir. Günümüzde bu tür devrelerin yerini aynı işlevi sağlayan PLC'li kumanda sistemleri almıştır. Küçük boyutlu birkaç PLC modeli dışında, yeni üretilmekte olan bütün PLC modellerinde, bir kontrol algoritması yazmak için gerekli aritmetik işlemlerin yapılmasını sağlayan komut desteği bulunur. Bir PLC'nin geribeslemeli kontrol sistemlerinde sayısal kontrolör olarak kullanılması, analog işaretleri sayısal işaretlere (ADC) ve sayısal işaretleri analog işaretlere dönüştüren (DAC) giriş-çıkış birimleri ile sağlanır.

Tez çalışmasında sistemin kontrolü Siemens S7-200 CPU 214 model bir PLC cihazı üzerinden yapılmaktadır. Simülatörde, gerilim cinsinden olan giriş-çıkış değerleri PLC'nin EM 235 analog giriş-çıkış ünitesinden sağlanmaktadır. Sisteme giriş işareti AQW0 analog çıkışından 0-10V değer aralığında gönderilmektedir. Sistemden elde edilen çıkış değeri 0-10V aralığında olup, AIW0 analog girişinden okunmaktadır. AIW0 ve AQW0 için değer 0V olduğu anda PLC'de bunun 16 bit karşılığı 0 kelime değeri, değer 10V olduğu anda ise 32000 kelime değeri elde edilmektedir. Dolayısıyla kelime değeri olarak sistem giriş ve çıkış ölçeği [0-32000] aralığında olmaktadır.

Giriş ve çıkış değerlerinin PLC programında gerilim olarak değil de kelime değeri şeklinde görülmesi nedeniyle tez çalışması boyunca verilen bütün grafikler için bu değerler 3200 kelime değerine bölünerek [0-10V] aralığına ölçeklendirilmiştir.

2.2 Sistem Modellerinin Basamak Yanıtı Yöntemi İle Elde Edilmesi

Sistem dinamiğinin modellenmesinde, en basit model olarak iki parametrelili model yapısı kullanılır (Aström ve Höggland, 1995). Birinci mertebeden ölü zamanı bulunmayan sistemlerin iki parametre ile modellenmesine ilişkin genel yapı

$$G_{2a}(s) = \frac{K}{1 + sT_{ar}} \quad (2.1)$$

şeklindedir. Bu yaklaşımda "K" sistem kazancı, "T_{ar}" ise sistem zaman sabitidir. T_{ar} zamanı kabaca basamak cevabının yerleşmesi için geçen süre olarak alınır.

Kendi yapılarında integral etkisi olan sistemlerin iki parametre ile modellenmesine ilişkin genel yapı ise

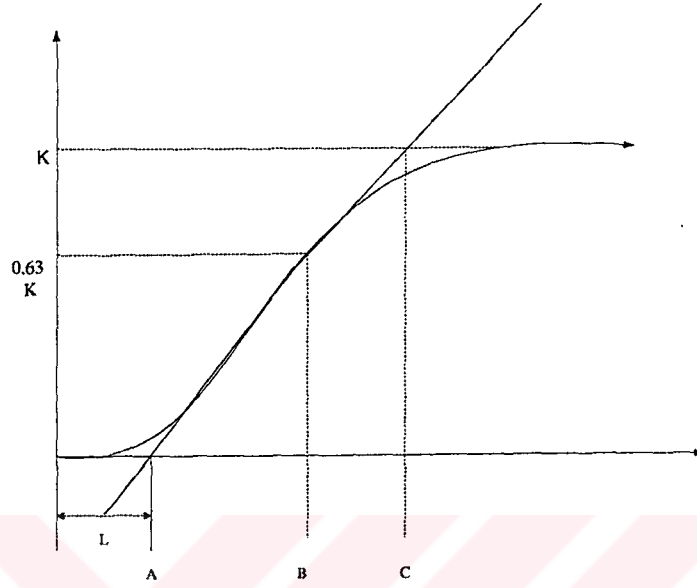
$$G_{2b}(s) = \frac{a}{sL} e^{-sL} \quad (2.2)$$

şeklinde verilir.

Daha iyi bir yaklaşım ise sistemin üç parametre ile modellenmesi ile elde edilir. Bu tür modelin genel yapısı

$$G(s) = \frac{K}{1 + sT} e^{-sL} \quad (2.3)$$

şeklindedir. "K" parametresi sistemin kazancını, "T" parametresi sistemin zaman sabitini ve "L" parametresi sistemin ölü zamanını ifade eder. Modele ait parametrelerin değerleri grafiksel olarak belirlenir. Bu yapı Şekil 2.2' de verilmiştir.



Şekil 2.2 Basamak cevaplı sistemlerin üç parametrelili modellerinin grafiksel olarak belirlenmesi

Sistem kazancı sistem çıkışının sürekli hal değerinden belirlenir. Basamak cevabına çizilen en büyük eğimli teğetin yatay eksenini kestiği nokta ölü zaman değerini verir. Zaman sabitini belirlemenin iki değişik yolu vardır. Bunlardan ilki AB mesafesini hesaplamak, diğeri ise AC mesafesini hesaplamaktır. C noktası teğetin $s(t) = K$ eğrisi ile kesiştiği noktadır. B noktası ise basamak cevabının $0.63K$ değerine ulaştığı zaman değeridir. AB değerine bağlı olarak hesaplanan zaman sabitine ilişkin bir model yapısı genellikle daha iyi bir yaklaşım vermektedir.

dört parametrelili model yapısının transfer fonksiyonu

$$G(s) = \frac{K}{(1 + sT_1)(1 + sT_2)} e^{-sL} \quad (2.4)$$

şeklindedir. Bu modelde K sistem kazancı, T_1 ve T_2 zaman sabitleri, L ölü zamandır. K ve L değerleri üç parametrelili model yapısında belirlendiği şekilde belirlenir. Eşitlik 2.4 ile verilen transfer fonksiyonunun basamak cevabı

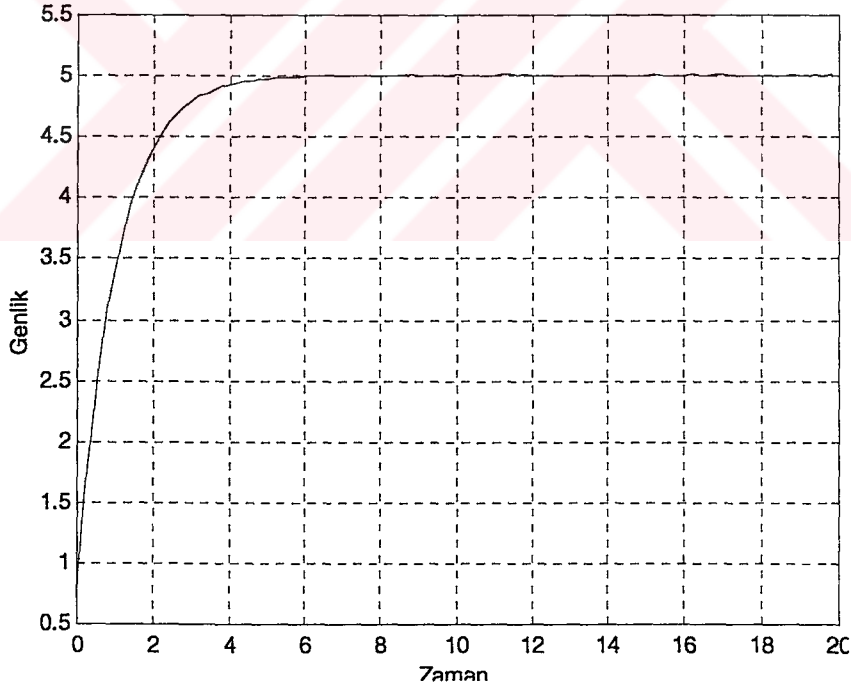
$$s(t) = K \left(1 + \frac{T_2 e^{-(t-L)/T_2} - T_1 e^{-(t-L)/T_1}}{T_1 - T_2} \right), T_1 \neq T_2 \quad (2.5)$$

şeklindedir. Elde edilen sistem basamak cevabında iki zaman değeri ve bu zaman değerlerine karşılık düşen genlik değerleri belirlenir. Bu zaman ve genlik değerleri için Eşitlik 2.5 ifadesi nümerik olarak çözülerek T_1 , T_2 zaman sabiti değerleri bulunur.

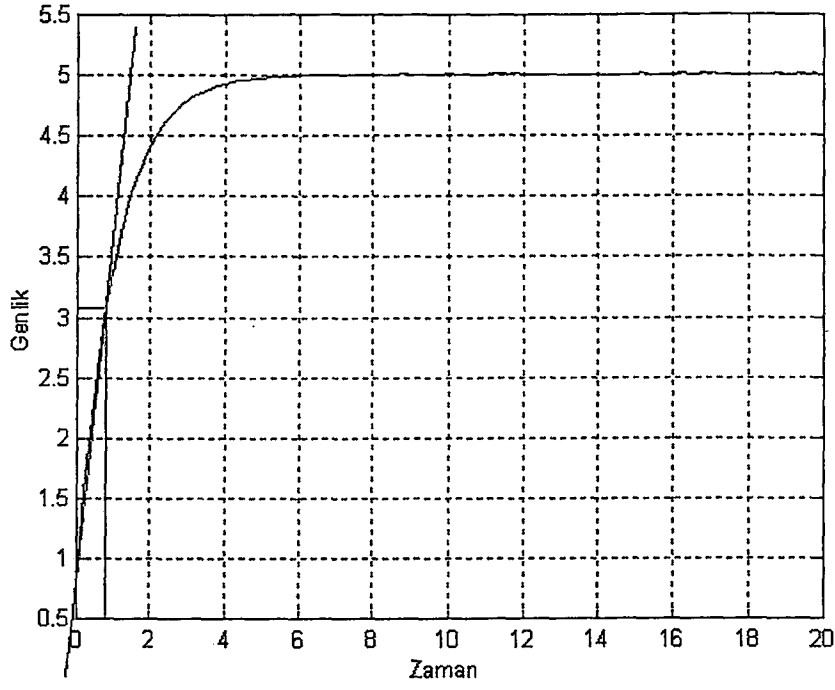
2.3 PCS 327 Proses Kontrol Simülatörünün Modellenmesi

2.3.1 PCS 327 Proses Simülatörünün Üç Parametre İle Modellenmesi

Önce, PCS 327 proses simülatörünün ölü zaman elemanı devre dışı bırakılarak, sistem ölü zamansız olarak modellenmiştir. Burada sistemin matematiksel modelini elde edebilmek için sisteme "5" (PLC programında kelime değeri karşılığı: 16000) genlikli birim basamak giriş işareti uygulanmıştır. Sistemin çıkışı Şekil 2.3' deki gibi gözlenmiştir.



Şekil 2.3 Ölü zamansız sistem için basamak işareti yanıtı



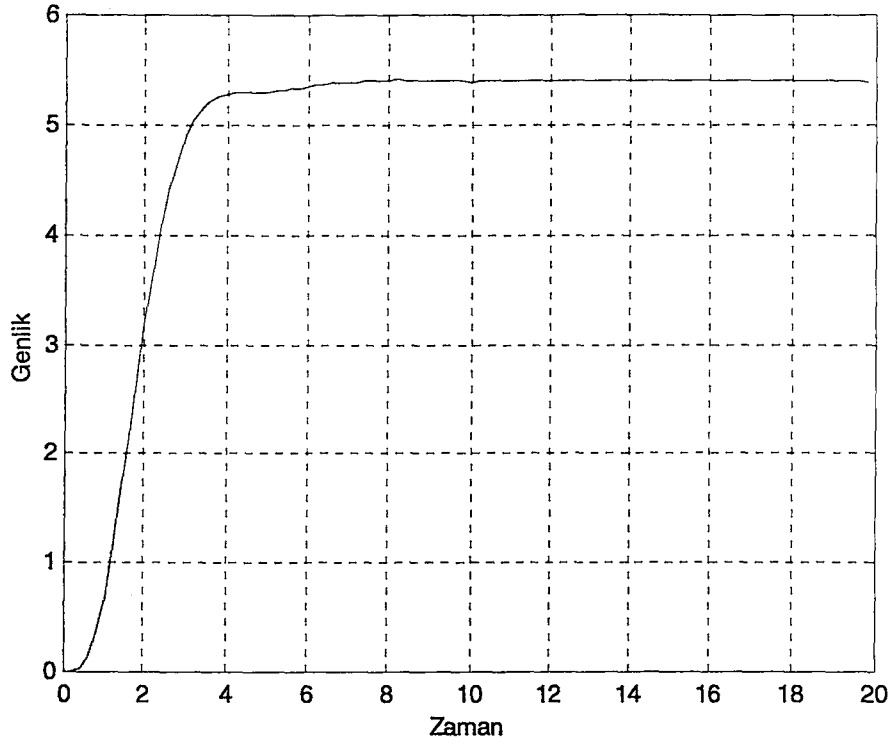
Şekil 2.4 Ölü zamansız sistem için üç parametrelili modele ait parametrelerin tespiti

Şekil 2.4’ de simülasyonda kullanılacak sistem modeline ilişkin parametrelerin tespiti görülmektedir. Buradan ölü zamansız sisteme ait üç parametrelili model için kazanç 1, ölü zaman 0 ve zaman sabiti 1 olarak hesaplanmıştır. O halde ölü zamansız sistemin üç parametrelili modelinin transfer fonksiyonu

$$G(s) = \frac{1}{s+1} \quad (2.6)$$

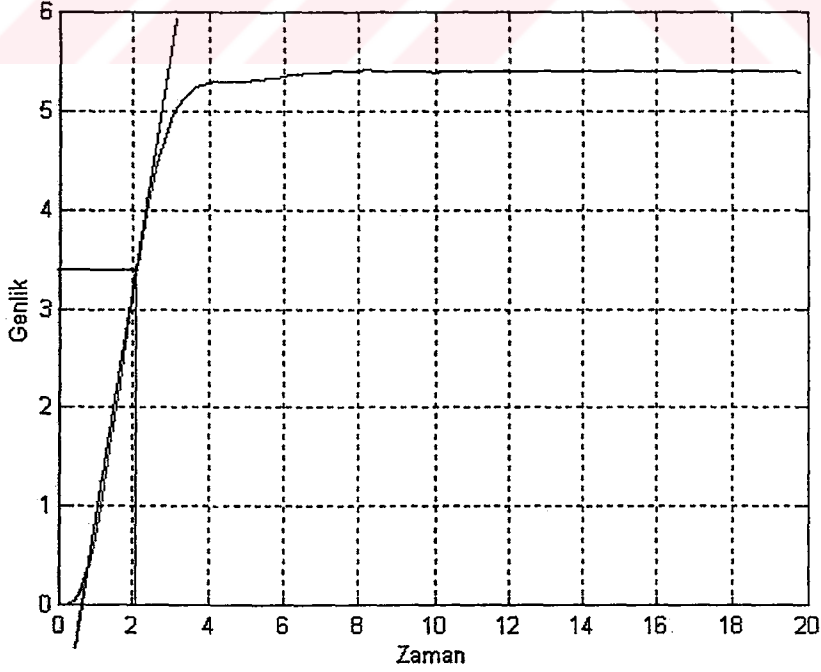
olarak hesaplanır.

Daha sonra simülatörün ölü zaman elemanı devreye alınarak sistem ölü zamanlı olarak modellenmiştir. Burada sistemin matematiksel modelini elde edebilmek için sisteme "5" (PLC programında kelime değeri karşılığı: 16000) genlikli birim basamak giriş işareti uygulanmıştır ve gerek Matlab simülasyon gerekse PLC programı 20 saniye süre ile çalıştırılmıştır. Bu koşullar altında sistemin çıkışı Şekil 2.5’ deki gibi gözlenmiştir.



Şekil 2.5 Ölü zamanlı sistem için basamak işareti yanıtı

Şekil 2.5' den de görüldüğü gibi, sistem giriş işaretine belirli bir gecikme sonrası cevap vermektedir, yani sistemde bir ölü zaman vardır. Sistemin basamak yanıtına bakıldığında yanıtın birinci mertebeden ölü zamanlı sistem yanıtı olduğu ve sistemin birinci mertebeden ölü zamanlı sistem şeklinde modellenebileceği düşünülmüştür.



Şekil 2.6 Ölü zamanlı sistem için üç parametrelili modele ait parametrelerin tespiti

Şekil 2.6' dan üç parametrelili model için sistem kazancı 1.08, zaman sabiti 1.2 ve ölü zaman 1 saniye olarak belirlenir. Hesaplanan bu parametre değerlerine göre sistemin transfer fonksiyonu

$$G(s) = \frac{1.08}{1.2s + 1} e^{-s} \quad (2.7)$$

olarak bulunur.

2.3.2 PCS 327 Proses Simülatörünün Dört Parametre İle Modellenmesi

Simülatörün ölü zamanlı modellenmesinde gerçek sisteme daha yakın bir modelin elde edilebilmesi maksadıyla dört parametrelili model yapısı uygun görülmüştür. Bu model yapısının transfer fonksiyonu

$$G(s) = \frac{K}{(1 + sT_1)(1 + sT_2)} e^{-sL} \quad (2.8)$$

şeklindedir. Bu transfer fonksiyonundaki dört adet parametreden kazanç ve ölü zaman değerleri üç parametrelili modelin elde edilmesindeki gibi bulunduğundan, dört parametrelili modelde de üç parametrelili modeldeki değerlerini koruyacaktır ve kazanç değeri 1.08, ölü zaman değeri ise 1 olacaktır. Diğer iki parametre olan T_1 ve T_2 değerlerinin bulunması için Eşitlik 2.8' in basamak cevabından faydalanılacaktır. Eşitlik 2.8'in basamak cevabı ise;

$$s(t) = K \left(1 + \frac{T_2 e^{-(t-L)/T_2} - T_1 e^{-(t-L)/T_1}}{T_1 - T_2} \right), \quad T_1 \neq T_2 \quad (2.9)$$

şeklindedir. O halde sistemin basamak cevabına ait iki adet zaman değeri ve bunlara karşılık gelen genlik değerlerinden faydalanılarak, Eşitlik 2.9 denklemi nümerik olarak çözülerek, T_1 ve T_2 değerleri bulunur.

Buna göre Şekil 2.5' de verilen basamak cevabında $t_1 = 1.52$, $s(t_1) = 2.0625$ ve $t_2 = 2.16$, $s(t_2) = 4.1875$ değerleri alınarak yapılan nümerik çözümde $T_1 = 0.2897$, $T_2 = 0.7$ olarak bulunmuştur. Bu değerler için elde edilen dört parametrelili model yapısı

$$G(s) = \frac{1.08}{(1 + 0.2897s)(1 + 0.7s)} e^{-s} \quad (2.10)$$

biçiminde olacaktır.

Gerek üç gerekse dört parametrelili model yapılarının elde edilmesinde kullanılan PLC programı Ek A' da verilmiştir.



3. PID KONTROLÖRÜ TASARIMI

3.1 PID Kontrol Algoritması

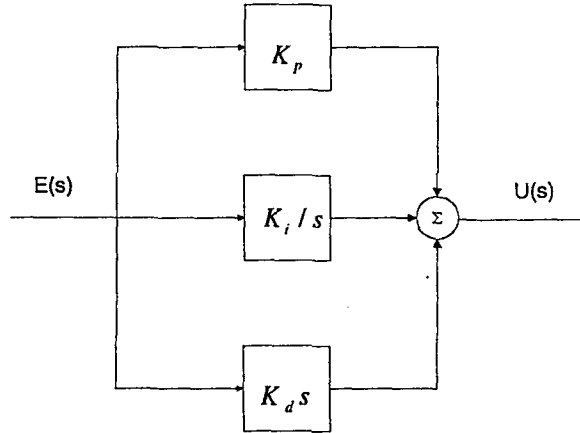
Endüstride en çok kullanılan kontrol algoritması PID kontrolörüdür. Çoğu geribesleme çevrimi bu algoritma veya bunun küçük değişikliklerle oluşturulmuş türevleriyle kontrol edilir. Bu bölümde daha önceki bölümde elde edilen üç ve dört parametrelili sistem modelleri ile gerçek sistem için PID kontrolörü tasarlamak maksadıyla, bu tip kontrolörün genel yapısı ile ilgili temel bilgiler verilmiştir. Genel olarak PID kontrolörüne ait kontrol işareti şu şekilde ifade edilebilir:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (3.1)$$

Bu standart PID gösteriminde $u(t)$ kontrol işareti, $e(t)$ hata işareti, K_p oran katsayısı, K_i integral katsayısı, K_d türev katsayısıdır. Eşitlik 3.1 ifadesinin Laplace Dönüşümü alınırsa

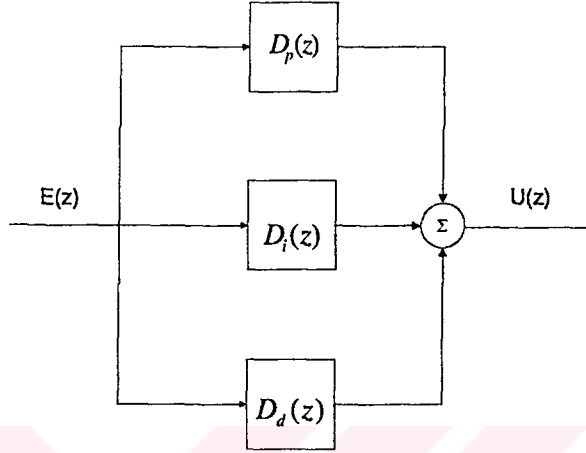
$$\frac{U(s)}{E(s)} = G(s) = K_p + \frac{K_i}{s} + K_d s \quad (3.2)$$

kontrolör transfer fonksiyonu elde edilir. Standart bir PID kontrolörünün blok gösterimi Şekil 3.1' de verilmiştir.



Şekil 3.1 Standart bir PID kontrolörünün blok gösterimi

Tüm kontrol algoritmaları sayısal olarak bilgisayar üzerinde gerçekleştirileceğinden, PID kontrolörünü de gerçekleyebilmek için kontrol algoritmasının ayrık fark denklemleri haline dönüştürülmesi gerekir. Böylece ayrık fark denklemleri ile ayrıklaştırılan PID kontrol algoritmasının integral ve türev kısımları sayısal olarak kolayca programlanabilir. Ayrık türev ve integral terimleri ile oran terimi birleştirilirse Şekil 3.2’ de gösterildiği gibi ayrık PID kontrolörü elde edilmiş olur.



Şekil 3.2 Ayrık PID kontrolörü blok gösterimi

Ayrık PID tasarımında poligon yaklaşımı ile ayrık integrasyon yöntemi seçilmesi durumunda Şekil 3.2’ de belirtilen ayrık PID kontrolörünün blok içerikleri

$$D_p(z) = K_p \quad (3.3a)$$

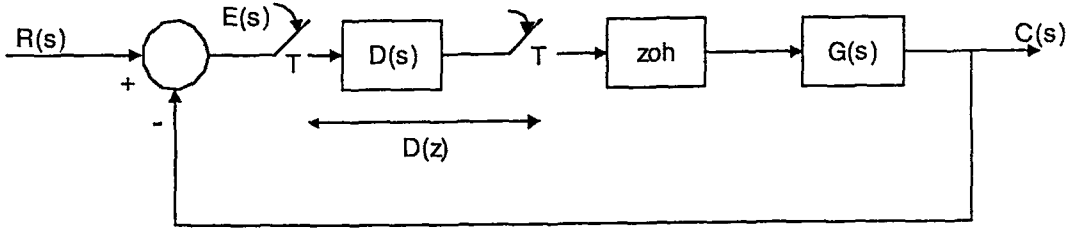
$$D_i(z) = \frac{K_i T(z+1)}{2(z-1)} \quad (3.3b)$$

$$D_d(z) = \frac{K_d(z-1)}{Tz} \quad (3.3c)$$

şeklinde elde edilir. Bu durumda ayrık PID kontrolörü transfer fonksiyonu

$$D(z) = K_p + \frac{K_i T(z+1)}{2(z-1)} + \frac{K_d(z-1)}{Tz} \quad (3.4)$$

biçiminde olur (Kuo, 1992). Bundan sonraki bölümde üç ve dört parametrelili sistem modelleri için tasarlanacak PID kontrolörü için Eşitlik 3.4’ de verilen ayrık PID transfer fonksiyonu kullanılacaktır.



Şekil 3.3 Sayısal kontrolöre sahip bir kontrol sisteminin blok gösterimi

Genel olarak sayısal bir kontrolöre sahip kontrol sisteminin blok gösterimi Şekil 3.3’de verilmiştir.

3.2 Üç Parametrelili Model İçin PID Kontrolörü Tasarımı

3.2.1 Ölü Zamansız Model İçin PID Kontrolörü Tasarımı

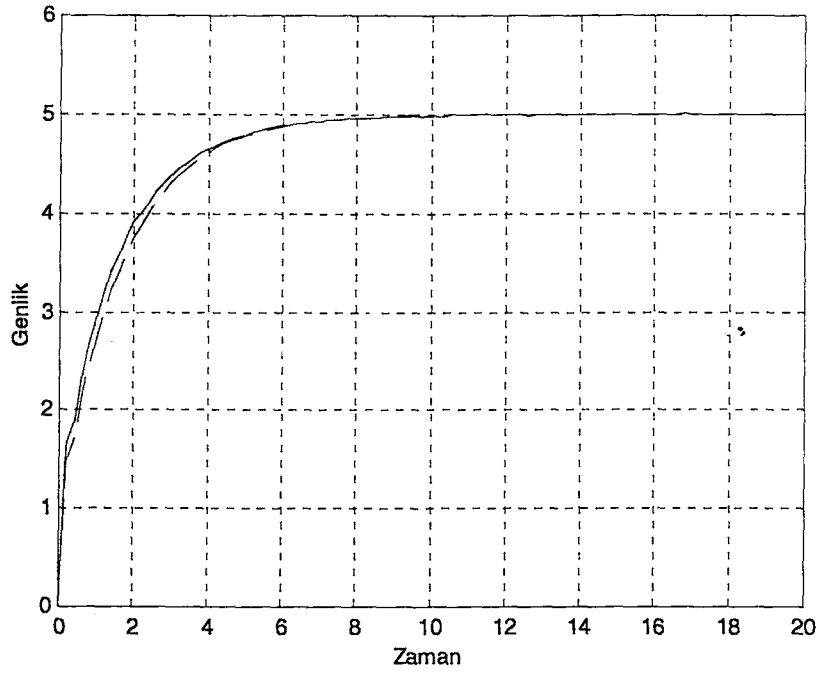
Önceki bölümde ölü zamansız sistemin üç parametrelili modeli

$$G(s) = \frac{1}{s+1} \quad (3.5)$$

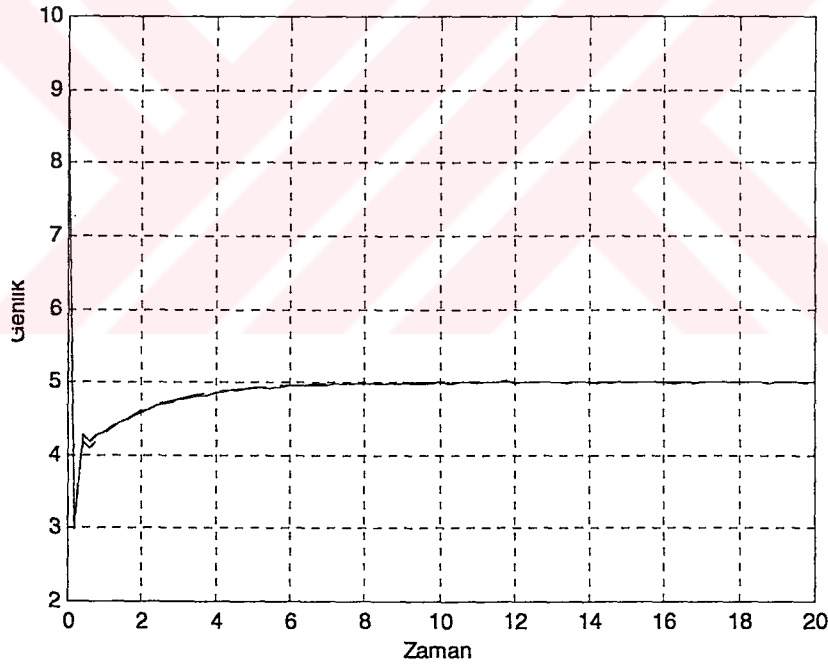
olarak elde edilmiştir. Örnekleme zamanı 0.2 saniye olmak üzere sistem modelinin z tanım bölgesindeki sıfırıncı dereceden tutuculu transfer fonksiyonu

$$G(z) = \frac{0.1813}{z-0.8187} \quad (3.6)$$

olarak bulunur. Ayrık PID kontrolörü sisteme iki adet sıfır ve bir kutup eklemektedir. İki adet sıfırdan birincisi sistem derecesini düşürmek ve sistemin $z = 0.8187$ ’ de bulunan kutbunu silmek amacıyla $z = 0.8187$ ’ de, ikinci sıfır ise kapalı çevrim sistem kutuplarını $z = 0.9175$ ve $z = -0.0988$ noktalarına yerleştirerek sistemin küçük aşım ve yerleşme zamanına sahip olması maksadıyla orijin ve sistem kutbu arasında, $z = 0.5$ ’de seçilmiştir. Seçilen bu sıfır değerleri için yapılan hesaplamalar sonucunda ayrık PID parametrelerinin değerleri $K_p = 1$, $K_i = 0.7124$, $K_d = 0.1572$ olarak bulunur. Hesaplanan parametreler için tasarlanan PID kontrolörüne ilişkin gerçek sistem ve Matlab simülasyon cevapları Şekil 3.4’ de, bu kontrolöre ilişkin kontrol işaretleri ise Şekil 3.5’ de görülmektedir.



Şekil 3.4 Ölü zamansız model ve gerçek sistem için PID kontrolörüne ilişkin sistem ve simülasyon cevapları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



Şekil 3.5 Ölü zamansız model ve gerçek sistem için tasarlanan PID kontrolörüne ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

PLC ve Matlab simülasyon programları 0.2 saniye örnekleme zamanı ile 20 saniye boyunca çalıştırılmış, referans değeri olarak simülasyonda 5 genlikli işaret, PLC programında 5 genlikli işarete karşılık düşen 16000 kelime değeri uygulanmıştır. Şekil 3.4' den görüleceği üzere, ölü zamansız sistem için tasarlanan PID

kontrolörüne ilişkin gerek gerçek sistem ve elde edilen üç parametrelili modele ait sistem cevapları birbirine benzerdir.

3.2.2 Ölü Zamanlı Model İçin PID Kontrolörü Tasarımı

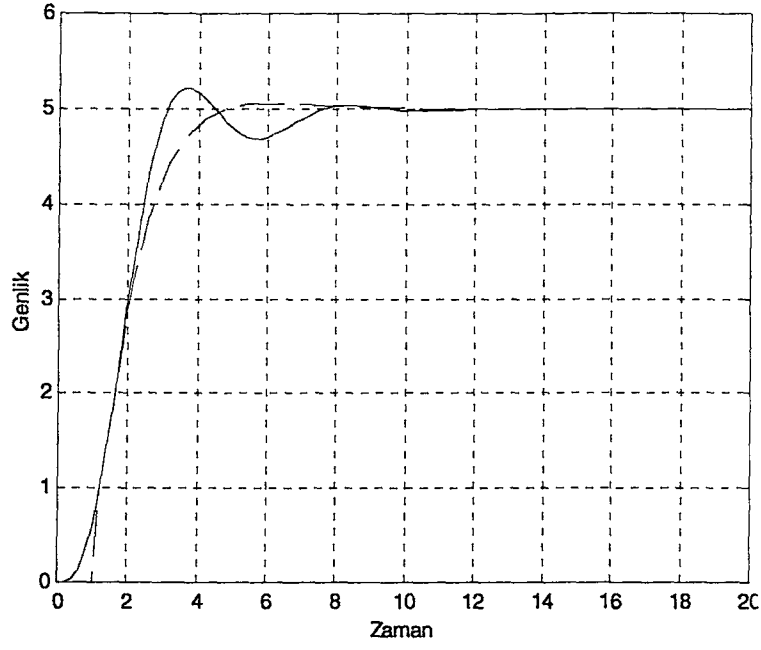
Önceki bölümde ölü zamanlı sisteme ilişkin üç parametrelili sistem modeli

$$G(s) = \frac{1.08}{1.2s + 1} e^{-s} \quad (3.7)$$

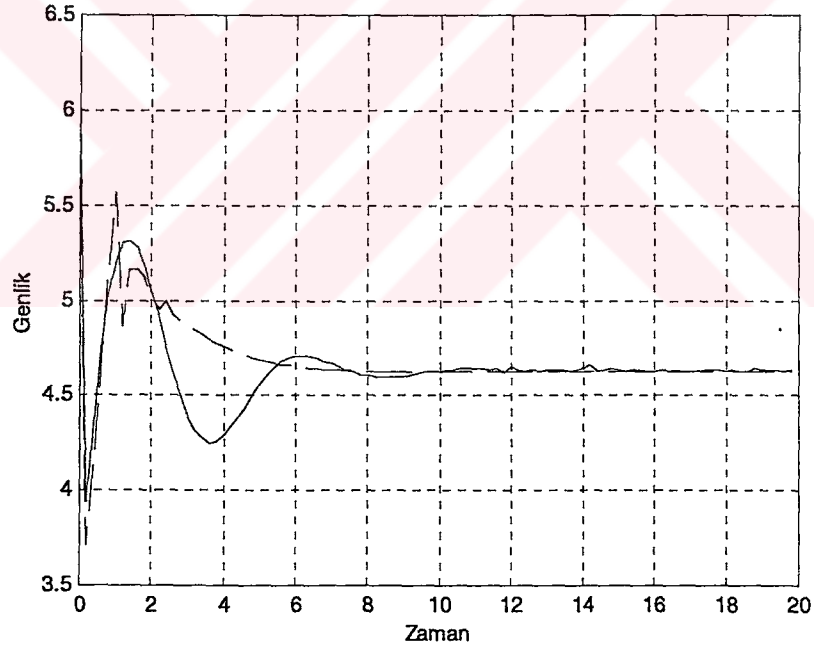
şeklinde elde edilmiştir. Örnekleme zamanı, $T = 0.2$ saniye için, sistemin z tanım bölgesindeki sıfırıncı dereceden tutuculu transfer fonksiyonu

$$G(z) = \frac{0.1658}{z^5(z - 0.8465)} \quad (3.8)$$

olarak hesaplanır. Kontrolör tasarımında PID kontrolörünün sıfırlarından birincisi sistem mertebesini bir derece düşürmek ve sistemin $z = 0.8465$ ' deki kutbunu silmek amacıyla $z = 0.8465$ 'de diğeri ise kapalı çevrim sistem kutuplarını $z_{1,2} = -0.6 \pm 0.37j$, $z_{3,4} = 0.04 \pm 0.72j$, $z_5 = 0.58$ ve $z_{6,7} = 0.78 \pm 0.1j$ noktalarına yerleştirerek sistemin küçük aşım ve yerleşme zamanına sahip olması amacıyla $z = 0.5$ 'de seçilmiştir. Bu durumda yapılan hesaplamalar neticesinde PID parametrelerinin değerleri $K_p = 0.65$, $K_i = 0.4629$, $K_d = 0.1022$ olarak bulunmuştur. Bu parametre değerleri ile tasarlanan kontrolöre ilişkin gerçek sistem ve Matlab simülasyon yanıtları Şekil 3.6' da, bu kontrolöre ait kontrol işaretleri Şekil 3.7' de görülmektedir. Simülasyonda referans olarak 5 genlikli işaret uygulanmış ve PLC programında çıkışın gerilim cinsinden olması nedeniyle programda 5 değerine karşılık düşen 16000 kelime değeri referans değeri olarak uygulanmıştır. Matlab simülasyonu ve PLC programı 0.2 saniye örnekleme zamanı ile 20 saniye müddetle çalıştırılmıştır.



Şekil 3.6 Ölü zamanlı model ve gerçek sistem için tasarlanan PID kontrolörüne ilişkin gerçek sistem ve simülasyon cevapları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



Şekil 3.7 Ölü zamanlı model ve gerçek sistem için tasarlanan PID kontrolörüne ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Ölü zamansız sistemde, Şekil 3.4' den görüldüğü üzere tasarlanan PID kontrolörü için simülasyon cevabı ile gerçek sistem cevabı arasında fark görülmezken, ölü zamanlı sistemde Şekil 3.6' da verildiği gibi aynı PID parametreleri için tasarlanan

kontrolöre ilişkin gerçek sistem cevabı ile üç parametrelili modele ait sistem simülasyon cevabı arasında fark olduğu görülmektedir.

3.3 Dört Parametrelili Model İçin PID Kontrolörü Tasarımı

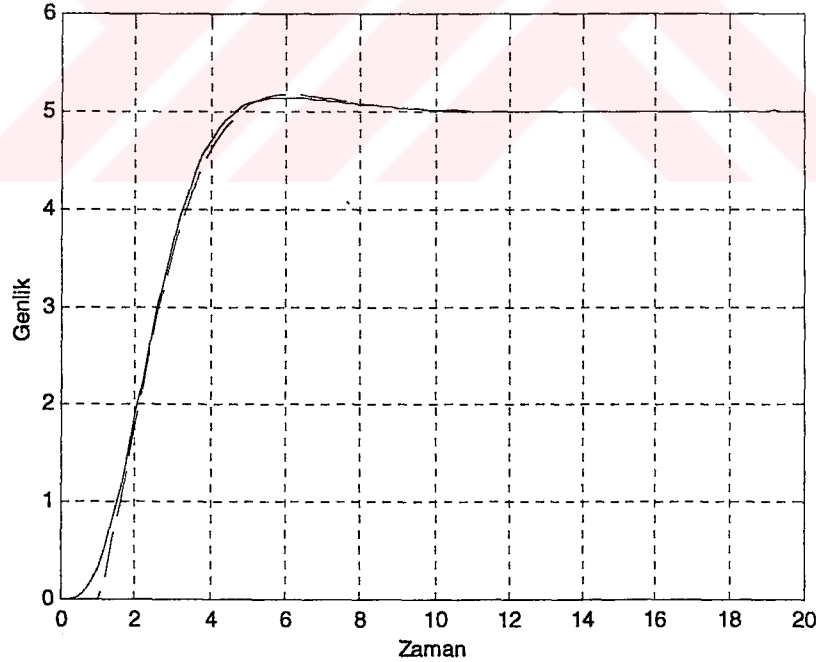
Bölüm 2’ de bulunan dört parametrelili modele ilişkin transfer fonksiyonu

$$G(s) = \frac{1.08}{(1 + 0.2897s)(1 + 0.7s)} e^{-s} \quad (3.9)$$

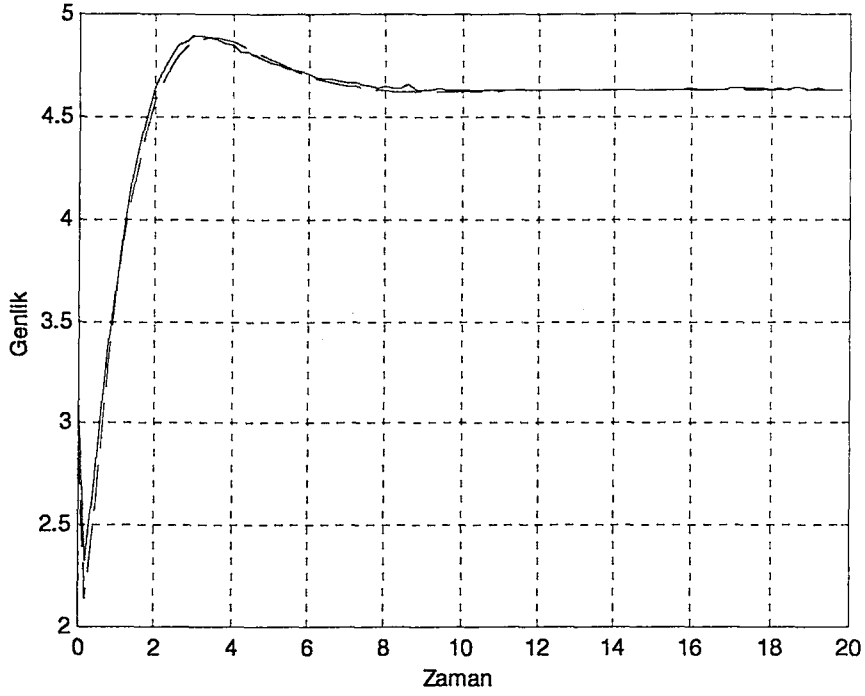
şeklinde elde edilmiştir. $T = 0.2$ saniye için sistemin z tanım bölgesindeki sıfırıncı dereceden tutuculu transfer fonksiyonu

$$G(z) = \frac{0.0777z + 0.05613}{z^5(z^2 - 1.253z + 0.3768)} \quad (3.10)$$

olarak elde edilir. Eşitlik 3.10’ da elde edilen dört parametrelili modele tasarlanan PID kontrolöre ilişkin gerçek sistem ve Matlab simülasyon cevapları Şekil 3.8’ de, bu kontrolöre ait kontrol işaretleri Şekil 3.9’ de görülmektedir.



Şekil 3.8 Dört parametrelili model ve gerçek sistem için tasarlanan PID kontrolörüne ilişkin sistem ve simülasyon cevapları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



Şekil 3.9 Dört parametrelili model ve gerçek sistem için tasarlanan PID kontrolörüne ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

PID kontrolör tasarımı, kontrolörün sıfırları, sistemin kutuplarını silerek sistem mertebesini düşürmek ve kapalı çevrim sistem kutuplarını $z_{1,2} = 0.8667 \pm 0.1j$, $z_{3,4} = 0.196 \pm 0.6j$, $z_5 = -0.45$ ve $z_{6,7} = -0.335 \pm 0.4j$ noktalarına yerleştirerek sistemin küçük aşım ve yerleşme zamanına sahip olması amacıyla $z = 0.5012$ ve $z = 0.7518$ 'e yerleştirilmiştir. Yapılan hesaplamalar sonucunda PID kontrolöre ait parametreler $K_p = 0.35$, $K_i = 0.3864$ ve $K_d = 0.047$ olarak bulunmuştur. PLC programı ve simülasyon programı 0.2 saniye örnekleme zamanıyla 20 saniye boyunca çalıştırılmıştır. Simülasyonda referans değeri olarak 5 genlikli işaret, PLC programında ise 5 genlikli işarete karşılık düşen 16000 kelime değeri alınmıştır. Şekil 3.8' den görüleceği üzere, tasarlanan PID kontrolörüne ilişkin gerçek sistem cevabı ile Matlab simülasyon cevabı birbirine yakın olarak elde edilmiştir.

Üç ve dört parametrelili sistem modelleri için tasarlanan PID kontrolörlerine ait PLC programı Ek B' de verilmiştir.

3.4 Üç Ve Dört Parametrelili Modeller İçin Tasarlanan PID Kontrolör Sonuçlarının İrdelenmesi

Daha önceki bölümde üç parametrelili modelde ölü zamansız ve ölü zamanlı sistem için, dört parametrelili modelde ölü zamanlı sistem için PID kontrolörleri tasarlanmış ve bu kontrolörlere ilişkin gerçek sistem ve Matlab simülasyon yanıtları verilmiştir.

Üç parametrelili ölü zamansız sistem modeli için Şekil 3.4' de verilen sistem yanıtlarında tasarlanan kontrolöre ait gerçek sistem ve simülasyon yanıtları arasında farkın olmadığı görülmektedir. Elde edilen bu sistem modeli gerçek sistem modelini iyi derecede temsil etmektedir.

Şekil 3.6' de, üç parametrelili ölü zamanlı sistem modeli için, tasarlanan kontrolöre ilişkin simülasyon cevabı ve gerçek sistem cevabı arasında bir farkın olduğu gözlenmektedir. Bu fark gerçek sisteme ait modelin, elde ettiğimiz üç parametrelili modelden farklı olması nedeniyle oluşmaktadır. Simülatörde bulunan ölü zaman elemanı sisteme doğrusal olmayan bir özellik eklemektedir. Bu etki, Şekil 2.4' de, basamak yanıtında görülmektedir. Doğrusal olmayan bu etkinin ölü zaman elemanının çeşitli devre elemanlarından tasarlanmış olmasından kaynaklandığı düşünülmüştür. Üç parametre ile modellenen ölü zamansız sisteme kıyasla, üç parametrelili ölü zamanlı sistemin gerçek sistem cevabı kadar iyi yanıt verememesinin sebebinin ölü zaman elemanının gösterdiği bu doğrusal olmayan özelliktir. Gerçekte bulunan üç parametrelili model gerçek sistem modelini iyi şekilde ifade etmemektedir.

Bunun aksine, Şekil 3.8' de verildiği üzere, dört parametrelili modele ait simülasyon yanıtı gerçek sistem yanıtına oldukça yakındır. Dört parametrelili sistem modeli, üç parametrelili modele nazaran bir merteye yüksek olup ikinci dereceden ölü zamanlı model tipidir. Parametre sayısına bağlı olarak sistem derecesi artırılarak daha hassas bir model yapısı elde edilmiştir. Üç parametrelili modelden kaynaklanan karesel hatalar ve ölü zaman elemanının göstermiş olduğu doğrusal olmayan özellik etkisi modelin daha yüksek mertebeli olmasından dolayı indirgenmiş ve gerçek sistem cevabına daha yakın bir sonuç elde edilmiştir.

4. BULANIK KONTROLÖR TASARIMI

Bulanık mantık, sistem dinamiğinin karmaşık yada yüksek derecede doğrusal olmayan özellik gösterdiği proseslerde kullanılır. İlk bulanık mantık algoritması 1974 yılında Mamdani tarafından insan operatörünün dilsel kontrolünü birleştirmek amacıyla oluşturulmuştur (Mamdani, 1974). Klasik kontrolörlerle karşılaştırıldığında, bu tip bulanık mantık kontrolörlerinin (FLC) daha başarılı olmasına rağmen, algoritma önemli derecede operatörün deneyim ve bilgisine dayanmaktadır. Operatör deneyimine bağlılıktan ötürü oluşan bu olumsuzluktan kurtulmak için ilk olarak bulanık mantık kontrolörlerin yapısı için 1976 yılında Mac Vicar-Whelan tarafından bazı genel kurallar önerilmiştir (Mac Vicar-Whelan, 1976). Mac Vicar-Whelan tarafından önerilen bu bulanık kurallar ile, PI ve PD kontrolör yapılarına yaklaşılmış ve hatta oldukça iyi sonuçlar elde edilmiştir (Tang ve Mulholland, 1987).

4.1 Bulanık PID Tipi Kontrolör Yapıları

Literatürde değişik tipte bulanık PID (PI ve PD içeren) kontrolörler önerilmektedir. Tasarımlarına göre, bulanık mantığın PID kontrolörlerine uygulanması temel olarak iki sınıfa ayrılabilir (Xu ve diğ. 2000).

- i. PID kontrolörün kazançları bilgi tabanı ve çıkarım mekanizmasına göre güncel şekilde ayarlanır ve kontrol sinyali PID kontrolörü tarafından üretilir.
- ii. Kontrol kurallarından tipik bir bulanık mantık kontrolörü oluşturulur ve kontrol sinyali Mac Vicar-Whelan yada köşegen kural tabanı oluşturma yaklaşımlarında Mizumoto (1992), Qin ve Borders (1994), Palm ve Driankov (1996), Xu ve diğ. (1998) olduğu gibi bilgi tabanı ve çıkarım mekanizmalarından elde edilir.

İkinci sınıfta bulunan kontrolörler bulanık PID tipi kontrolörler olarak bilinir, çünkü giriş-çıkış ilişkisi bakımından yapıları PID kontrolörü yapılarına benzemektedir. Bu tip kontrolörlerin PID kontrolörlerine göre en önemli kazancı iki adet girişe sahip

olması ve kural tabanının iki boyutlu olmasıdır. Ancak bulanık PD tipi kontrolör ile PD kontrolörü arasındaki denklik bazı özel koşullar altında oluşturulmaktadır (Galichet ve Foulloy, 1995)(Qiao ve Mizumoto, 1996).

Bulanık PI tipi kontrol, bulanık PD tipi kontrole nazaran daha kullanışlı olarak bilinir çünkü bulanık PD tipi kontrol için sürekli hal hatasını ortadan kaldırmak oldukça zordur. Ancak integrasyon nedeniyle, bulanık PI tipi kontrolörün yüksek mertebeli prosesler için geçici hal cevabında düşük başarımlar gösterdiği bilinmektedir. Dolayısıyla teorik olarak bulanık PID tipi kontrol başarımları daha fazla artıracaktır. Bulanık PID tipi kontrolör için gerçekte 3-D kural tabanı gerekmektedir ancak uzman operatörün algılama kapasitesi açısından bu oldukça güçtür. Bu sebepten ötürü oran, integral ve türev kontrol elemanlarını biraraya getirerek PI ve PD etkilerini birleştirmek yoluyla bulanık PID tipi kontrolör elde etmek mümkündür (Kwok ve diğ. 1990)(Li ve Gatland, 1996)(Qiao ve Mizumoto, 1996)(Golob, 2001)(Güzelkaya ve diğ. 2001).

Bulanık PID tipi kontrolörü ya bulanık PI ve PD tipi kontrolörleri iki ayrı kural tabanı ile birleştirerek ya da bir bulanık PD tipi kontrolörünün çıkışına integral ve toplam elemanı koyarak elde etmek mümkündür (Qiao ve Mizumoto, 1996).

Kontrolör için tasarım parametreleri iki ayrı grupta toplanabilir (Hu ve diğ. 1999).

- i. Yapısal parametreler
- ii. Ayarlama parametreleri

Temel olarak, yapısal parametreler bulanık çıkarım için giriş / çıkış (I/O) değişkenleri, bulanık dilsel kümeleri, üyelik fonksiyonları, bulanık kuralları, çıkarım mekanizmasını ve durulama mekanizmasını içerir. Ayarlama parametreleri ise giriş / çıkış (I/O) ölçekleme faktörlerini (SF) ve üyelik fonksiyon parametrelerini (MF) içerir. Genellikle yapısal parametreler güncel olmayan (off-line) biçimde ayarlanırken, ayarlama parametreleri sistem belirsizliğine ve proses gürültüsüne karşı uyarılma kapasitesinin düzenlenmesi ve dolayısıyla proses başarımlarını artırmak amacıyla güncel (on-line) olarak ayarlanır. Bu özelliklerden dolayı bulanık PID tipi kontrolörler özayarlamasız ve özayarlamalı kontrolörler olarak iki sınıfta toplanabilir.

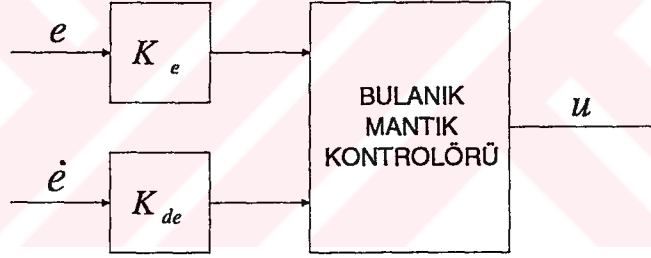
4.1.1 Özayarlamasız Bulanık PID Tipi Kontrolörler

Tasarlanacak bulanık kontrolörünün hata (e) ve hatanın değişimi ($\dot{e}$) olmak üzere iki girişinin olması, çıkarım mekanizması olarak çarpım-toplam çıkarım mekanizmasının kullanılması ve hata ile hatanın değişiminin her bir dilsel değeri için üçgen tip üyelik fonksiyonunun kullanılması durumunda kontrolöre ait kontrol işareti

$$u = A + K_e Pe + K_{de} D \dot{e} \quad (4.1)$$

şeklinde elde edilir (Qiao ve Mizumoto, 1996).

Elde edilen kontrol işaretine bakıldığında tasarlanan bulanık kontrolörün yaklaşık olarak bir PD kontrolörü gibi davrandığı görülmektedir. Bu sebepten ötürü bu tip kontrolörlere bulanık PD tipi kontrolörler adı verilir. Bu tip kontrolöre ilişkin blok gösterimi Şekil 4.1' de görülmektedir.

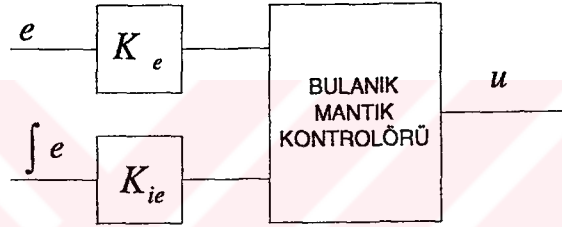


Şekil 4.1 Bulanık PD tipi kontrolöre ait blok gösterimi

Bir PID kontrolörün başarımı, kontrolöre ait K_p oransal parametresi, K_i integral parametresi ve K_d türev parametresi tarafından belirlenir. Oransal kontrol sistem cevabının hızlı olmasını garanti eder, integral kontrolü sistem yanıtında oluşacak olan sürekli hal hatasını ortadan kaldırır, türev kontrolü ise sistem yanıtının sönümünü artırır, böylece aşımı ve osilasyonu azaltır. Dolayısıyla PID kontrolörü eğer düzgün olarak tasarlanırsa hızlı yükselme zamanlı, düşük aşımlı, sürekli hal hatası olmayan bir sistem yanıtına neden olur. Eğer kontrol edilen sistem tipi sıfır ise bir P yada PD kontrolörü sistem yanıtında sürekli hal hatasının oluşmasına sebep olur.

Bulanık PD tipi kontrolörde yaklaşık olarak bir PD kontrolörü gibi davranacağından, eğer sistem tipi sıfır ise o zaman bulanık PD tipi kontrolörde sistem yanıtında sürekli hal hatasının oluşmasına neden olacaktır. Bu nedenden ötürü tıpkı PD kontrolörü gibi, bulanık PD kontrolörü için de kontrol başarımı tatmin edici olmayacaktır. Ancak sistem başarımı bulanık kontrolöre integral kontrolünün eklenmesi ile artırılabilir.

Sistem yanıtında oluşacak sürekli hal hatasını ortadan kaldırmak maksadıyla integral kontrolü bulanık kontrolöre dahil etmenin ilk yolu, bulanık PD tipi kontrolörün girişlerinden olan hatanın değişimi yerine hatanın integralini koymak olacaktır. Ancak bu yöntem bulanık kontrolörün, parametresi zamanla değişen bir PI kontrolörü gibi davranmasına neden olacaktır. Bu tip kontrolöre ait blok gösterim Şekil 4.2' de görülmektedir.



Şekil 4.2 Girişlerin hata ve hatanın integrali olarak alınması durumunda elde edilen bulanık kontrolöre ait blok gösterimi

Elde edilen bu tip kontrolörle sürekli hal hatası ortadan kaldırılmaktadır ancak oransal kontrol parametresinin küçük seçilmesi durumunda sistem yanıtı yavaş yükselme zamanına ve oransal yada integral kontrol parametrelerinin büyük seçilmesi durumunda ise sistem yanıtı büyük bir aşımaya sahip olacaktır. Ayrıca sistemi kontrol eden operatör için kontrolör girişinde bulunan hatanın integralini takip etmek pek mümkün değildir. Dolayısıyla kontrolör girişlerini hata ve hatanın değişimi olarak ve kontrolör çıkışının integralini alarak kontrol işaretinin elde edilebileceği yeni bir kontrolör tasarlanabilir. Bu tip kontrolöre ait blok gösterimi Şekil 4.3' de görülmektedir.



Şekil 4.3 Bulanık PI tipi kontrolöre ait blok gösterimi

Tasarlanan bu tip kontrolöre ait kontrol işareti

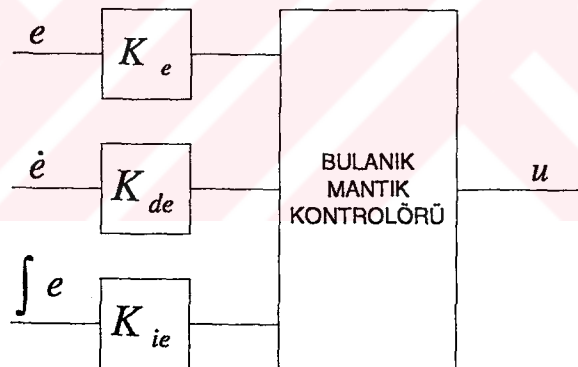
$$u = \beta A t + \beta K_{de} D e + \beta K_e P \int e dt \quad (4.2)$$

şeklinde olacaktır (Qiao ve Mizumoto, 1996).

Elde edilen kontrol işaretine bakıldığında kontrolörün aslında parametresi zamanla değişen PI kontrolörü gibi davrandığı görülmektedir. Bu tip kontrolörlere bulanık PI tipi kontrolörler adı verilir.

Bulanık PD tipi kontrolör kullanılması sonucunda ortaya çıkan sürekli hal hatası ve bulanık PI tipi kontrolör kullanılması durumunda karşılaşılabilecek büyük aşım ve yavaş yükselme zamanı gibi olumsuzluklar ancak integral ve türev kontrollerrinin birarada kullanılması ile ortadan kaldırılabilir. Böylece sistem başarımı önemli derecede artırılmış olur.

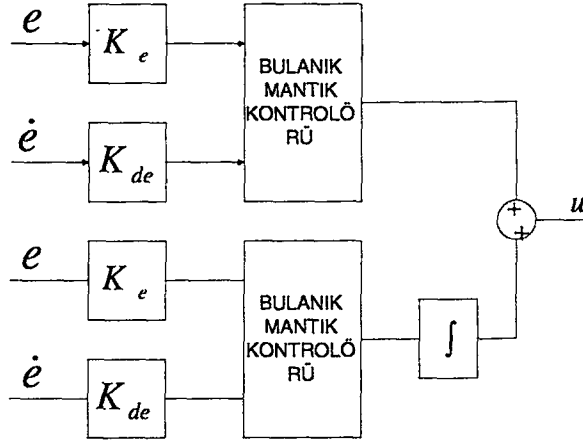
Türev ve integral kontrolün birleştirilmesi, kontrolör girişinde hata, hatanın değişimi ve hatanın integralinin kullanılması ile gerçekleştirilebilir. Bu tip kontrolöre ait blok gösterimi Şekil 4.4' de görülmektedir.



Şekil 4.4 Girişlerin hata, hatanın değişimi ve hatanın integrali olarak alınması durumunda elde edilen bulanık kontrolöre ait blok gösterimi

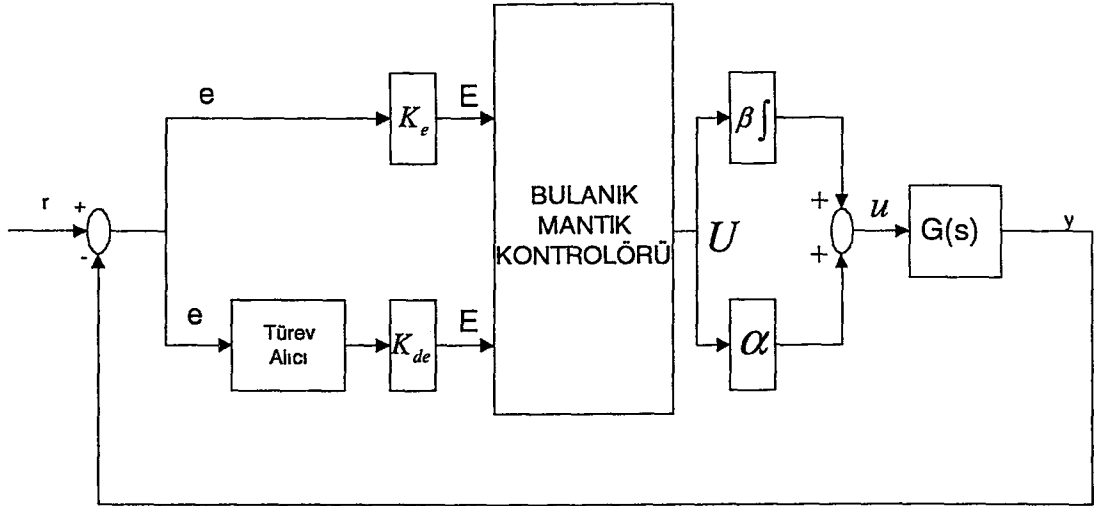
Ancak bu yöntem uygulamada oldukça zordur, çünkü kontrolör girişinde kullanılan hatanın integralinin sistemi kontrol eden operatör tarafından takip edilip kontrol işleminde düzenlemeler yapılması çok güçtür. Buna ilave olarak kontrolöre ait giriş sayısının ikiden üçe çıkarılması ile beraber bulanık kontrolör yapısında oluşturulacak kural sayısında da önemli derecede artış olacaktır. Örneğin, iki adet giriş için herbir girişe ait beş adet dilsel değişken kullanılması durumunda toplam 25 adet kural yazılması gerekirken, kontrolör giriş sayısının üçe çıkarılması ile beraber kural sayısı

da 125 olacaktır. Kontrolör tasarımı karşılaşılabilecek bu iki önemli olumsuzluktan kurtulmak için Şekil 4.5’ de verilen kontrolör yapısı oluşturulabilir.



Şekil 4.5 Hata ve hatanın değişimi olmak üzere iki adet giriş ve iki ayrı kural tabanına sahip bulanık kontrolöre ait blok gösterimi

Şekil 4.5’ den de görüldüğü üzere iki ayrı kural tabanına sahip bulanık PD ve bulanık PI tipi kontrolörlerin birleştirilmesi ile yeni bir bulanık kontrolör elde edilmektedir. Kontrolörlerin girişlerinde hata ve hatanın değişimi alınarak girişte hatanın integralini takip etme durumu ortadan kaldırıldığı gibi bulanık mantık kontrolöründe oluşturulacak kural sayısında, yine her bir girişe ait beş dilsel değişken kullanılması durumunda bulanık PD tip kontrolör için 25 ve bulanık PI tip kontrolör için 25 olmak üzere toplamda 50’ ye indirilmektedir. Ancak bu yapıdan daha kullanışlı başka bir kontrolör yapısı Şekil 4.6’ da görülmektedir.



Şekil 4.6 Bulanık PID tipi kontrolöre ait blok gösterimi

Şekil 4.6' dan da görüldüğü gibi önce bir bulanık PD tipi kontrolör, daha sonra kontrolör çıkışının integrali alınarak ve β katsayısı ile çarpılarak bir bulanık PI tipi kontrolör oluşturulmakta ve bu kontrolör elde edilen bulanık PD tipi kontrolörün α değeri ile çarpımı ile toplanarak yeni bir bulanık kontrolör yapısı oluşturulmaktadır. Elde edilen kontrol işareti

$$u = \alpha A + \beta A t + (\alpha K_e P + \beta K_{de} D)e + \beta K_e P \int e dt + \alpha K_{de} D \dot{e} \quad (4.3)$$

şeklindedir (Qiao ve Mizumoto, 1996).

Kontrol işaretine bakıldığında, tasarlanan kontrolörün parametresi zamanla değişen PID kontrolörü gibi davrandığı görülmektedir. Bu tip kontrolörlere bulanık PID tipi kontrolörler adı verilir. Kontrolör yapısına bakıldığında aslında bulanık PD ve PI tipi kontrolörlerin paralel olarak bağlanmasıyla bulanık PID tipi kontrolör elde edilmektedir.

Burada K_e ve K_{de} 'ye giriş ölçekleme faktörleri, α ve β 'ya ise çıkış ölçekleme faktörleri adı verilmektedir. Bu tip kontrolörde, kontrolör girişleri hata ve hatanın değişimi alınmakta böylece Şekil 4.4 ve Şekil 4.5' te verilen kontrolör tiplerine nazaran daha az sayıda bulanık kurala sahip olunmaktadır. Örneğin her bir girişe ait beş adet dilsel değişken tanımlanması durumunda toplam 25 adet kural yazılması gerekmektedir. Ancak burada dikkat edilmesi gereken önemli bir nokta, Şekil 4.5' te verilen kontrolör tipiyle karşılaştırıldığında, bu kontrolör daha az sayıda ölçekleme faktörüne sahiptir. Dolayısıyla burada ölçekleme faktörlerinin ayarlanması daha büyük bir önem taşımaktadır. Bundan sonraki bölümlerde elde edilen modellere bulanık PID tipi kontrolör tasarımında Şekil 4.6' da verilen kontrolör yapısı kullanılacaktır.

4.1.2 Özayarlama Bulanık PID Tipi Kontrolörler

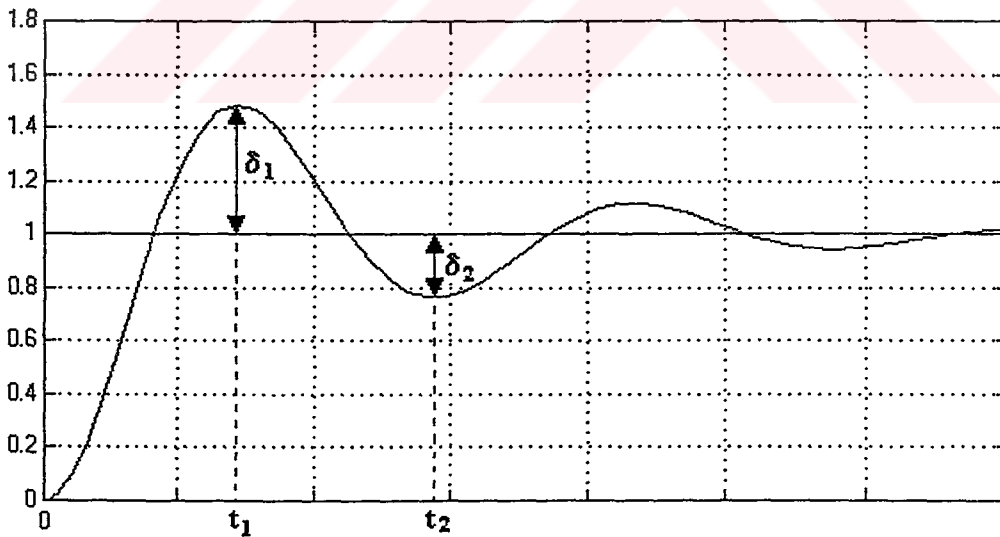
Bulanık kontrolör tasarımı üç önemli safhadan oluşmaktadır. Bunlar; bilgi tabanı tasarımı, ayarlama parametreleri ve üyelik fonksiyonları olarak sıralanabilir. Eğer bulanık kontrolörün istenen amacı başarılması isteniyorsa o zaman bulanık kontrolöre ait bu üç safhanın ayarlanması gerekmektedir. Ancak gerçek zaman kontrolü için sadece ölçekleme faktörlerinin ayarlanması yeterli olmaktadır. Dolayısıyla bulanık kontrol uygulaması için bulanık kontrolöre ait ölçekleme faktörlerinin güncel olarak

ayarılanması önemli bir ihtiyaçtır. Bulanık PID tipi kontrolörlerin ölçekleme faktörlerinin ayarlanmasına ilişkin bir çok çalışma yapılmıştır (Ollero ve Garcia-Cerezo, 1989)(Peng, 1990)(Vega ve diğ. 1991).

Bulanık PID tipi kontrolörlerin başarımını artırmak maksadıyla bazı özayarlamalı mekanizmalar yapılan çalışmalarda önerilmiştir. Buna ilişkin üç yöntem aşağıda daha detaylı olarak verilmektedir.

4.1.2.1 Aşım Gözleyicisi İle Parametre Uyarlamalı Yöntem

Bir önceki bölümde elde edilen bulanık PID tipi kontrolör gerek bulanık PD gerekse bulanık PI tipi kontrolörlerin başarımını artırmıştır. Bulanık PID tipi kontrolörde eğer integral teriminin değeri küçükse o zaman sistem yanıtı yavaş olmakta, eğer integral teriminin değeri büyükse o zaman da sistem kararsızlığa gidebilmektedir. Bu halde kontrolöre ait integral teriminin bahsedilen bu kazanç ve olumsuzlukları gözönünde tutularak bu terim üzerinde iyileştirmeler yapılabilir. Örneğin bulanık kontrolörün integral bileşeninin değeri zamanla değişmektedir. Dolayısıyla sistem yanıtının erken safhalarında terimin değeri büyük seçilebilir ve zamanla değer küçültülerek sistemin sönümü artırılabilir, sistem daha kararlı hale getirilebilir. Bu yolla sistem yanıtında hızlı yükselme ve kısa yerleşme kriterleri sağlanabilir.



Şekil 4.7 Bir kontrol sistemi basamak cevabının değişik evreleri

Şekil 4.7' de genel olarak bir kontrol sisteminin basamak yanıtı görülmektedir. Sistem yanıtı tepe değerlerinin oluşturduğu zaman değerleri ile çeşitli safhalara ayrılabilir. 0- t_1 zaman aralığında hata değeri geniş bir değer aralığını ifade etmekte,

t_1 zamanından sonra hata değeri $[\delta_1, -\delta_1]$ değer aralığı dışına çıkmamakta (δ_1, t_1 anında meydana gelen mutlak tepe değeridir), t_2 anında yeni bir tepe değeri elde edilmekte ve t_2 anından sonra hata değeri $[\delta_2, -\delta_2]$ aralığının dışına çıkmamakta ve oluşan tepe değerleri ile hata aralıkları böylece devam etmektedir. O zaman integral teriminin değeri her tepe değerinde mutlak tepe değerine bağlı olarak azaltılabilir.

Daha önce Eşitlik 4.3’ de verilen bulanık PID tipi kontrolöre ait kontrol işaretinin oransal, integral ve türev kontrol bileşenlerinin değerleri sırasıyla

$$\alpha K_e P + \beta K_{de} D \quad (4.4a)$$

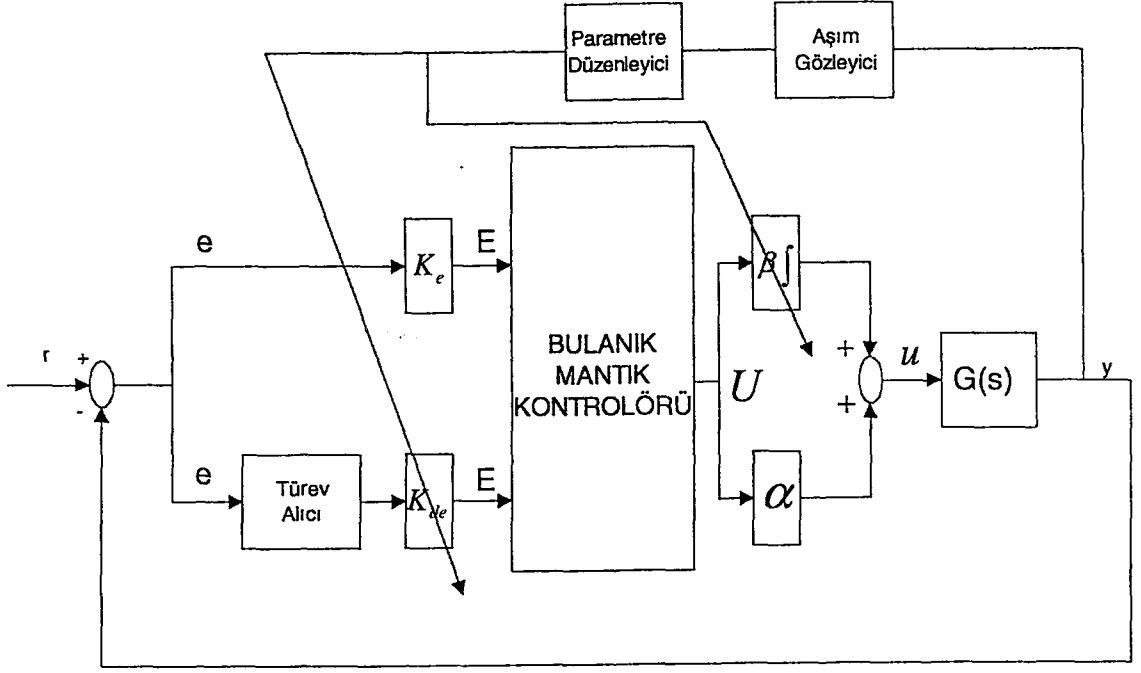
$$\beta K_e P \quad (4.4b)$$

$$\alpha K_{de} D \quad (4.4c)$$

şeklinde yazılabilir.

Eğer β ölçekleme faktörü değeri kademe kademe düşürülürse, integral teriminin değeri de azalacak ve buna karşın sistemin sönümü artacak, sistem daha kararlı hale gelecektir. Ancak burada dikkat edilmesi gereken önemli bir nokta, kontrolöre ait oransal kontrol bileşeninin de β parametresine sahip olmasıdır. Başka bir deyişle β parametre değerinin düşürülmesi ile beraber oransal kontrol bileşeninin değeri de azalacaktır ve böylelikle sistemin hataya karşı olan tepkisinde yavaşlayacaktır. Bu olumsuzluğu ortadan kaldırmak için K_{de} ölçekleme faktörünün değeri β ’nın azalma oranıyla doğru orantılı olarak artırılmalıdır. Dolayısıyla oransal kontrol şiddeti değişmeyecektir ve sistem hataya karşı olan hızlı tepkisini her zaman korumuş olacaktır. Ayrıca K_{de} ölçekleme faktörü değerinin artması ile türev kontrolü bileşeninin değeri de artacaktır ancak bu artış sistem başarımına olumsuz bir etkide bulunmaz çünkü türev kontrolü, meydana gelebilecek salınım ve aşım karşı sistemin direncini artırmaktadır.

Bu düşüncelerden yola çıkılarak aşım gözleyicisi kullanan parametre uyarlamalı bulanık PID tipi kontrolör tasarlanmıştır (Qiao ve Mizumoto, 1996). Bu kontrolöre ait blok gösterimi Şekil 4.8’ de verilmektedir.



Şekil 4.8 Aşım gözleyicisi ile parametre uyarlamalı bulanık PID tipi kontrolörün kapalı çevrim kontrol yapısı

Şekil 4.8' den görüldüğü üzere, aşım gözleyicisi ile parametre uyarlamalı bulanık PID tipi kontrolör temel olarak bir aşım gözleyicisi, bir parametre düzenleyicisi ve bir bulanık PID tipi kontrolörden oluşmaktadır.

Aşım gözleyicisi sistem çıkışını sürekli olarak gözler, her tepe zamanında bir işaret üretir ve mutlak tepe değerini ölçer. Parametre düzenleyicisi, tepe değerine bağlı olarak her tepe değerinde eşzamanlı olarak β ve K_{de} parametre değerlerini düzenler. Bu parametrelerin ayarlanmasına ilişkin algoritma

$$K_{de} = \frac{K_{des}}{\delta_k} \quad (4.5)$$

$$\beta = \delta_k \beta_s \quad (4.6)$$

şeklindedir. Burada K_{des} ve β_s , K_{de} ve β 'nın başlangıç değerleri ve δ_k tepe zamanlarındaki mutlak tepe değeridir.

4.1.2.2 Fonksiyon Ayarlayıcısı İle Parametre Uyarlamalı Yöntem

Eğer bulanık PID tipi kontrolöre ait bulanık kurallarda ve ölçekleme faktörlerinde değişiklik yapılmaz ve sadece üyelik fonksiyonlarında birtakım ayarlamalar yapılırsa o zaman sadece sistem yanıtının sürekli hal cevabında iyileştirmeler yapılabilir, buna karşın sistem yanıtının geçici hal cevabında iyileştirme yapılamaz. Dolayısıyla sistemin gerek sürekli hal gerekse geçici hal cevabında iyileştirme yapılabilmesi için ölçekleme faktörlerinin ayarlanması kaçınılmazdır. Bu amaçla iki adet deneysel fonksiyon

$$f(e(t)) = a_1 \times \text{abs}(e(t)) + a_2 \quad (4.7)$$

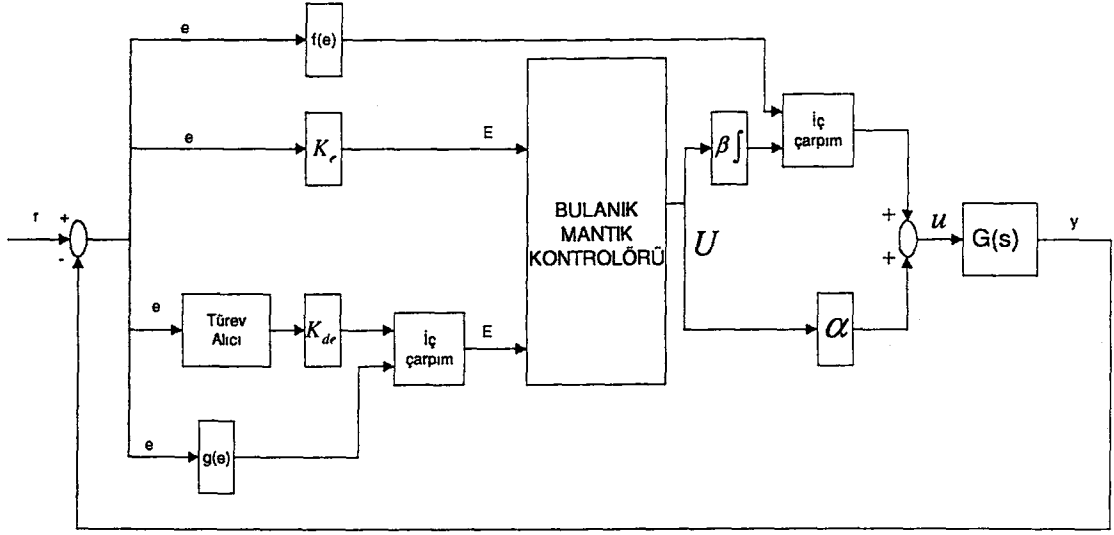
$$g(e(t)) = b_1 \times (1 - \text{abs}(e(t))) + b_2 \quad (4.8)$$

şeklinde tanımlanmaktadır (Woo ve diğ. 2000). Burada a_1 , a_2 , b_1 , b_2 pozitif sabitlerdir. Zamanla değişen ölçekleme faktörlerinin ayarlanmasına ilişkin algoritma

$$\beta(e(t)) = \beta_s \times f(e(t)) \quad (4.9)$$

$$K_{de}(e(t)) = K_{des} \times g(e(t)) \quad (4.10)$$

şeklindedir. Burada β_s ve K_{des} sırasıyla β ve K_{de} ölçekleme faktörlerinin başlangıç değerleridir. Eşitlik 4.9' dan da görüldüğü gibi, $f(e(t))$ fonksiyonunun amacı $\beta(e(t))$ ölçekleme faktörü değerini hatanın zamanla değişimine bağlı olarak azaltmaktır. Bunun aksine $g(e(t))$ fonksiyonunun amacı $K_{de}(e(t))$ ölçekleme faktörü değerini hatanın zamanla değişimine bağlı olarak artırmaktır. Sistem yanıtının başlangıç anında hatanın değeri maksimum olacağından $f(e(t))$ fonksiyonunun değeri de maksimum olacaktır, buna karşın $g(e(t))$ fonksiyonunun değeri minimum olacaktır. Ancak zamanla hatanın değeri küçüleceğinden $f(e(t))$ fonksiyonunun değeri azalırken $g(e(t))$ fonksiyonunun değeri artacaktır. Böylelikle sistem yanıtının erken safhalarında bulanık PID kontrolöre ait integral kontrol bileşeninin değeri büyük olacak ve zamanla bu değer azalarak sistemin sönümü artarak sistem daha kararlı hale gelecektir. Bu tip kontrolöre ait blok gösterimi Şekil 4.9' da görülmektedir.



Şekil 4.9 Fonksiyon ayarlayıcısı ile parametre uyarlamalı bulanık PID tipi kontrolörün kapalı çevrim kontrol yapısı

Her iki fonksiyonda verilen pozitif katsayıların tespiti oldukça önemlidir. Bu katsayıların belirlenmesine ilişkin bir yöntem fonksiyonların başlangıç ve sürekli hal anında alacağı değerlerden belirlenmesidir. Örneğin referans değerinin "1" olması durumunda katsayıların belirlenmesi ilişkin yöntem şu şekilde olacaktır:

Başlangıç anında hata değeri "1" ve $f(1) = a_1 + a_2$, $g(1) = b_1 + b_2$ buna karşın sürekli hal anında hata değeri "0" ve $f(0) = a_2$, $g(0) = b_1 + b_2$ değerleri elde edilecektir. Dolayısıyla bulanık PID tipi kontrolörün girişe ait ölçekleme faktörü değeri başlangıç anında $\beta(e(t)) = \beta_s(a_1 + a_2)$ ve sürekli hal anında $\beta(e(t)) = \beta_s a_2$, çıkışa ait ölçekleme faktörü değeri başlangıç anında $K_{de} = K_{des} b_2$ ve sürekli hal anında $K_{de} = K_{des}(b_1 + b_2)$ olacaktır. Böylelikle $\beta(e(t))$ 'nin başlangıç $\beta_s(a_1 + a_2)$ ve son $\beta_s a_2$ değerlerini belirlemek yoluyla a_1 ve a_2 katsayı değerleri ayarlanabilir. Benzer şekilde $K_{de}(e(t))$ 'nin başlangıç $K_{des} b_2$ ve sonuç $K_{des}(b_1 + b_2)$ değerlerini belirlemek yoluyla da b_1 ve b_2 katsayı değerleri ayarlanabilir. Ayrıca fonksiyonlara ait katsayıların değerleri genetik arama algoritması ya da benzer algoritmalarla da belirlenebilir.

4.1.2.3 Bağıl Hız Gözleyicisi İle Parametre Uyarlamalı Yöntem

Bilindiği gibi PID tipi kontrolörlerde kontrolöre ait integral bileşeninin değeri küçük seçilirse sistem yanıtı yavaş bir yükselmeye sahip olacaktır. Buna karşın integral parametresi büyük seçilirse bu durumda da sistem yanıtı kararsız hale gelebilecek kadar hızlı bir yükselmeye sahip olacaktır. Ayrıca PID kontrolörünün türev bileşeni sistem yanıtını yavaşlatacak şekilde davranır. Dolayısıyla eğer parametreleri zamanla değişen PID kontrolörü gerçekleştirilirse o zaman sistem yanıtının erken safhalarında büyük bir integral terimi değerine ve küçük bir türev terimi değerine sahip olunabilir ve sistem yanıtı istenilen referans değerine yaklaştıkça integral terimi değeri kademe kademe düşürülebilir, buna karşın türev terimi değeri de integral değerinin düşürülmesi ile doğru orantılı olarak artırılabilir. Zaman içinde PID kontrolörünün parametrelerinin ayarlanmasına ilişkin bu düşünce ancak sisteme ait hata değeri dikkate alındığında gerçekleştirilebilir.

Eğer bir şekilde sistem yanıtının hızı hakkında bilgi edinilebilirse o zaman PID kontrolörüne ait parametrelerin ayarlanması az önce bahsedilen sistem yanıtının konumuna bağlı parametre ayarlamasından bir miktar farklı olacaktır. Örneğin sistem yanıtının hızı "çok yavaş" ve sistem yanıtı referans değerine "çok yakın" iken integral teriminin değeri azaltılmayacaktır. Halbuki sadece sisteme ait hata değeri yada sistem yanıtının konumu dikkate alındığında integral değeri azaltılacaktır. Bu durumun aksini düşündüğümüzde sistem yanıtı "çok hızlı" ve referans değerine "çok uzak" iken kontrolöre ait integral terimi azaltılırken, sisteme ait hata yada sistem yanıtının konumu dikkate alındığında integral teriminin değeri azaltılmayacaktır.

Daha önceki bölümlerde anlatılan aşım gözleyicisi ile ve fonksiyon ayarlayıcısı ile parametre uyarlanması sadece sisteme ait hata değerine bağlıydı. Ancak bu yöntem sisteme ait hata değeri ve sistem yanıtının hızı bilgilerini beraberce kullanarak bulanık PID tipi kontrolöre ait ölçekleme faktörlerini ayarlamaktadır. Bulanık PID tipi kontrolöre ait integral ve türev katsayıları bulanık bir çıkarım mekanizması kullanılarak güncel biçimde ayarlanmaktadır. İntegral ve türev terimlerine ait katsayıları ayarlayan çıkarım mekanizmasının " e " olarak belirtilen "sistem hatası" ve " r_v " olarak adlandırılan yeni bir değişken olan "bağıl hız" olmak üzere iki adet girişi mevcuttur (Eksin ve diğ. 2001). Bağıl hız sistem yanıtının hızlılığı yada yavaşlılığı hakkında bilgi vermektedir. Bu sebepten ötürü bu bilgiyi sağlayan

mekanizmaya "Bağıl hız gözleyicisi" (BHG) adı verilir. Ayarlama mekanizmasına sahip kontrolör yapısına geçmeden önce bağıl hız hakkında bilgi vermek uygun olacaktır.

Bağıl hız, $r_v(k)$

$$r_v(k) = \frac{de(k) - de(k-1)}{de(.)} = \frac{dde(k)}{de(.)} \quad (4.11)$$

şeklinde ifade edilir. Burada $de(k)$ hata değerindeki değişim olarak bilinir ve

$$de(k) = e(k) - e(k-1) \quad (4.12)$$

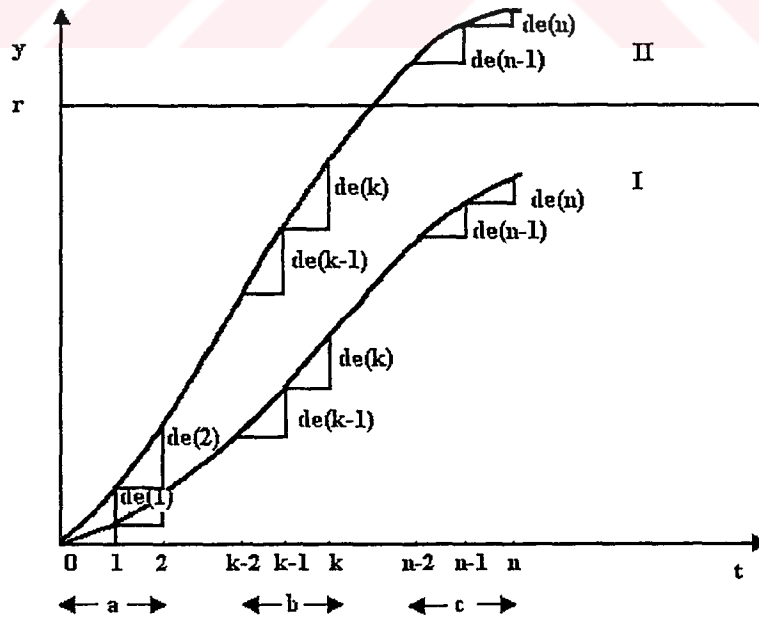
olarak verilir. Eşitlik 4.11' de verilen $dde(k)$ ise hata değerindeki hızlanmadır ve

$$dde(k) = de(k) - de(k-1) \quad (4.13)$$

biçiminde ifade edilir. Eşitlik 4.11' de verilen $de(.)$ ise

$$de(.) = \begin{cases} de(k) & \Leftrightarrow |de(k)| \geq |de(k-1)| \\ de(k-1) & \Leftrightarrow |de(k)| < |de(k-1)| \end{cases} \quad (4.14)$$

şeklinde ifade edilir.



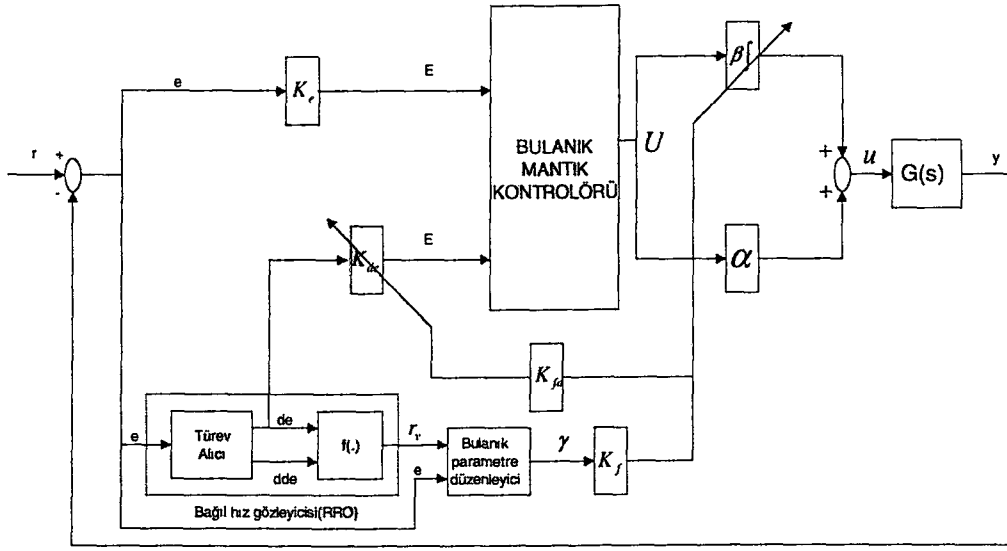
Şekil 4.10 Basamak girişi için sistem yanıtlarına ait bağıl hızların gösterimi

Sistem yanıtı yumuşak ve kararlı artma yada azalmaya sahip olduğu zaman $de(k)de(k-1)$ çarpımı pozitif olacaktır ve sistem yanıtının "hızlılık" ve "yavaşlık" bilgisi yeni değişken r_v kullanılarak belirlenebilecektir. Bağlı hız kavramının önemli ve ilginç bir özelliği Şekil 4.10' dan da görüldüğü üzere farklı zaman sabitlerine sahip iki sistem yanıtı eğrilerinin aynı $r_v(k)$ değerlerine sahip olmasıdır. Eğer hatadaki değişimin mutlak değeri bir önceki değer olan $|de(k-1)|$ ' den büyük ise o zaman sistem yanıtı "hızlı" biçimde artmakta yada azalmaktadır. Bunun aksine hatadaki değişimin mutlak değeri bir önceki değer olan $|de(k-1)|$ ' den küçük ise o zaman da sistem yanıtı "yavaş" biçimde artmakta yada azalmaktadır. Tablo 4.1 $dde(k)$ ve $de(.)$ işaretlerine bağlı olarak sistem yanıtının "hızlılık" yada "yavaşlık" durumu hakkındaki bilgileri göstermektedir.

Tablo 4.1 $De(.)$ ile $dde(k)$ işaretleri arasındaki ilişki ve sistem yanıtının yapısı

$de(k-1)$ yada $de(k)$	$dde(k)$	Sistem yanıtı
Pozitif	Pozitif	Hızlı
Pozitif	Negatif	Yavaş
Negatif	Pozitif	Yavaş
Negatif	Negatif	Hızlı

Eşitlik 4.11' de verilen bağlı hız denklemi sistem yanıtının hızına ilişkin $[-1, 1]$ aralığında bir değer üretir. Eğer sistem yanıtı çok hızlı ise o zaman $r_v(k)$ "1" değerine ve eğer sistem yanıtı çok yavaş ise $r_v(k)$ "-1" değerine yakınsar. Sistem yanıtı sabit bir hızla azalıyor yada artıyorsa hızlılığın orta seviyede olduğu anlaşılır ve bu kez $r_v(k)$ "0" değerine yakınsar. Bağlı hız gözleyicisi ile parametre uyarlamalı yöntemle ilişkin blok gösterimi Şekil 4.11' de görülmektedir.



Şekil 4.11 Bağıl hız gözleyicisi ile parametre uyarlamalı bulanık PID tipi kontrolörün kapalı çevrim kontrol yapısı

Bulanık parametre düzenleyicisinin çıkışı γ olarak ifade edilir. K_{de} ölçekleme faktörü, bu ölçekleme faktörünün başlangıç değeri ile γ değerinin çarpımıyla ayarlanırken, β ölçekleme faktörü, bu ölçekleme faktörünün başlangıç değeri ile γ değerinin bölümüyle ayarlanır. Ölçekleme faktörlerinin ayarlanmasına ilişkin algoritma

$$K_{de} = K_{des} \times K_{fd} \times K_f \times \gamma \quad (4.15)$$

$$\beta = \frac{\beta_s}{K_f \times \gamma} \quad (4.16)$$

şeklinde (Güzelkaya ve diğ.). Algoritmada verilen K_{des} , K_{de} ölçekleme faktörünün başlangıç değeri, β_s , β ölçekleme faktörünün başlangıç değeri, K_f , bulanık parametre düzenleyicisi bloğunun çıkış ölçekleme faktörü ve K_{fd} ise sadece bulanık PID tipi kontrolörün türev terimini etkileyen ilave parametredir. Eşitlik 4.15 ve Eşitlik 4.16' dan da görüldüğü üzere eğer K_{fd} parametre değeri "1" seçilirse o zaman kontrolörün oransal kontrol şiddeti değişmez. γ değerini belirlemede kullanılabilecek kural tabanı aşağıdaki biçimde verilebilir. Burada girişlere üç ve çıkışa dört adet dilsel ifade atanmış olup bu değerler kontrolör tasarımcısına bağlı olarak artabilir ya da azalabilir.

γ değerini belirlemede kullanılan yarı-kurallar;

- Sistem yanıtı yavaş olduğu zaman bulanık PID tipi kontrolörün türev etkisi azaltılmalı
- Hata küçük ve sistem yanıtı hızlı olduğu zaman bulanık PID tipi kontrolörün türev etkisi artırılmalı

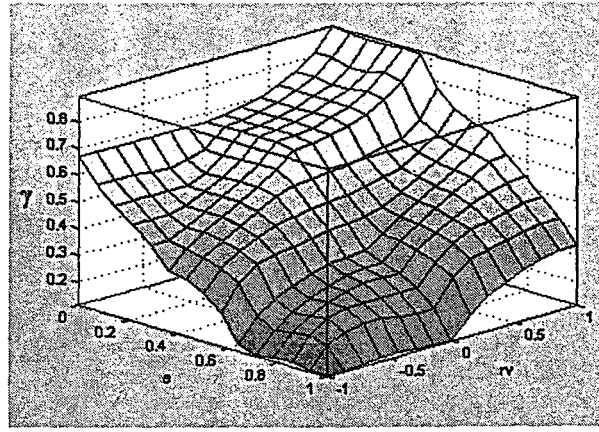
biçiminde ifade edilir. Tablo 4.2' de verilen kural tabanı yukarıda belirtilen yarı kurallar kullanılarak oluşturulmuştur. Bulanık kural tabanının girişleri hatanın mutlak değeri $|e|$ ve bağıl hız değeri r_v ' dir. $|e|$ ve r_v giriş değişkenleri ile γ çıkış değişkenine atanan dilsel ifadeler;

L: büyük; M: orta; SM: küçük orta; S: küçük; F: hızlı; M: orta; S: yavaş şeklindedir.

Tablo 4.2 Çıkış değişkeni (γ) için kural tabanı

$ e \backslash r_v$	S	M	F
S	M	M	L
SM	SM	M	L
M	S	SM	M
L	S	S	SM

Giriş değişkenleri ile çıkış değişkenine düzgün dağılımlı simetrik üçgensel üyelik fonksiyonları atanmıştır. Yukarıda verilen kural tabanına ait yüzey görünümü Şekil 4.12' de verilmiştir.



Şekil 4.12 Çıkış değişkeni (γ) için kural tabanı görünümü

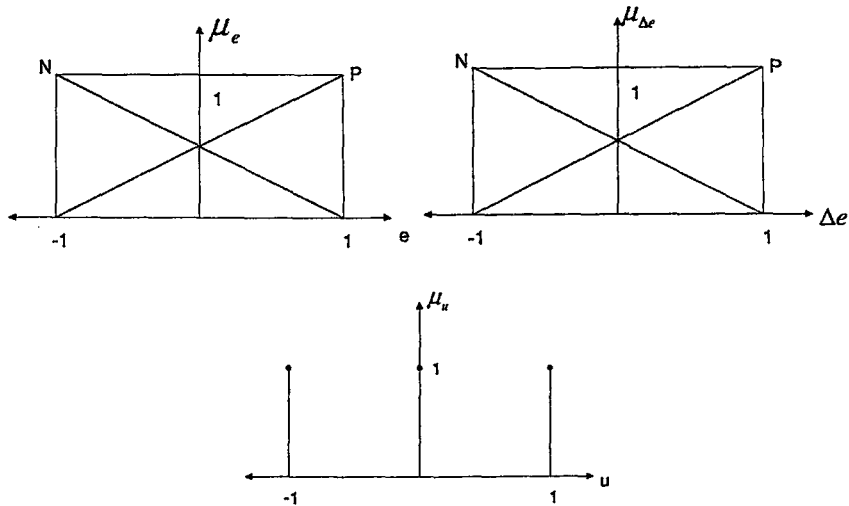
4.2 Özayarlamaşız Bulanık PID Tipi Kontrolör Tasarımı

Özayarlamaşız bulanık PID tipi kontrolör tasarımında Bölüm 2' de de verildiği üzere, ölü zamansız sistem için sistem transfer fonksiyonu $G(s) = \frac{1}{s+1}$, ölü zamanlı

sistem için ise üç parametrelili model yapısı için sistem transfer fonksiyonu $G(s) = \frac{1.08}{1.2s+1} e^{-s}$ ve dört parametrelili model yapısı için sistem transfer fonksiyonu

$G(s) = \frac{1.08}{(1+0.2897s)(1+0.7s)} e^{-s}$ olarak alınmıştır. Şekil 4.6' dan da görüldüğü

üzere bulanık PID tipi kontrolöre hata (e) ve hatanın değişimi ($\dot{e}$) giriş olarak gelir ve kontrolör çıkışından "U" işareti elde edilir.



Şekil 4.13 Bulanık kontrol için üyelik fonksiyonları

Üyelik fonksiyonları Şekil 4.13' de gösterildiği gibi hata ve hatanın değişimi için negatif anlamda "N" ve pozitif anlamda "P", kontrolör çıkış işareti için ise 0, 1, -1 olmak üzere tekil değer şeklinde seçilmiştir. Kontrolör tasarımında minimum sayıda bulanık kurala sahip kural tabanı oluşturularak kontrol işareti elde edilmesi amaçlandığı için her bir giriş değişkenine iki adet dilsel ifade tanımlanmış ve böylelikle toplam dört adet bulanık kural yazılmıştır. Bu bulanık kurallar Tablo 4.3' de gösterilmektedir.

Tablo 4.3 Bulanık kontrol kural tablosu

$e/\dot{e}$	N	P
N	-1	0
P	0	1

Tablo biçiminde verilen bulanık kurallar ayrıca aşağıdaki şekildeki gibi dilsel ifadeler biçiminde de oluşturulabilir.

1. Eğer hata (e) N ve hatanın değişimi ($\dot{e}$) N ise o zaman kontrol işaretinde yapılacak değişim değeri (v) "-1"dir.
2. Eğer hata (e) N ve hatanın değişimi ($\dot{e}$) P ise o zaman kontrol işaretinde yapılacak değişim değeri (v) "0"dır.
3. Eğer hata (e) P ve hatanın değişimi ($\dot{e}$) N ise o zaman kontrol işaretinde yapılacak değişim değeri (v) "0" dır.
4. Eğer hata (e) P ve hatanın değişimi ($\dot{e}$) P ise o zaman kontrol işaretinde yapılacak değişim değeri (v) "1" dir.

Burada verilen pozitif (P) ve negatif (N) anlamındaki fonksiyonlar hata değeri için

$$\mu_p = \frac{e+1}{2} \quad (4.17a)$$

$$\mu_N = \frac{1-e}{2} \quad (4.17b)$$

hatanın değişimi için ise

$$\mu_p = \frac{e+1}{2} \quad (4.18a)$$

$$\mu_N = \frac{1-e}{2} \quad (4.18b)$$

şeklinde alınmıştır. Kuralların elde edilmesinin ardından her bir kurala ait ateşleme şiddeti çarpım kuralı kullanılarak

$$\lambda_j = \mu_i^j * \mu_{i+1}^j \quad (i = 1, j = 1,2,3,4) \quad (4.19)$$

biçiminde hesaplanır. Burada belirtilen " μ ", bağımsız olarak her kuralda hata ve hatanın değişimi için elde edilen üyelik fonksiyon değerleridir. Daha sonra her bir kural için elde edilen ateşleme şiddetleri o kurala ait " v " değeri ile çarpılır, hesaplanan değerlerin toplamı ateşleme şiddeti toplamına bölünerek bulanık PID tipi kontrolörün çıkışına ait " U " işareti elde edilir. " U " işaretinin hesaplanmasında kullanılan algoritma

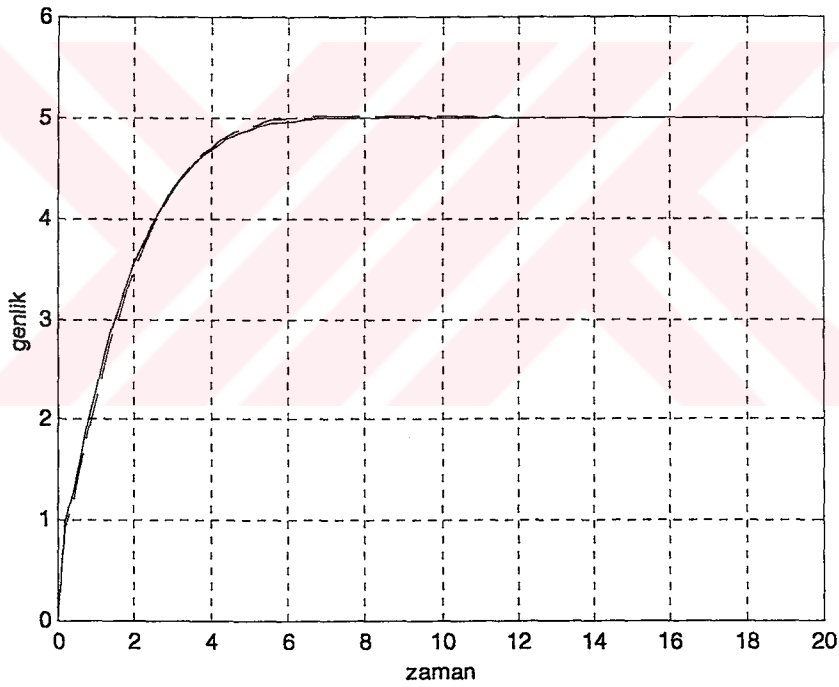
$$U = \frac{\lambda_1 v_1 + \lambda_2 v_2 + \lambda_3 v_3 + \lambda_4 v_4}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4} \quad (4.20)$$

şeklindedir.

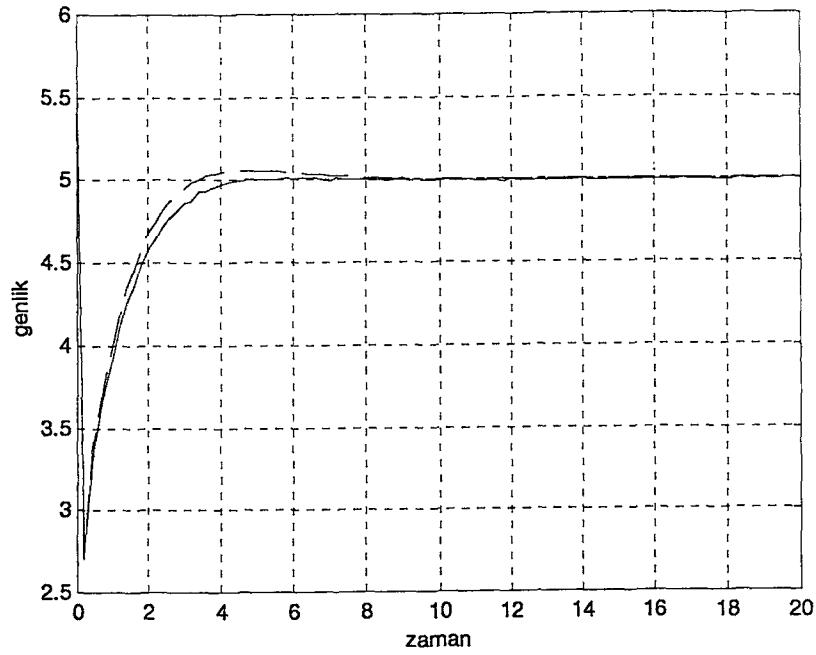
PLC programında, hesaplanan hata ve hatanın değişimi, ilk olarak normalizasyon yapılması amacıyla 10V'a karşılık düşen 32000 kelime değerine bölünmüş ve daha sonra sırasıyla K_e ve K_{de} giriş ölçekleme faktörleri ile çarpılarak kontrolör girişleri elde edilmiştir. Daha sonra kontrolör çıkışından elde edilen " U " işareti 32000 kelime değeri ve ardından sırasıyla β ve α çıkış ölçekleme faktörleri ile çarpılmıştır. Bu değerler toplanarak kontrol edilen sisteme uygulanacak olan kontrol işareti elde edilmiştir. Özayarlamaşız bulanık PID tipi kontrolöre ait PLC programı Ek C1' de verilmiştir.

4.2.1 Üç Parametrelili Ölü Zamansız Modele Uygun Olarak Tasarlanan Bulanık PID Tipi Kontrolörün Uygulanması

Ölü zamansız sistem için tasarlanan bulanık PID tipi kontrolöründe simülasyon ve PLC programında örnekleme zamanı 0.2 saniye olarak alınmış ve programlar 20 saniye boyunca çalıştırılmıştır. Simülasyonda referans olarak 5 genlikli işaret, PLC programında ise 5 genlik değerine karşılık gelen 16000 kelime değeri uygulanmıştır. Gerek simülasyonda gerekse PLC programında tasarlanan kontrolöre ilişkin ölçekleme faktörleri sistem yanıtının büyük aşım yapmaması ve kısa sürede yerleşmesi için $K_e = 0.5$, $K_{de} = 1$, $K_i = 0.5$, $K_d = 1$ olarak alınmıştır. Tasarlanan özayarlamasız bulanık PID tipi kontrolöre ilişkin gerçek sistem ve simülasyon cevapları Şekil 4.14’ de, bu kontrolöre ait kontrol işaretleri Şekil 4.15’ de gösterilmiştir.



Şekil 4.14 Tasarlanan bulanık PID tipi kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)

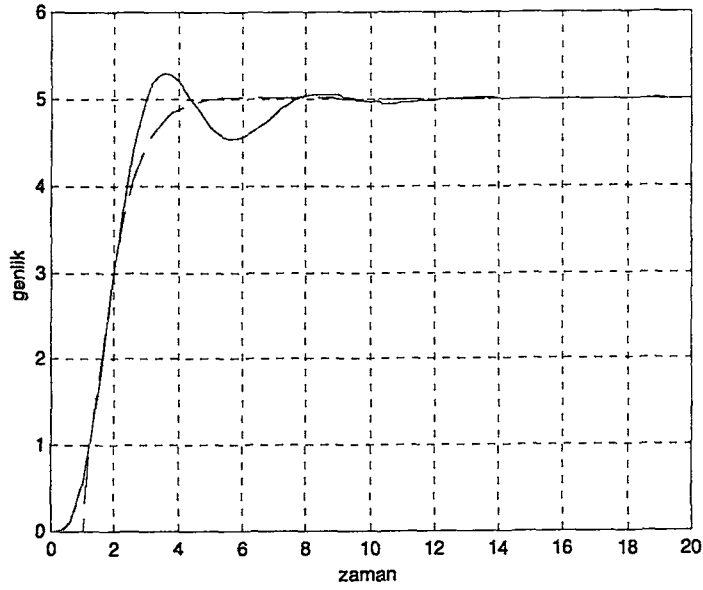


Şekil 4.15 Tasarlanan bulanık PID tipi kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

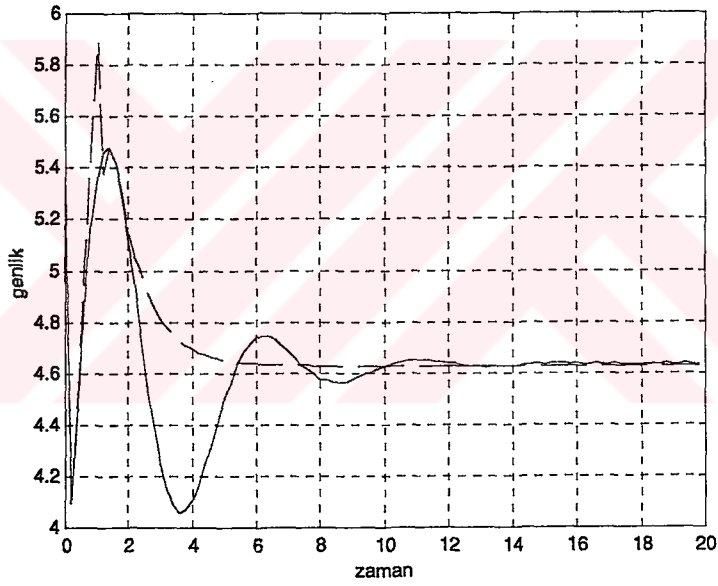
Şekil 4.14' den görüldüğü üzere, ölü zamansız sistem için tasarlanan bulanık PID tipi kontrolörü için gerçek sistem ve elde edilen üç parametrelili model cevapları birbirine yakın elde edilmiştir.

4.2.2 Üç Parametrelili Ölü Zamanlı Modele Uygun Olarak Tasarlanan Bulanık PID Tipi Kontrolörün Uygulanması

Ölü zamanlı sistem için tasarlanan bulanık PID tipi kontrolörün hem gerçek sistemde hem de simülasyonda sistem yanıtının büyük aşım yapmaması ve kısa sürede yerleşmesi için kontrolöre ait ölçekleme faktörleri $K_e = 1.5$, $K_{de} = 0.9$, $K_i = 0.12$, $K_d = 0.78$ şeklinde hesaplanmıştır. Gerçek sistemde referans değeri olarak 5 genlikli işarete denk olan 16000 kelime değeri kullanılmış, simülasyonda ise referans değeri 5 genlikli işaret uygulanmıştır. Her iki programda 0.2 saniye örnekleme değeri ile 20 saniye boyunca çalıştırılmıştır. Şekil 4.16' da, tasarlanan bulanık PID tipi kontrolörün üç parametrelili ölü zamanlı sistem modeline ve gerçek sisteme uygulanması sonucu elde edilen sistem cevabı ve simülasyon cevabı görülmektedir. Şekil 4.17' de bu kontrolöre ait kontrol işaretleri görülmektedir.



Şekil 4.16 Tasarlanan bulanık PID tipi kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



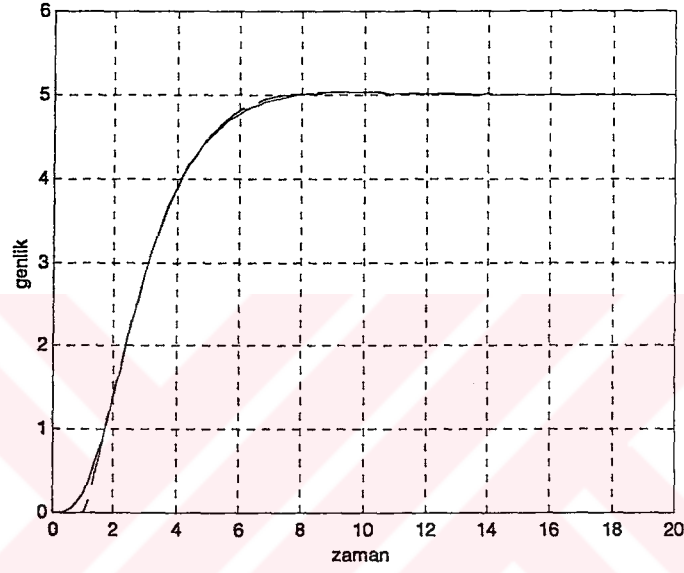
Şekil 4.17 Tasarlanan bulanık PID tipi kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Şekil 4.16' dan da görüldüğü üzere tasarlanan bulanık PID tipi kontrolöre ilişkin gerçek sistem cevabı ile simülasyon cevabı arasında fark meydana gelmiştir.

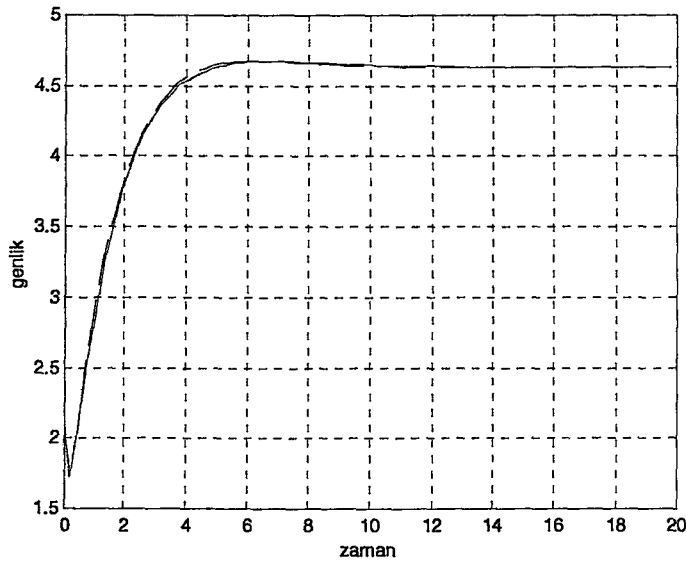
4.2.3 Dört Parametrelili Modele Uygun Olarak Tasarlanan Bulanık PID Tipi Kontrolörün Uygulanması

Dört parametrelili model ve gerçek sistem için tasarlanan bulanık PID tipi kontrolörün hem gerçek sistemde hem de simülasyonda sistem yanıtının büyük aşım yapmaması

ve kısa sürede yerleşmesi için kontrolöre ait ölçekleme faktörleri $K_e = 0.4$, $K_{de} = 0.3$, $K_i = 0.3$, $K_d = 0.9$ şeklinde hesaplanmıştır. Gerçek sistemde referans değeri olarak 5 genlikli işarete denk olan 16000 kelime değeri kullanılmış, simülasyonda ise referans değeri 5 genlikli işaret uygulanmıştır. Her iki programda 0.2 saniye örnekleme değeri ile 20 saniye boyunca çalıştırılmıştır. Şekil 4.18’ de, tasarlanan bulanık PID tipi kontrolörün dört parametrelili sistem modeline ve gerçek sisteme uygulanması sonucu elde edilen sistem cevabı ve simülasyon cevabı görülmektedir. Şekil 4.19’ da bu kontrolöre ait kontrol işaretleri görülmektedir.



Şekil 4.18 Tasarlanan bulanık PID tipi kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



Şekil 4.19 Tasarlanan bulanık PID tipi kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Şekil 4.18' den de görüldüğü üzere tasarlanan bulanık PID tipi kontrolöre ilişkin gerçek sistem cevabı ile simülasyon cevabı arasında fark yoktur.

4.2.4 Üç Ve Dört Parametrelili Sistem Modelleri İçin Tasarlanan Bulanık PID Tipi Kontrolör Sonuçlarının İrdelenmesi

Elde edilen üç ve dört parametrelili sistem modellerine ait özayarlamasız bulanık PID tipi kontrolöre ilişkin Matlab simülasyon yanıtları ile gerçek sistem yanıtları arasındaki farklar, PID kontrolörü tasarımında elde edilen sonuçlar arasındaki farklılara oldukça benzemektedir.

Ölü zamansız sistem için, ölü zaman elemanının model yapısını elde etmede kullanılmamasından ötürü, gerçek sistem modeline yakın bir model bulunmuş ve Şekil 4.14' de verilen birbirine yakın gerçek sistem ve Matlab simülasyon yanıtları elde edilmiştir.

Üç parametrelili ölü zamanlı sistem modeli, ölü zaman elemanının göstermiş olduğu doğrusal olmayan özellikten dolayı yetersiz kalmakta ve gerçek sistem modelinden daha farklı bir model yapısını temsil etmektedir. Bu nedenle Şekil 4.16' da görüleceği gibi, tasarlanan bulanık PID tipi kontrolöre ilişkin Matlab simülasyon yanıtı gerçek sistem yanıtından farklı elde edilmektedir.

Ölü zamanlı sistem için üç parametrelili model yapısının yetersiz kalması, kontrolörün modelin bir derece artırılarak dört parametre ile oluşturulan ikinci dereceden ölü zamanlı sistem modeline uygulanmasını zorunlu kılmıştır. Sistem model mertebesinin ve parametre sayısının artması ile üç parametrelili modelde bulunan karesel hatalar ve ölü zaman elemanının göstermiş olduğu doğrusal olmayan özellik azaltılarak Şekil 4.18' den görüldüğü üzere tasarlanan bulanık PID tipi kontrolöre ilişkin Matlab simülasyon yanıtının gerçek sistem cevabına yakın olması sağlanmıştır.

Ayrıca bulanık PID tipi kontrolöre ait sistem yanıtları PID kontrolörüne ait sistem yanıtları ile karşılaştırıldığında, bulanık kontrolöre ait ölçekleme faktörlerinin düzgün olarak ayarlanması neticesinde bu kontrolörün PID kontrolöründen daha iyi sonuçlar verdiği elde edilen yanıtlardan açıkça görülmektedir.

4.3 Özayarlama Bulanık PID Tipi Kontrolör Tasarımı

Bu bölümde gerçek sistem ile elde edilen dört parametrelili modele sırasıyla Bölüm 4.1.2' de anlatılan aşım gözleyicisi, fonksiyon ayarlayıcısı ve bağıl hız gözleyicisi yoluyla parametreleri uyarlanan özayarlama bulanık PID tipi kontrolör tasarlanmıştır. Kontrolör tasarımında Bölüm 4.2' de verilen kural tabanı ve üyelik fonksiyonları kullanılmıştır.

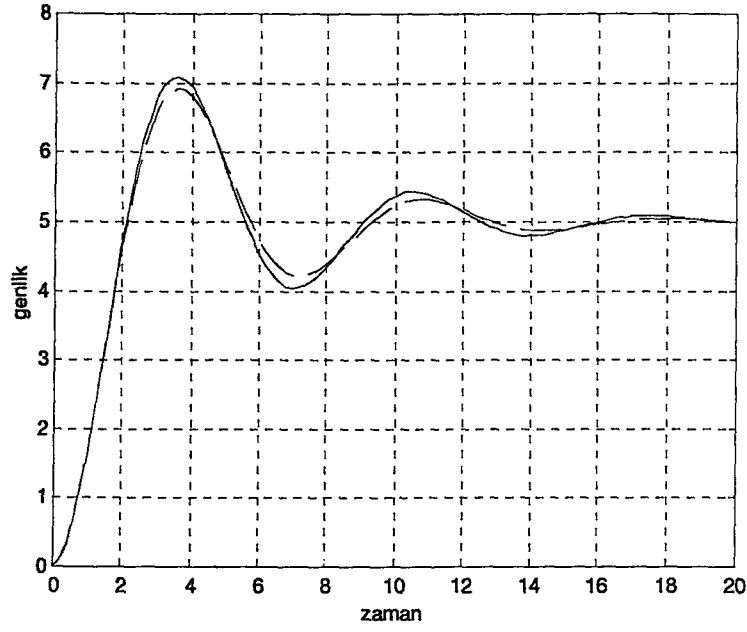
Bir önceki bölümde gerçek sistem ile elde edilen üç ve dört parametrelili modellere özayarlama bulanık PID tipi kontrolör tasarlanmış ve tasarlanan kontrolör için dört parametrelili modelin gerçek sistem yanıtına çok daha yakın bir yanıtı sahip olduğu görülmüştür. Bu nedenden ötürü bu bölümde tasarlanacak olan özayarlama bulanık PID tipi kontrolör için Matlab simülasyonunda sadece dört parametrelili model yapısı alınmıştır.

Daha önceki bölümde elde edilen sistemlerin modellenmesinde ölü zaman elemanın sisteme doğrusal olmayan özellik eklediği ve bu özelliğin sistem yanıtlarını önemli ölçüde etkilediği gösterilmiştir. Bu nedenden ötürü bu bölümde tasarlanacak kontrolöre ait üç ayrı ayarlama yöntemi için de kontrol algoritmaları, ilk olarak ölü zamansız sistemlere uygulanarak elde edilen gerçek sistem ve Matlab simülasyon yanıtları karşılaştırma yoluyla kontrol edilmiş, daha sonra ölçekleme faktörleri düzgün olarak ayarlanamayan kontrolörün tasarlandığı ölü zamanlı sisteme uygulanmıştır. Kontrolör tasarımında sırasıyla kullanılan yöntemlere ait PLC programları Ek C2, Ek C3, Ek C4 ve Ek C5' de verilmiştir.

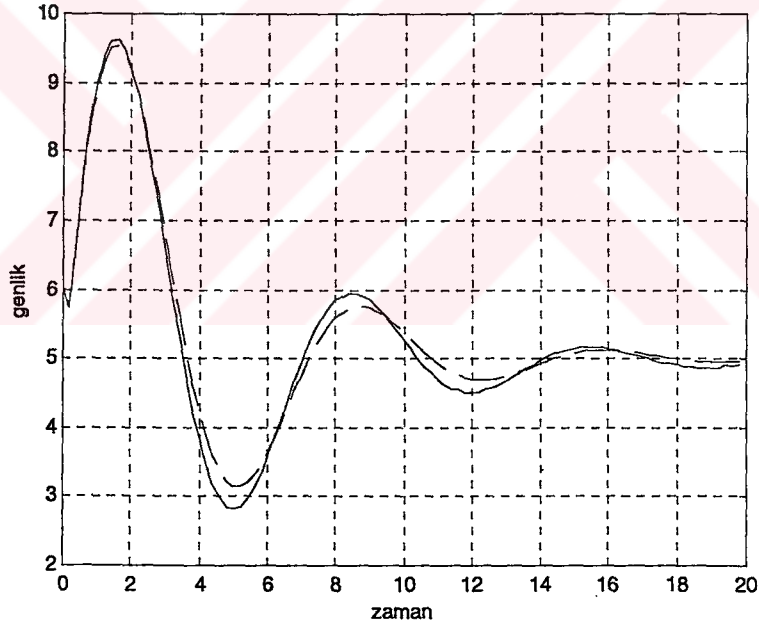
4.3.1 Aşım Gözleyicisi İle Parametre Uyarlamalı Yöntem

4.3.1.1 Aşım Gözleyicisi Yönteminin Ölü Zamansız Sisteme Uygulanması

İlk olarak, transfer fonksiyonu $G(s) = \frac{1}{(s+1)^2}$ olan ölü zamansız ikinci dereceden doğrusal sisteme aşım yapması amacıyla ölçekleme faktörleri $K_e = 0.9$, $K_{de} = 0.6$, $K_i = 0.6$, $K_d = 1$ şeklinde seçilen özayarlama bulanık PID tipi kontrolör tasarlanmıştır. Bu kontrolöre ait sistem yanıtları Şekil 4.20' de ve kontrol işaretleri Şekil 4.21 ' de görülmektedir.

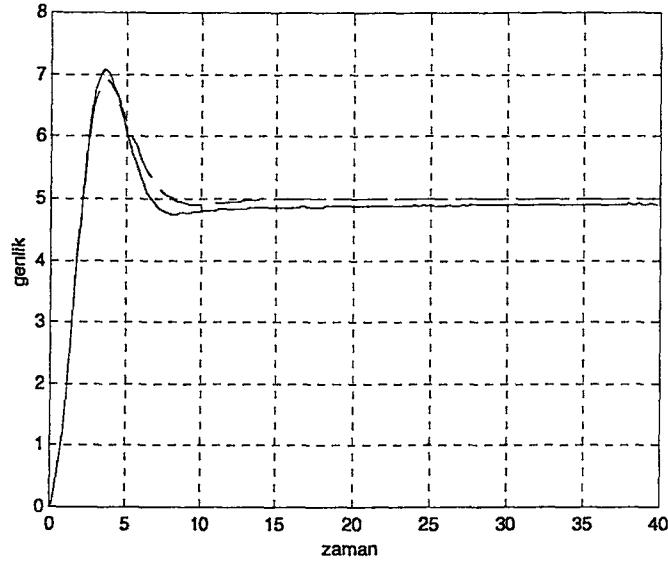


Şekil 4.20 Ölü zamansız ikinci dereceden doğrusal sisteme tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)

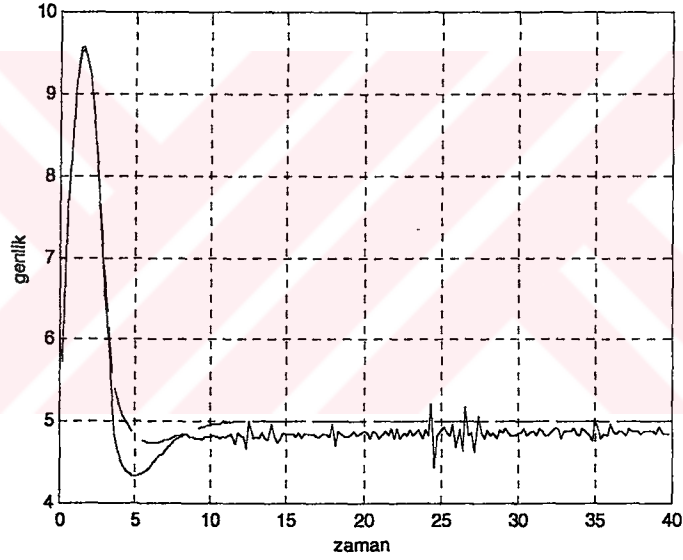


Şekil 4.21 Ölü zamansız ikinci dereceden doğrusal sisteme tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait kontrol işaretleri (düz: gerçek sistem yanıtına ait kontrol işareti, kesikli: simülasyon yanıtına ait kontrol işareti)

Daha sonra tasarlanan bu kontrolöre aşım gözleyicisi ile parametre uyarlamalı yöntem uygulanmıştır. Sistem yanıtlarının daha iyi şekilde incelenmesi maksadıyla programlar 40 sn boyunca çalıştırılmıştır. Tasarlanan bu kontrolöre ait sistem yanıtları ise Şekil 4.22' de, kontrol işaretleri Şekil 4.23' de görülmektedir.



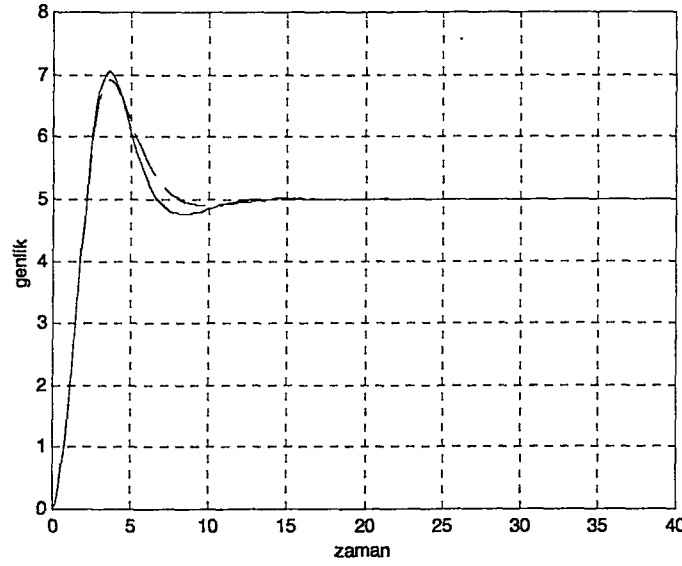
Şekil 4.22 Eşik değerine sahip olmayan kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



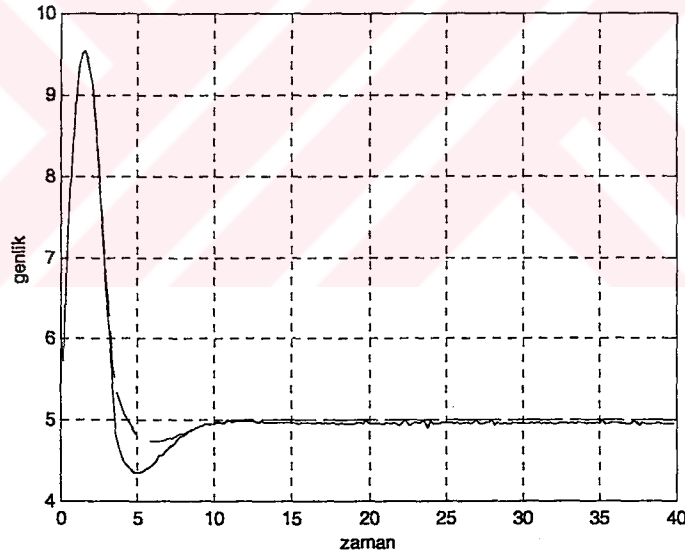
Şekil 4.23 Eşik değerine sahip olmayan kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Sistem yanıtında meydana gelen tepe mutlak değerlerinin zamanla çok küçük değerler alması ve bu değerlerin ayarlama algoritmasında giriş ölçekleme faktörünü bölme işlemiyle çok fazla artırması, buna karşın çıkış ölçekleme faktörünü de çarpma işlemiyle çok fazla azaltması neticesinde, Şekil 4.22' den de görüldüğü üzere elde edilen gerçek sistem yanıtında sürekli hal hatası meydana gelmektedir. Bu nedenden ötürü aynı sisteme tepe değeri için 0.25V eşik değerine sahip yeni bir aşım gözleyicisi yoluyla parametreleri uyarlanan bulanık kontrolör tasarlanmıştır. Bu

kontrolöre ait sistem yanıtları Şekil 4.24’ de ve kontrol işaretleri Şekil 4.25’ de görülmektedir.



Şekil 4.24 Eşik değerine sahip olan kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)

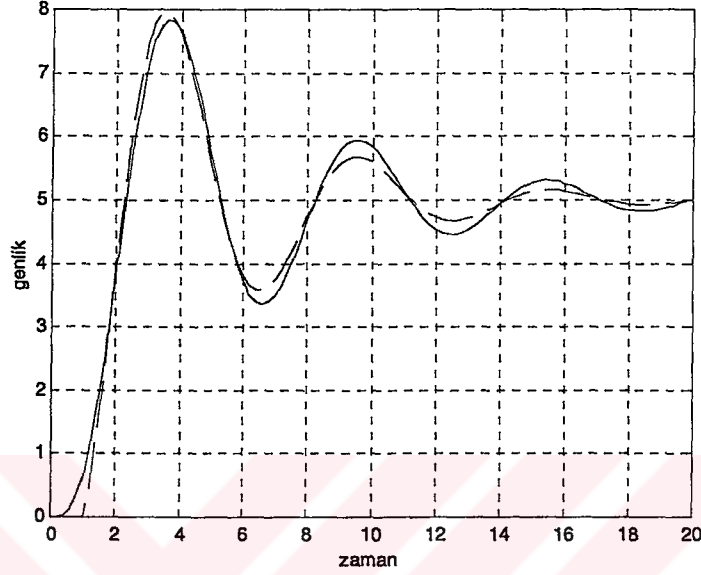


Şekil 4.25 Eşik değerine sahip olan kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

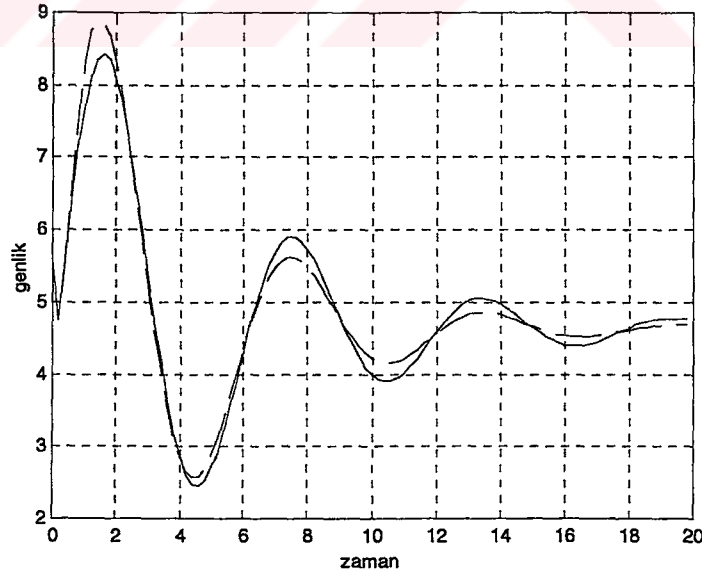
Şekil 4.24’ den de görüldüğü gibi tasarlanan kontrolöre eşik değerinin ilave edilmesi neticesinde elde edilen gerçek sistem yanıtındaki sürekli hal hatası ortadan kalkmakta ve yanıt belirli bir süre sonra referans değerini takip etmektedir. Şekil 4.22 ve Şekil 4.24’ den görüldüğü gibi aşım gözleyicisi yoluyla parametreleri uyarlanan bulanık PID tipi kontrolöre ait gerçek sistem ve simülasyon yanıtları birbirine yakın elde edilmiştir.

4.3.1.2 Aşım Gözleyicisi Yönteminin Ölü Zamanlı Sisteme Uygulanması

Şekil 4.26' de ise dört parametrelili model ve ölü zamanlı gerçek sisteme aşım yapması amacıyla ölçekleme faktörleri $K_e = 0.9$, $K_{de} = 0.7$, $K_i = 0.4$, $K_d = 1$ şeklinde seçilerek tasarlanan özayarlamasız bulanık PID tipi kontrolör yanıtları ve Şekil 4.27' de kontrol işaretleri görülmektedir.



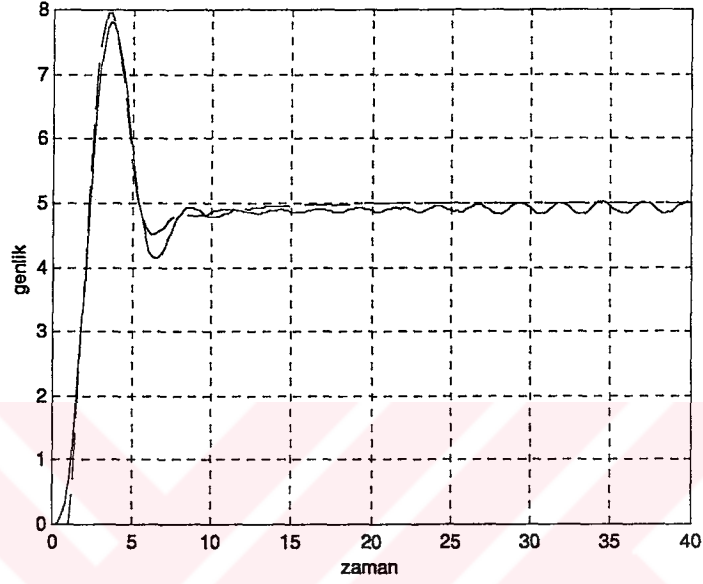
Şekil 4.26 Dört parametrelili model ve gerçek sistem için tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



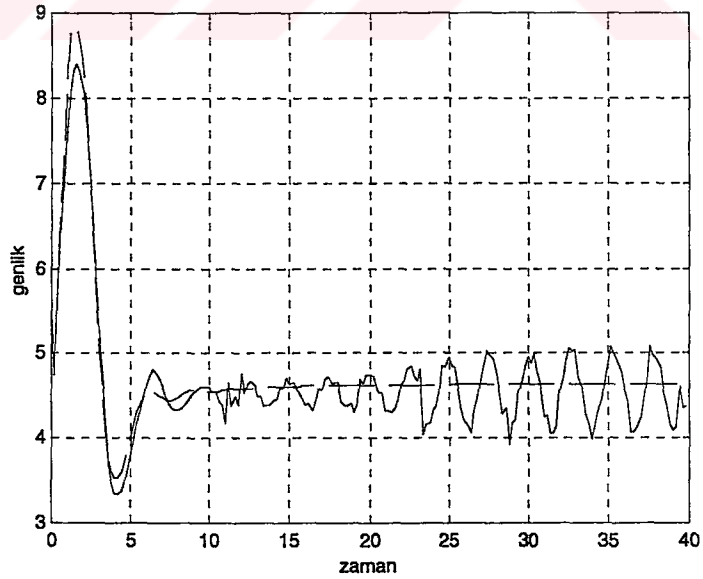
Şekil 4.27 Dört parametrelili model ve gerçek sistem için tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Kontrolör tasarımı referans olarak 5 genlikli işaret değeri alınmış ve simülasyon ile PLC programı 20 sn süre ile çalıştırılmıştır.

Tasarlanan bu kontrolöre ilk olarak eşik değerine sahip olmayan aşım gözleyicisi yoluyla parametre uyarlamalı yöntem uygulanmıştır. Programların çalıştırılma süreleri 40 sn olması durumunda kontrolöre ait sistem yanıtları Şekil 4.28’ de, kontrol işaretleri ise Şekil 4.29’ da görülmektedir.

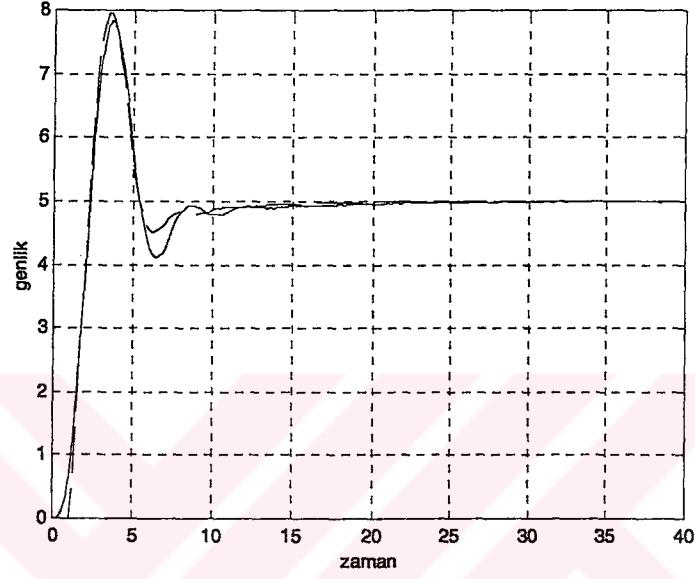


Şekil 4.28 Eşik değerine sahip olmayan kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)

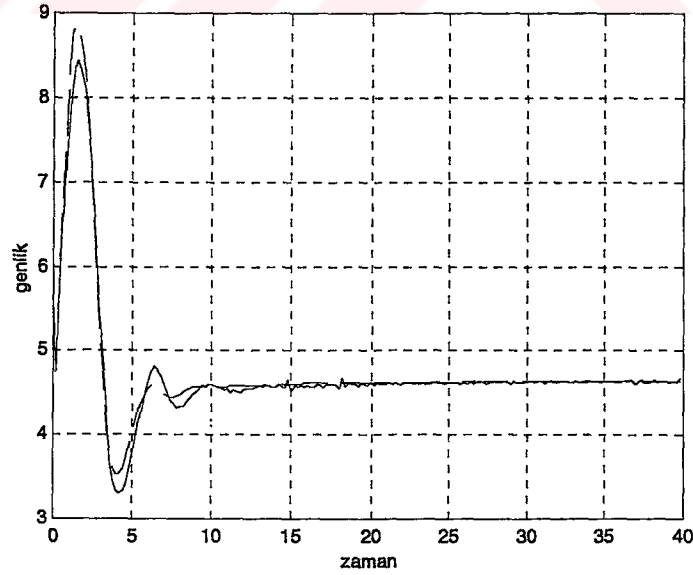


Şekil 4.29 Eşik değerine sahip olmayan kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Şekil 4.28’ den de görüldüğü üzere ayarlama algoritmasında hesaplanan giriş ve çıkış ölçekleme faktörlerinin tepe mutlak değerlerine bağlı olarak sırasıyla çok büyük ve küçük değerlere ulaşması sebebiyle elde edilen gerçek sistem yanıtında sürekli hal hatası meydana gelmekte ve belirli bir süre sonra sistem yanıtında salınım ortaya çıkmaktadır. Bu olumsuzlukları ortadan kaldırmak amacıyla aynı sisteme bu kez yine tepe değeri için 0.25V eşik değerine sahip bir kontrolör tasarlanmıştır. Bu kontrolöre ait sistem yanıtları Şekil 4.30’ da, kontrol işaretleri Şekil 4.31’ de görülmektedir.



Şekil 4.30 Eşik değerine sahip olan kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



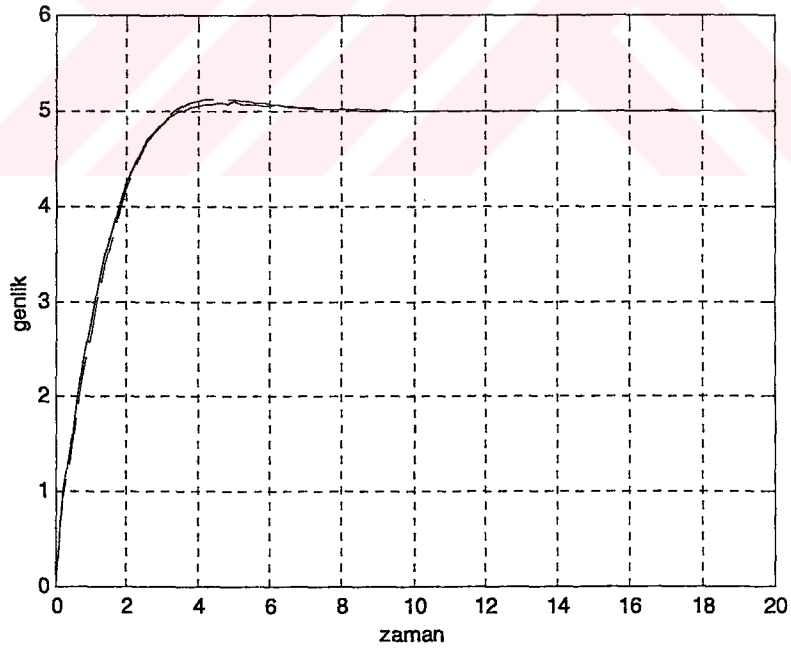
Şekil 4.31 Eşik değerine sahip olan kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Şekil 4.30' dan da görüldüğü üzere tasarlanan kontrolöre eşik değerinin eklenmesi ile eşik değerine sahip olmayan kontrolöre nazaran sistem yanıtında meydana gelen sürekli hal hatası ve salınımlar ortadan kaldırılmaktadır. Ayrıca Şekil 4.28 ve Şekil 4.30' dan görüldüğü gibi aşım gözleyicisi yoluyla parametreleri uyarlanan bulanık PID tipi kontrolöre ait gerçek sistem ve simülasyon yanıtları birbirine yakın elde edilmiştir.

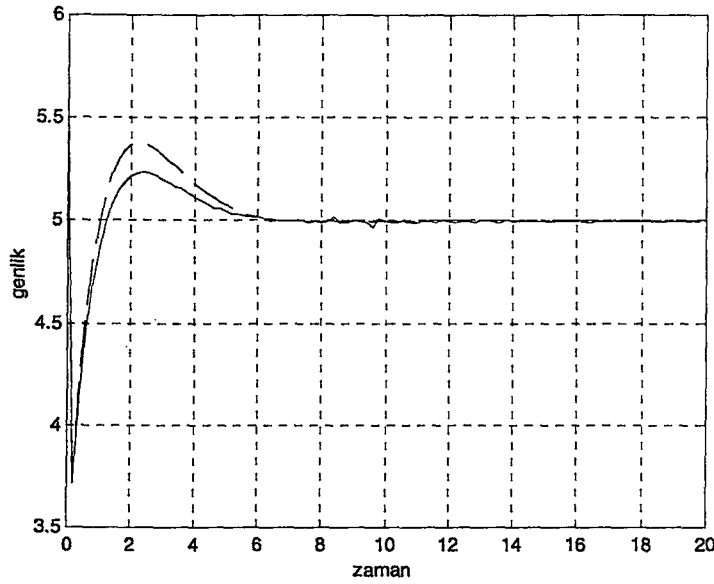
4.3.2 Fonksiyon Ayarlayıcısı İle Parametre Uyarlamalı Yöntem

4.3.2.1 Fonksiyon Ayarlayıcısı Yönteminin Ölü Zamansız Sisteme Uygulanması

İlk aşamada, transfer fonksiyonu $G(s) = \frac{1}{(s+1)}$ olan ölü zamansız birinci dereceden doğrusal sisteme ölçekleme faktörleri $K_e = 0.9$, $K_{de} = 0.7$, $K_i = 0.4$, $K_d = 1$ şeklinde seçilen özayarlamasız bulanık PID tipi kontrolör tasarlanmıştır. Kontrolör tasarımında referans 5 genlikli işaret değeri alınmıştır ve programlar 20 sn boyunca çalıştırılmıştır. Bu kontrolöre ait sistem yanıtları Şekil 4.32' de ve kontrol işaretleri Şekil 4.33' de görülmektedir.

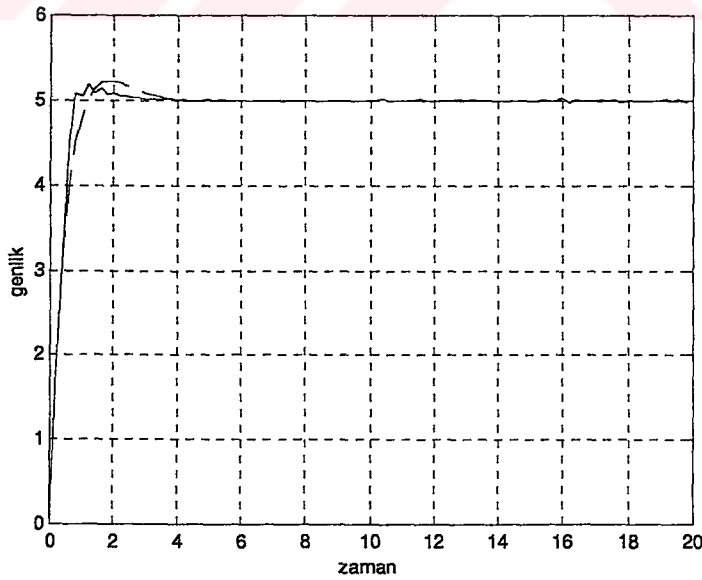


Şekil 4.32 Ölü zamansız birinci dereceden doğrusal sisteme tasarlanan özayarlamasız bulanık PID tipi kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)

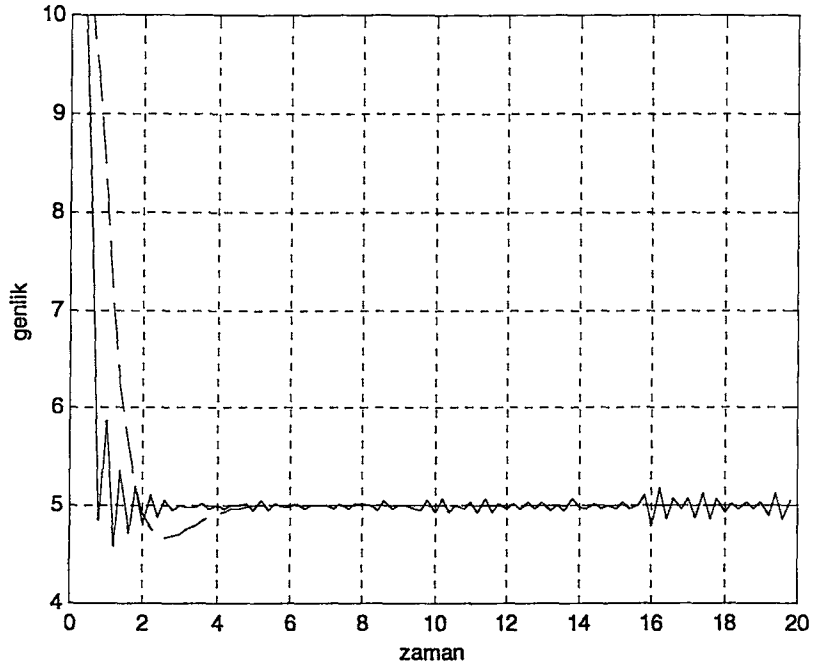


Şekil 4.33 Ölü zamansız birinci dereceden doğrusal sisteme tasarlanan bulanık PID tipi kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

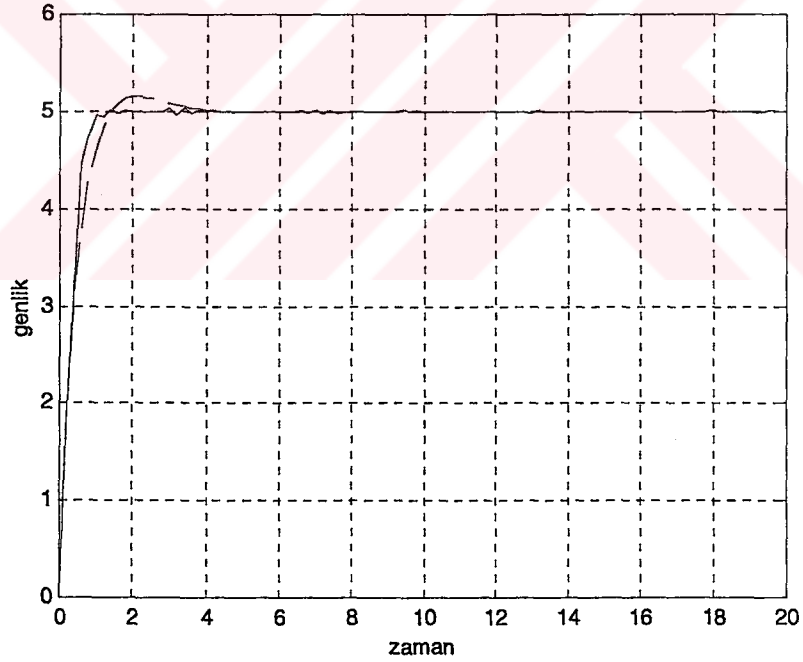
Daha sonra bu sisteme fonksiyon ayarlayıcısı yoluyla parametresi uyarlanan bir bulanık PID tipi kontrolör tasarlanmıştır. Kontrolör tasarımında $f(e(t))$ ve $g(e(t))$ fonksiyonlarına ait olan a_1, a_2, b_1 ve b_2 pozitif katsayıları $[0, 30]$ değer aralığında genetik algoritma yoluyla araştırılmıştır. Değişik katsayı değerleri için ayrı ayrı sistem yanıtları elde edileceğinden iki değişik parametre değerleri için elde edilen gerçek sistem yanıtları ve kontrol işaretleri aşağıda gösterilmiştir.



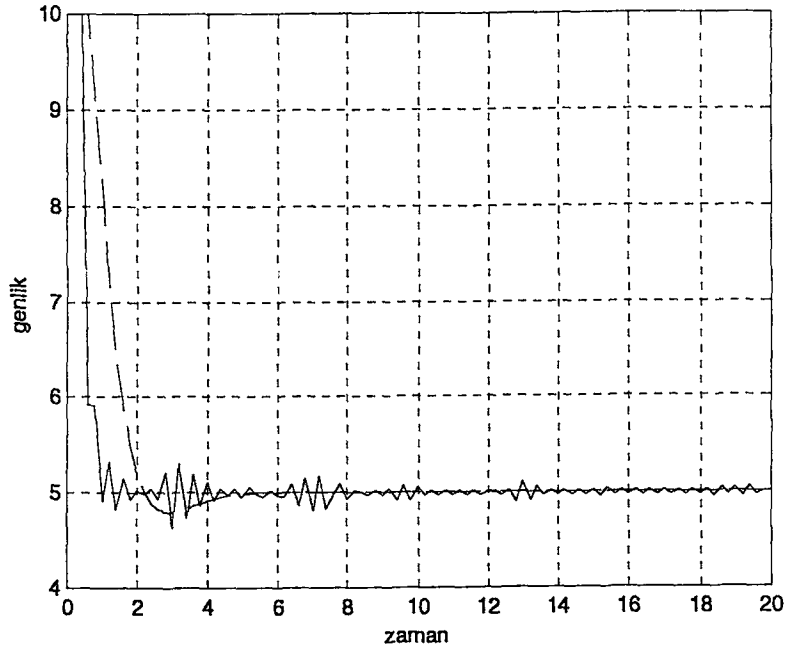
Şekil 4.34 Katsayıların $a_1 = 2$, $a_2 = 8$, $b_1 = 0.5$, $b_2 = 4.5$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



Şekil 4.35 Katsayıların $a_1 = 2$, $a_2 = 8$, $b_1 = 0.5$, $b_2 = 4.5$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)



Şekil 4.36 Katsayıların $a_1 = 2.3$, $a_2 = 7.5$, $b_1 = 0.3$, $b_2 = 5$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)

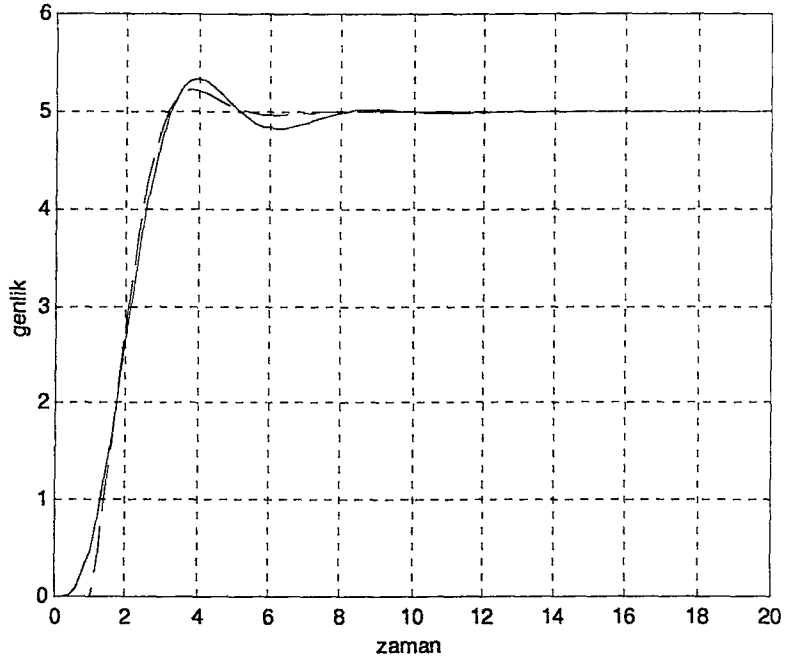


Şekil 4.37 Katsayıların $a_1 = 2.3$, $a_2 = 7.5$, $b_1 = 0.3$, $b_2 = 5$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

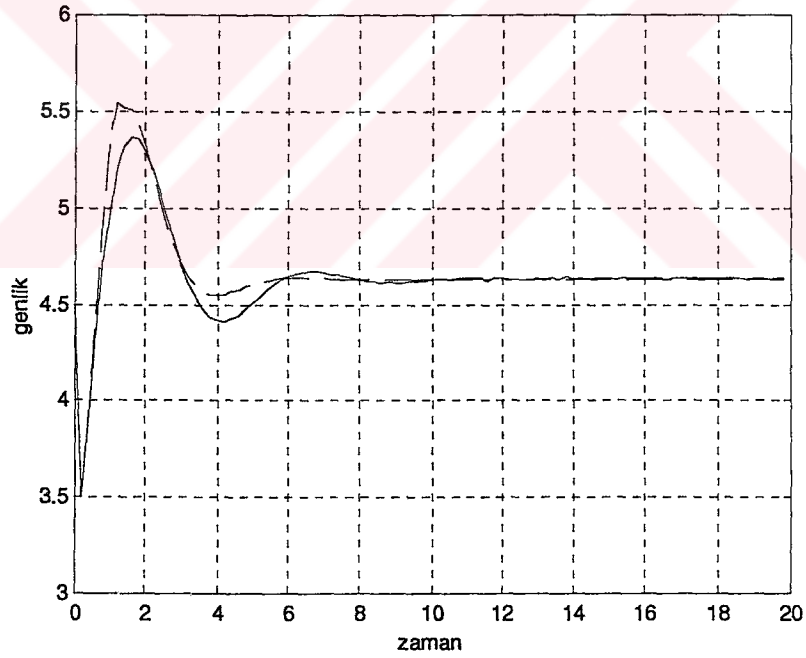
Şekil 4.34 ve Şekil 4.36' dan da görüldüğü üzere tasarlanan fonksiyon ayarlayıcısı ile parametresi uyarlanan bulanık PID tipi kontrolöre ait gerçek sistem ve Matlab simülasyon yanıtları birbirine oldukça yakın elde edilmiştir.

4.3.2.2 Fonksiyon Ayarlayıcısı Yönteminin Ölü Zamanlı Sisteme Uygulanması

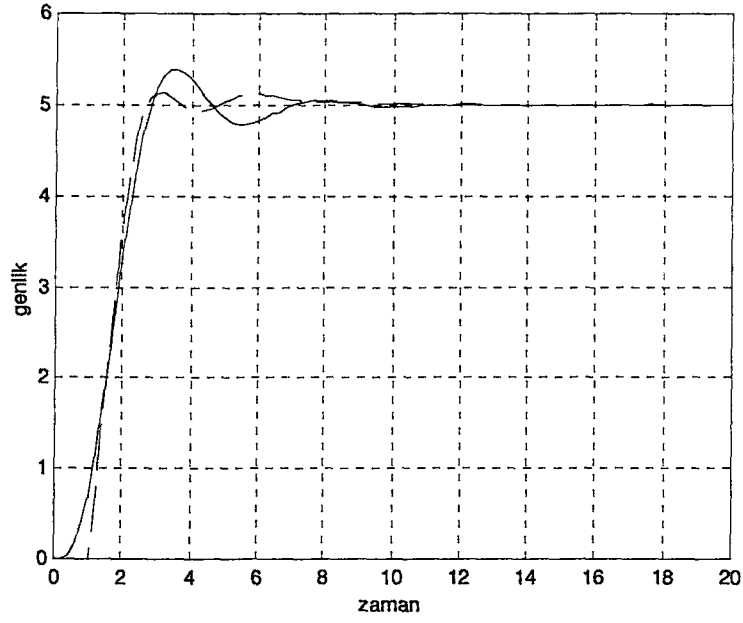
Daha önce dört parametrelili model ve ölü zamanlı gerçek sistem için ölçekleme faktörleri $K_e = 0.9$, $K_{de} = 0.7$, $K_i = 0.4$, $K_d = 1$ şeklinde seçilerek elde edilen bulanık PID tipi kontrolöre, fonksiyon ayarlayıcısı yöntemi uygulanarak elde edilen gerçek sistem ve simülasyon yanıtları Şekil 4.38' de, bu kontrolöre ait kontrol işaretleri ise Şekil 4.39' da görülmektedir. Kontrolör tasarımında fonksiyonlara ait pozitif katsayılar genetik algoritma yolu ile $a_1 = 0.25$, $a_2 = 0.4$, $b_1 = 0.05$ ve $b_2 = 0.8$ olarak hesaplanmıştır. Fonksiyonlara ait pozitif katsayıların $a_1 = 0.28$, $a_2 = 0.45$, $b_1 = 0.14$ ve $b_2 = 2.2$ olarak seçilmesi durumunda elde edilecek kontrolör için gerçek sistem ve dört parametrelili modele ait simülasyon yanıtları Şekil 4.40' da, kontrol işaretleri Şekil 41' de görülmektedir.



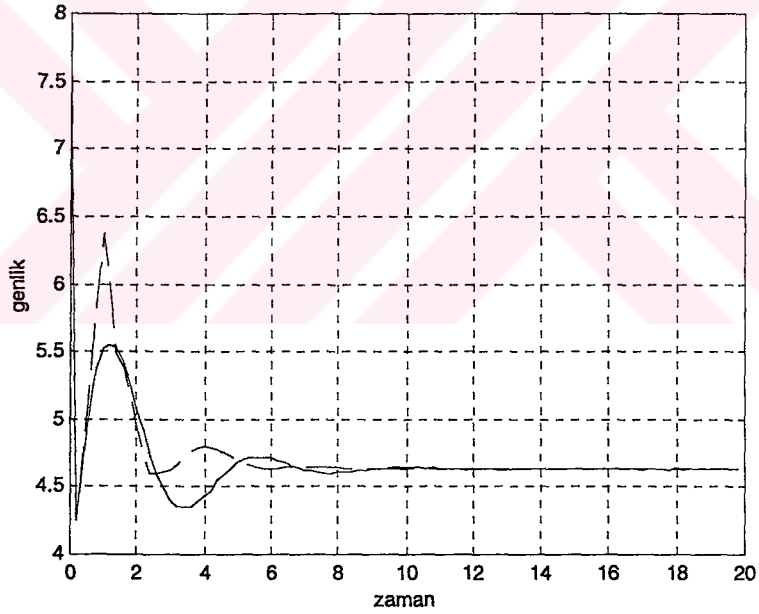
Şekil 4.38 Katsayıların $a_1 = 0.25$, $a_2 = 0.4$, $b_1 = 0.05$, $b_2 = 0.8$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



Şekil 4.39 Katsayıların $a_1 = 0.25$, $a_2 = 0.4$, $b_1 = 0.05$, $b_2 = 0.8$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri (düz: gerçekte sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)



Şekil 4.40 Katsayıların $a_1 = 0.28$, $a_2 = 0.45$, $b_1 = 0.14$ ve $b_2 = 2.2$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



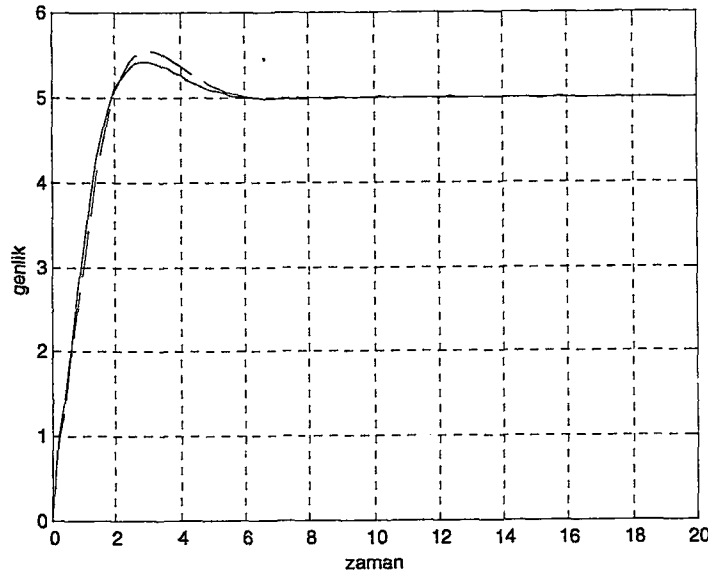
Şekil 4.41 Katsayıların $a_1 = 0.28$, $a_2 = 0.45$, $b_1 = 0.14$ ve $b_2 = 2.2$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Şekil 4.38 ve Şekil 4.40' dan görüldüğü üzere elde edilen gerçek sistem ile simülasyon yanıtları arasında fark meydana gelmektedir. Bu fark temel olarak ayarlama algoritmasında mevcut olan ve her biri hata değerine bağlı olan deneysel fonksiyonlardan meydana gelmektedir.

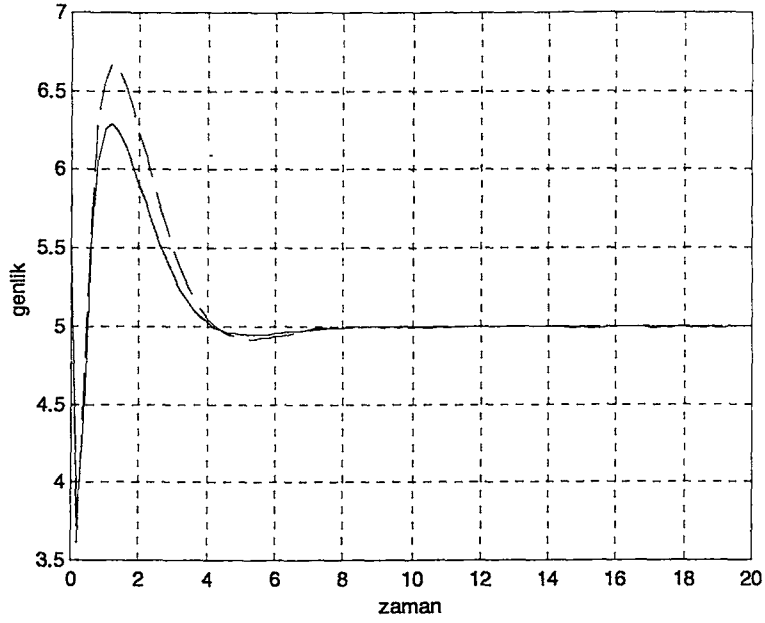
4.3.3 Bağıl Hız Gözleyicisi İle Parametre Uyarlamalı Yöntem

4.3.3.1 Bağıl Hız Gözleyicisi Yönteminin Ölü Zamansız Sisteme Uygulanması

Bağıl hız gözleyicisi ile parametresi uyarlanan bulanık PID tipi kontrolör tasarımı ilk olarak yöntem, transfer fonksiyonu $G(s) = \frac{1}{(s+1)}$ olan ölü zamansız birinci dereceden doğrusal sisteme ölçekleme faktörleri $K_e = 0.9$, $K_{de} = 0.7$, $K_i = 0.4$, $K_d = 1$ şeklinde seçilerek tasarlanan özayarlamasız bulanık PID tipi kontrolöre uygulanmıştır. Referansın 5 genlikli işaret değeri alındığı ve 20 sn boyunca çalıştırılarak elde edilen kontrolöre ait gerçek sistem ve simülasyon yanıtları ile kontrol işaretleri sırasıyla Şekil 4.32 ve Şekil 4.33’ de verilmiştir. Yöntemin uygulanması neticesinde elde edilen gerçek sistem ve Matlab simülasyon yanıtları Şekil 4.42’ de, kontrol işaretleri ise Şekil 4.43’ de gösterilmiştir. Tasarımda kontrolöre ait K_f ve K_{fd} katsayıları $[0, 30]$ değer aralığında genetik algoritma ile sırasıyla 1.2 ve 1 olarak hesaplanmıştır. Böylelikle oransal kontrol şiddeti değiştirilmemiş ve sistemin hataya karşı olan hızlı tepkisinin her zaman korunması sağlanmıştır. Ayrıca bulanık parametre düzenleyicisi çıkışı olan γ değerini belirlemede kullanılan kural tabanı Tablo 4.2’ de verildiği şekildedir. Tasarımda $|e|$, r_v ve γ için üçgen tip üyelik fonksiyonları alınmıştır.

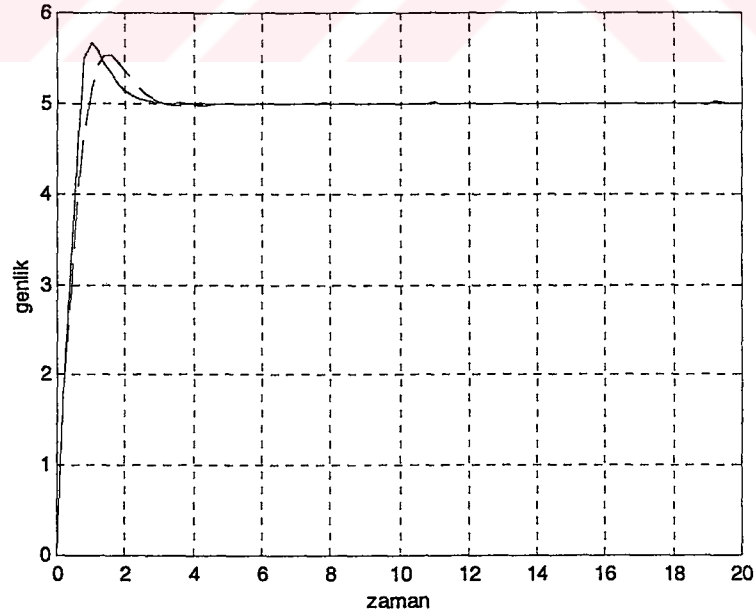


Şekil 4.42 Katsayıların $K_f = 1.2$ ve $K_{fd} = 1$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)

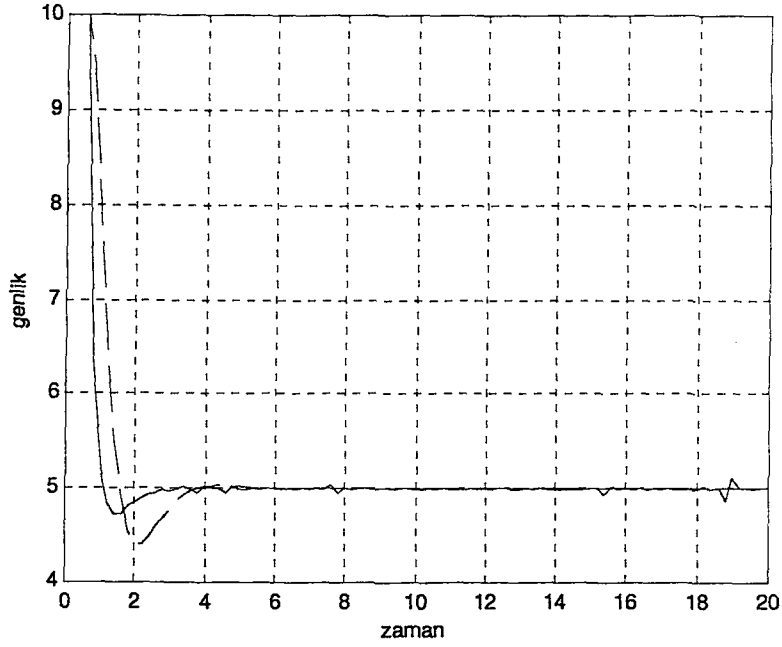


Şekil 4.43 Katsayıların $K_f = 1.2$ ve $K_{fd} = 1$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri (düz: gerçek sistem ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

Oransal kontrol şiddetinin değiştirilmesi ile hesaplanan yeni katsayılar neticesinde elde edilen kontrolöre ait sistem yanıtları ise Şekil 4.44' de, kontrol işaretleri ise Şekil 4.45' de gösterilmiştir. Bu durumda kontrolör tasarımında hesaplanan katsayı değerleri ise $K_f = 0.21$ ve $K_{fd} = 22$ şeklindedir.



Şekil 4.44 Katsayıların $K_f = 22$ ve $K_{fd} = 0.21$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)

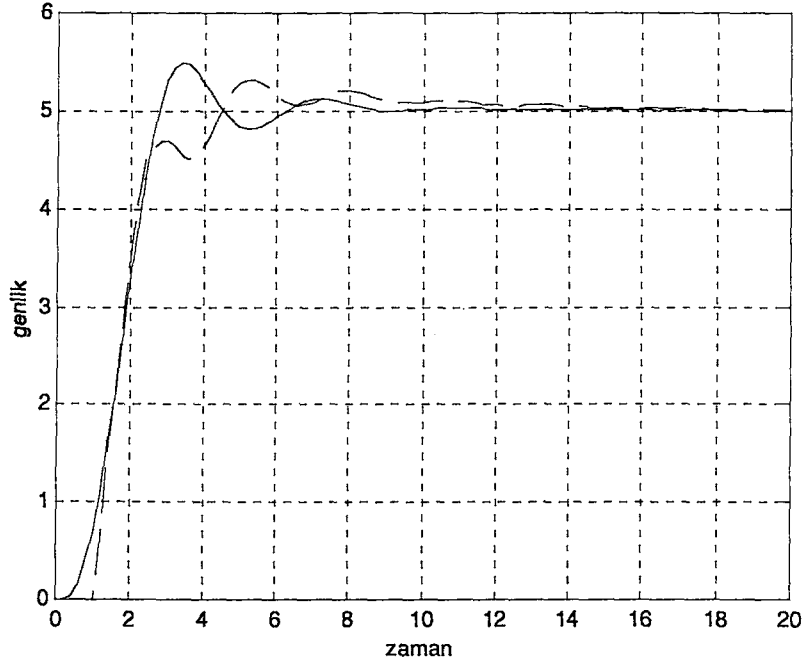


Şekil 4.45 Katsayıların $K_f = 22$ ve $K_{fd} = 0.21$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

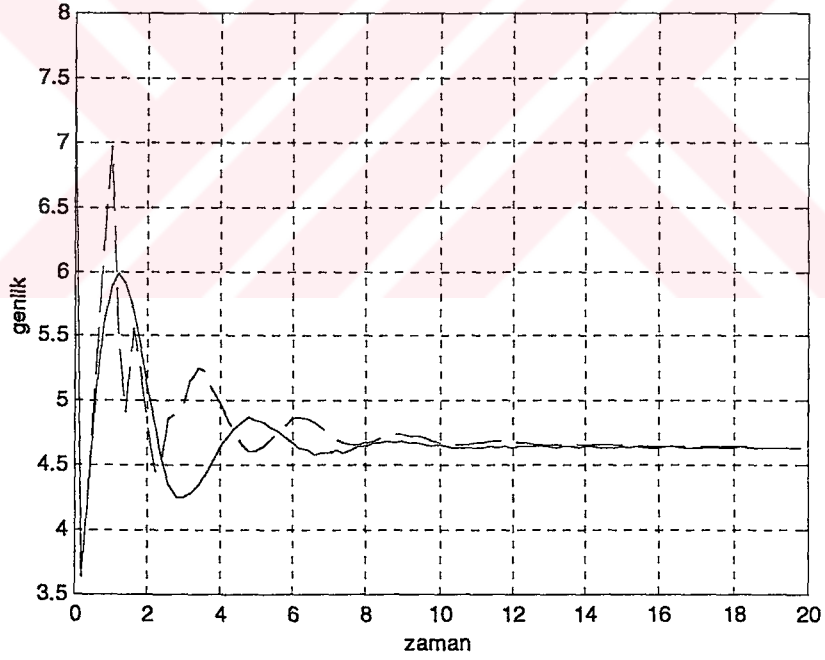
Şekil 4.42 ve Şekil 4.44' den görüldüğü üzere tasarlanan kontrolöre ait gerçek sistem ve simülasyon yanıtları arasında önemli bir fark oluşmamaktadır.

4.3.3.2 Bağlı Hız Gözleyicisi Yönteminin Ölü Zamanlı Sisteme Uygulanması

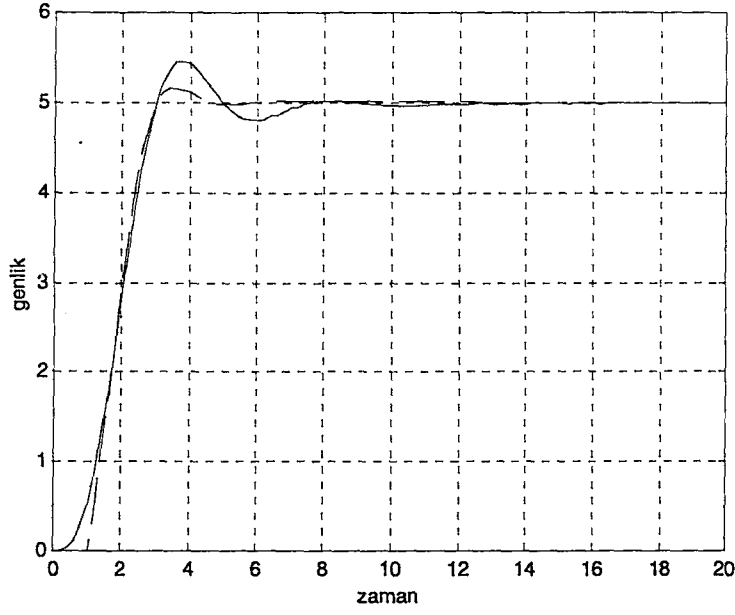
Şekil 4.46' da, daha önce dört parametrelili model ve ölü zamanlı gerçek sistem için ölçekleme faktörleri $K_e = 0.9$, $K_{de} = 0.7$, $K_i = 0.4$, $K_d = 1$ şeklinde seçilerek elde edilen bulanık PID tipi kontrolöre bağlı hız gözleyicisi yöntemi uygulanarak elde edilen gerçek sistem ve simülasyon yanıtları, Şekil 4.47' de ise bu kontrolöre ait kontrol işaretleri görülmektedir. Kontrolör tasarımında K_f ve K_{fd} katsayıları genetik algoritma yolu ile sırasıyla 6.45 ve 1 olarak hesaplanmıştır. Böylece oransal kontrol şiddeti değiştirilmemiş ve sistemin hataya karşı olan hızlı tepkisinin her zaman korunması sağlanmıştır. K_{fd} parametresini 1 değerinde sabit tutmayarak oransal kontrol şiddetinin değiştirilmesi sonucunda hesaplanan $K_f = 7.8$ ve $K_{fd} = 0.25$ katsayı değerleri için tasarlanan kontrolöre ait sistem yanıtları Şekil 4.48' de, kontrol işaretleri Şekil 4.49' da görülmektedir.



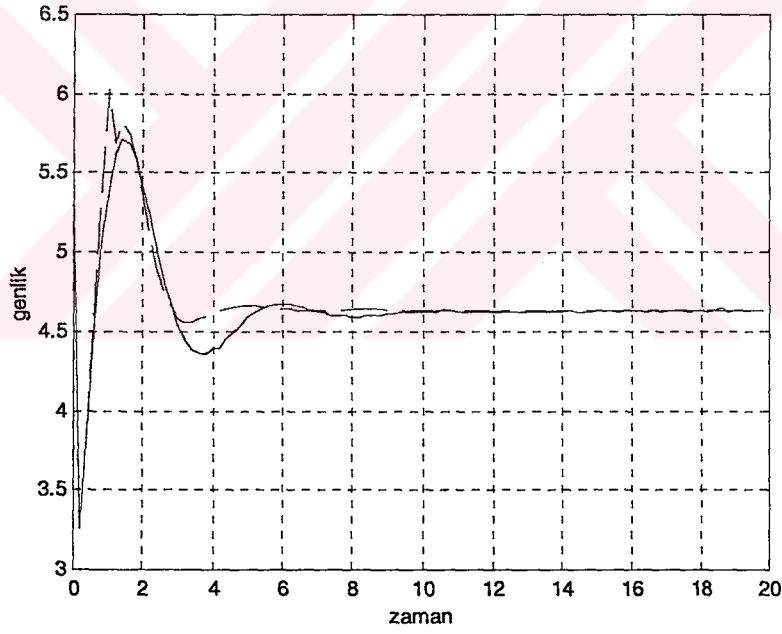
Şekil 4.46 Katsayıların $K_f = 6.45$ ve $K_{fd} = 1$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



Şekil 4.47 Katsayıların $K_f = 6.45$ ve $K_{fd} = 1$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)



Şekil 4.48 Katsayıların $K_f = 7.8$ ve $K_{fd} = 0.25$ seçilmesi durumunda elde edilen kontrolöre ait sistem yanıtları (düz: gerçek sistem yanıtı, kesikli: simülasyon yanıtı)



Şekil 4.49 Katsayıların $K_f = 7.8$ ve $K_{fd} = 0.25$ seçilmesi durumunda elde edilen kontrolöre ait kontrol işaretleri (düz: gerçek sisteme ait kontrol işareti, kesikli: simülasyona ait kontrol işareti)

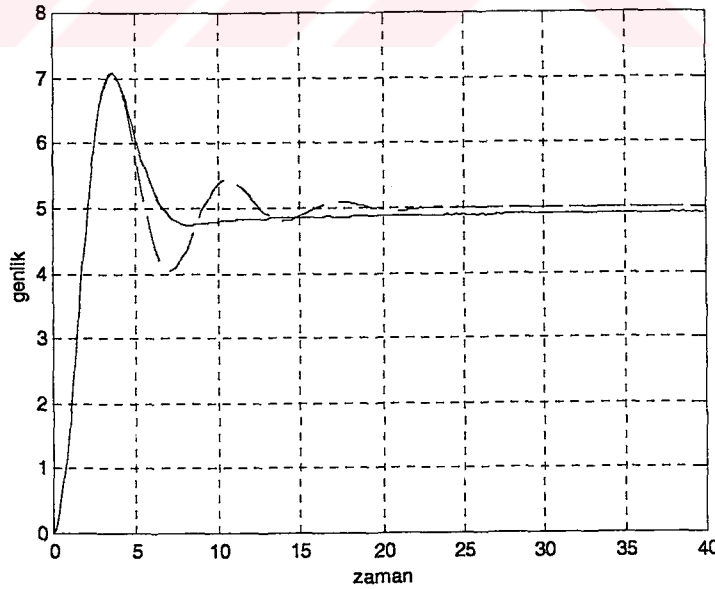
Ölü zamanlı sistem için tasarlanan kontrolöre ait gerçek sistem ve simülasyon yanıtlarında ise önemli bir farkın olduğu görülmektedir. Bu fark ise ayarlama algoritmasının giriş-çıkış ölçekleme faktörlerini sistem yanıtının hızına ve hata değerine bağlı olarak uyarlaması neticesinde oluşmaktadır.

4.3.4 Dört Parametrelili Sistem Modeli İçin Tasarlanan Özayarlama Bulanık PID Tipi Kontrolör Sonuçlarının İrdelenmesi

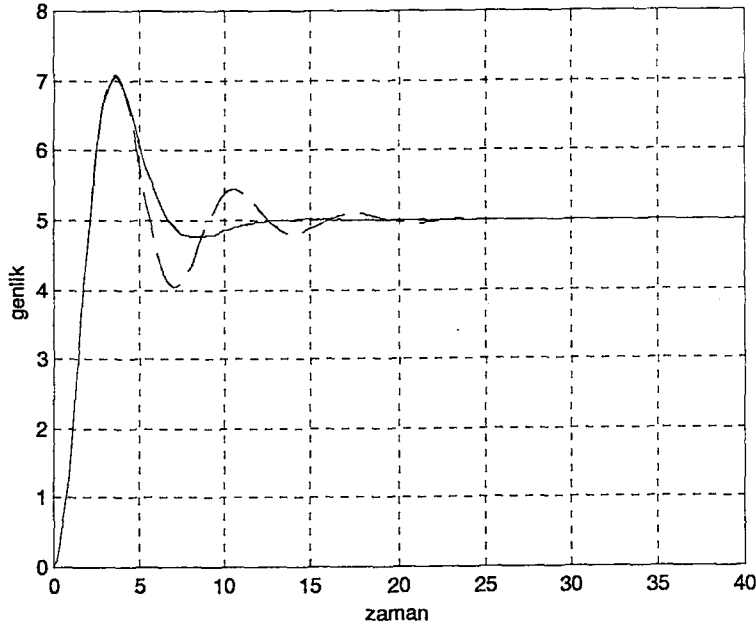
4.3.4.1 Aşım Gözleyicisi Yöntemi İle Yapılan Özayarlamanın Etkileri

Şekil 4.22 ve Şekil 4.24’ de görülen ölü zamansız sistem için ve Şekil 4.28 ile Şekil 4.30’ da görülen ölü zamanlı sistem için aşım gözleyicisi yoluyla parametreleri ayarlanan bulanık PID tipi kontrolöre ait gerçek sistem ve simülasyon yanıtları birbirine yakın olarak elde edilmiştir.

Transfer fonksiyonu $\frac{1}{(s+1)^2}$ olan ölü zamansız gerçek sisteme Bölüm 4.3.1’de tasarlanan özayarlama bulanık kontrolör ile aşım gözleyicisi ile ölçekleme faktörleri ayarlanan bulanık kontrolöre ait sistem yanıtları beraber olarak Şekil 4.50’ de görülmektedir. Parametre uyarlama yöntemi elde edilen tepe değerlerine bağlı olduğu için ilk tepe değerine kadar sistem yanıtında iyileştirme yapılamamaktadır. Ancak zamanla sistemin sönümü artırılarak sistem başarımı yükseltilmektedir. Buna karşın hesaplanan tepe değerlerinin zamanla çok küçük değerler alması, algoritmada mevcut olan ölçekleme faktörlerinin çarpım ve bölüm işlemleri nedeniyle çok küçük ve büyük değerlere ulaşması neticesinde sistem yanıtında sürekli hal hatası oluşmaktadır.



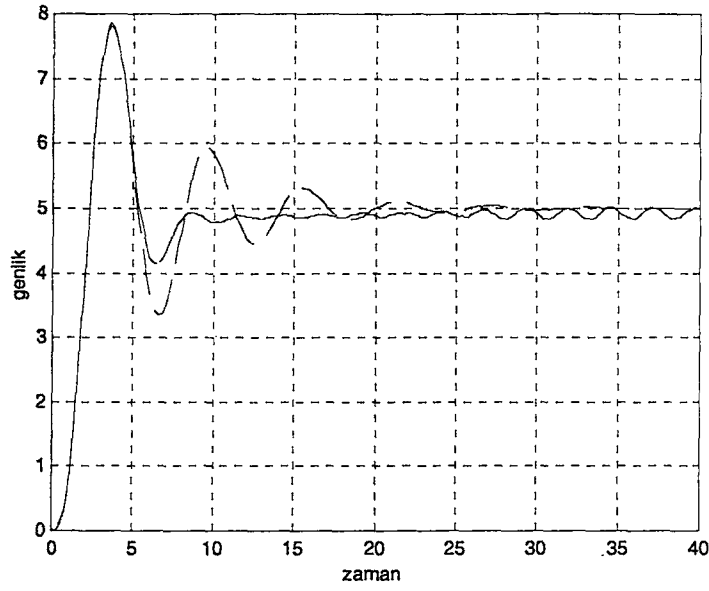
Şekil 4.50 Ölü zamansız gerçek sistem için özayarlama ve özayarlama bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: eşik değerine sahip olmayan kontrolöre ait sistem yanıtı, kesikli: özayarlama bulanık kontrolöre ait sistem yanıtı)



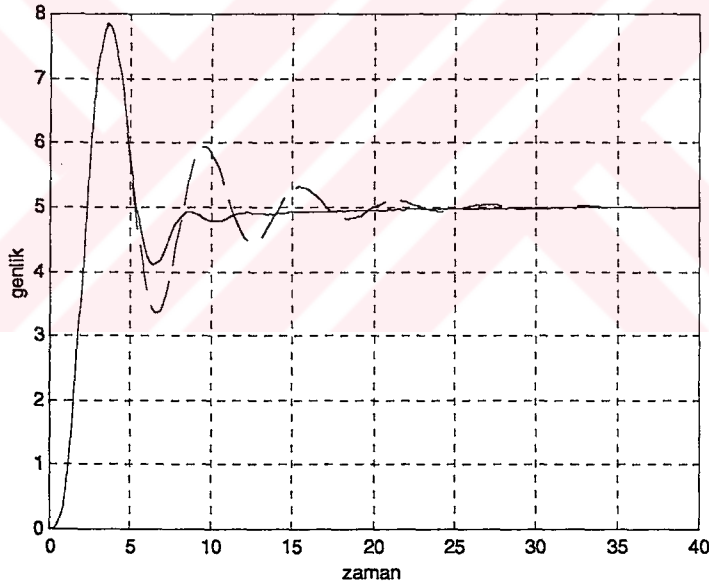
Şekil 4.51 Ölü zamansız gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: eşik değerine sahip olan kontrolöre ait sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)

Şekil 4.50' de verilen sistem yanıtında sürekli hal hatasının oluşması algoritmaya tepe değeri için 0.25V eşik değerinin konulması ile ortadan kaldırılmıştır. Eşik değerine sahip kontrolöre ait sistem yanıtı Şekil 4.51' de görülmektedir. Algoritmada, tepe mutlak değerlerinin 0.25V değeri altında hesaplanması durumunda bu mutlak değerlerin bir önceki değerlerini koruması sağlanmış, böylece sistem daha küçük yerleşme zamanına sahip olmuştur.

Ölü zamanlı gerçek sistem için özayarlamalı ve özayarlamasız bulanık kontrolörlere ait sistem yanıtlarının karşılaştırılmasına ilişkin grafik Şekil 4.52' de görülmektedir. Ölü zamansız sistem için ortaya çıkan sürekli hala hatası problemi bu sistem için de geçerli olmuş ve bu problem yine ayarlama algoritmasına 0.25V eşik değerinin ilave edilmesi ile ortadan kaldırılmıştır. Eşik değerine sahip kontrolöre ilişkin sistem yanıtı ise Şekil 4.53' de görülmektedir. Gerek ölü zamanlı gerekse ölü zamansız sistemler için, aşım gözleyicisi yöntemi sistem yanıtında ilk tepe değerinden sonra yerleşme zamanlarına ve mevcut salınımların ortadan kaldırılmasına ilişkin önemli derecede iyileştirmeler sağlamaktadır. Ancak ilk tepe değerinin oluşması anına kadar düzeltme yapılamaması sistemin yükselme zamanı değerinin aynı kalmasına neden olmaktadır.



Şekil 4.52 Ölü zamanlı gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: eşik değerine sahip olmayan kontrolöre ait sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)



Şekil 4.53 Ölü zamanlı gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: eşik değerine sahip olan kontrolöre ait sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)

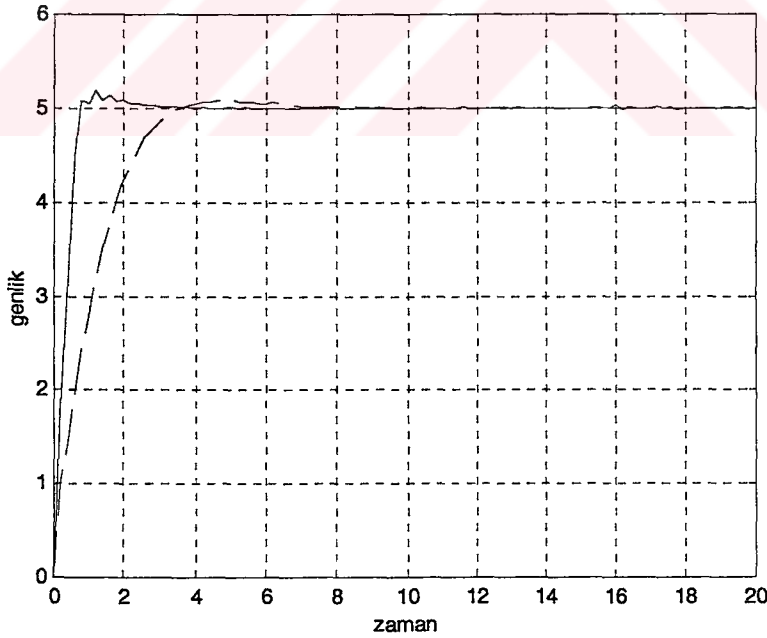
4.3.4.2 Fonksiyon Ayarlayıcısı Yöntemi İle Yapılan Özayarlamanın Etkileri

Şekil 4.34 ve Şekil 4.36' dan görüldüğü üzere ölü zamansız sistem için fonksiyon ayarlayıcısı yoluyla parametreleri uyarlanan bulanık PID tipi kontrolöre ait gerçek sistem ve simülasyon yanıtları benzer elde edilmekte ancak ölü zaman elemanının eklenmesi ile elde edilen dört parametrelili model ve gerçek sistem yanıtları arasında

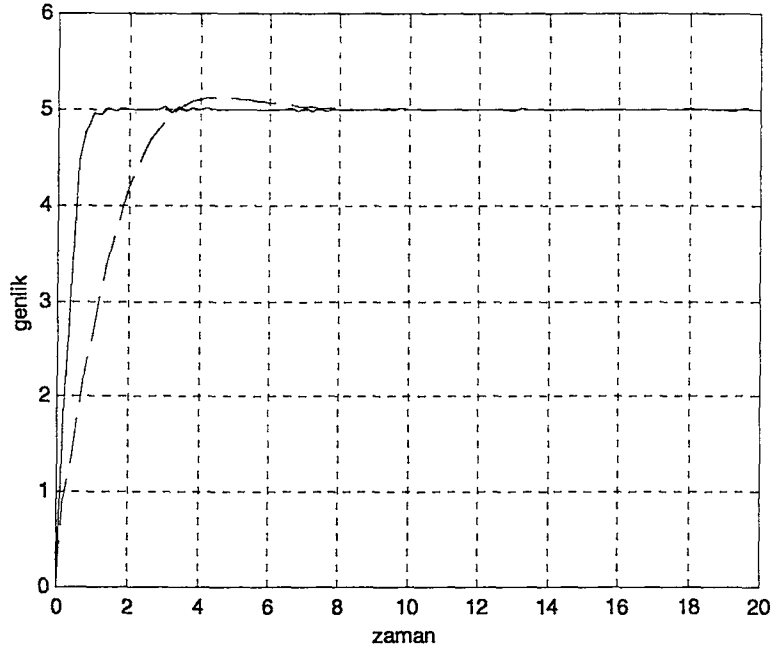
tasarlanan kontrolör için Şekil 4.38 ve Şekil 4.40' dan da görüldüğü gibi birtakım farklılıklar meydana gelmektedir. Gerçek sisteme ait yanıtla simülasyona ait yanıt arasında oluşan bu fark temelde iki sistemin sahip olduğu farklı ölü zaman değerlerinden oluşmaktadır. İki farklı ölü zaman değeri sebebiyle gerçek sistem yanıtı ile simülasyon yanıtı başlangıç safhasında farklı hata değerlerine sahip olmaktadır. Bu yöntemin sistem yanıtının sahip olduğu hata değerine bağlı olarak kontrol işareti üretmesi, sistem yanıtlarında kontrol işaretine bağlı olarak farklılıkların meydana gelmesine neden olmaktadır.

Bölüm 4.3.2' de ölü zamansız ve ölü zamanlı gerçek sistemler için elde edilen ölçekleme faktörleri düzgün olarak ayarlanamayan bulanık PID tipi kontrolör yanıtı ile fonksiyon ayarlayıcısı yoluyla parametreleri uyarlanan bulanık PID tipi kontrolör yanıtı beraber olarak sırasıyla Şekil 4.54, Şekil 4.55, Şekil 4.56 ve Şekil 4.57' de görülmektedir.

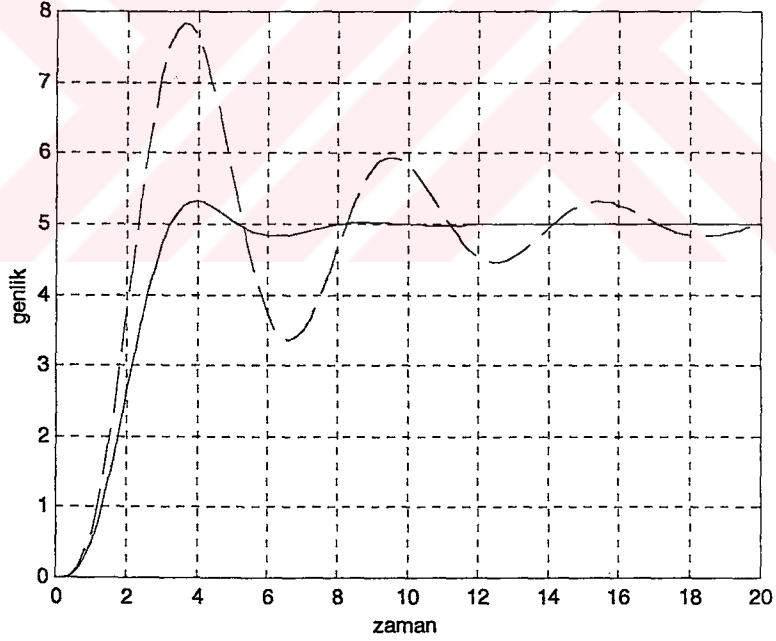
Şekil 4.54 ve Şekil 4.55' de ölü zamansız gerçek sistem için yanıtlara bakıldığında, özayarlamalı bulanık kontrolör ile birlikte sistemin yükselme ve yerleşme zamanlarında önemli derecede iyileştirme olduğu, aşım değerinde ise önemli bir değişikliğin olmadığı görülmektedir.



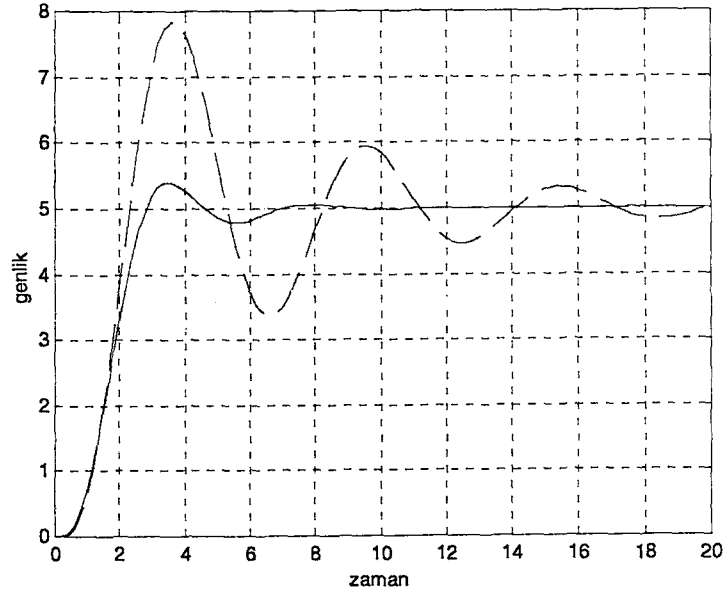
Şekil 4.54 Ölü zamansız gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: $a_1 = 2$, $a_2 = 8$, $b_1 = 0.5$, $b_2 = 4.5$ katsayıları için sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)



Şekil 4.55 Ölü zamansız gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: $a_1 = 2.3$, $a_2 = 7.5$, $b_1 = 0.3$, $b_2 = 5$ katsayıları için sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)



Şekil 4.56 Ölü zamanlı gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: $a_1 = 0.25$, $a_2 = 0.4$, $b_1 = 0.05$ ve $b_2 = 0.8$ katsayıları için sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)

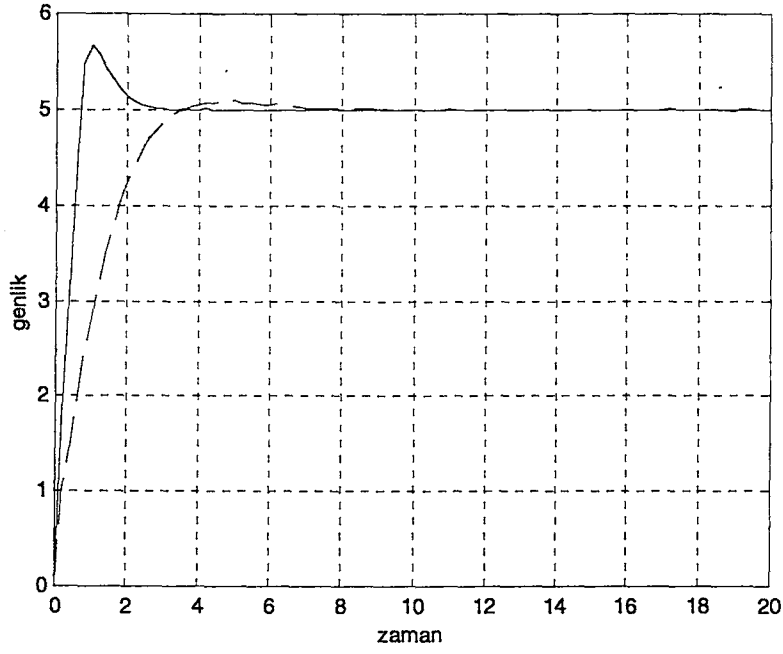


Şekil 4.57 Ölü zamanlı gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: $a_1 = 0.28, a_2 = 0.45, b_1 = 0.14$ ve $b_2 = 2.2$ katsayıları için sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)

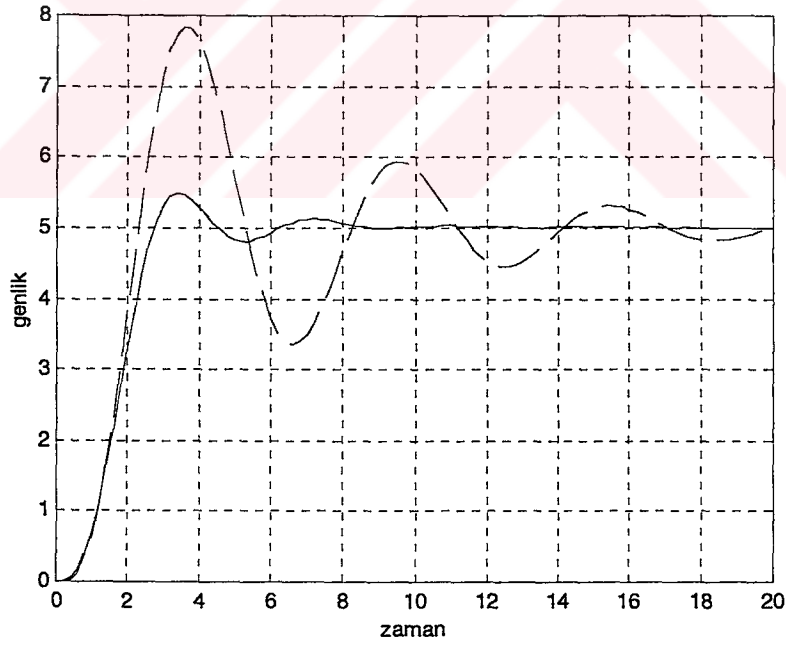
Şekil 4.56 ve Şekil 4.57' de ölü zamanlı gerçek sistem yanıtlarında ise hem aşım değerinde hem de yerleşme zamanına iyileşme olduğu, yükselme zamanında önemli bir değişikliğin olmadığı görülmektedir. Ayarlama mekanizması ile kontrolörün integral bileşeninin değeri zamanla azaltılarak sistemin sönümü artırılmakta dolayısıyla sistemde ilk etapta meydana gelen salınımlar ortadan kaldırılmaktadır. Ancak burada önemli bir nokta ise kontrolöre ait integral ve türev bileşenleri aynı oranda artırılıp azaltılmadığı için kontrolörün oransal kontrol bileşeni sabit değerde tutulamamakta ve her zaman hataya karşı olan hızlı reaksiyon garanti edilememektedir. Ayrıca sistem yanıtında iyileştirmenin sağlanabilmesi için kontrolöre ait fonksiyonların sahip olduğu dört adet pozitif katsayı değerlerinin hesaplanabilmesi uzun zaman almaktadır.

4.3.4.3 Bağlı Hız Gözleyicisi Yöntemi İle Yapılan Özayarlamanın Etkileri

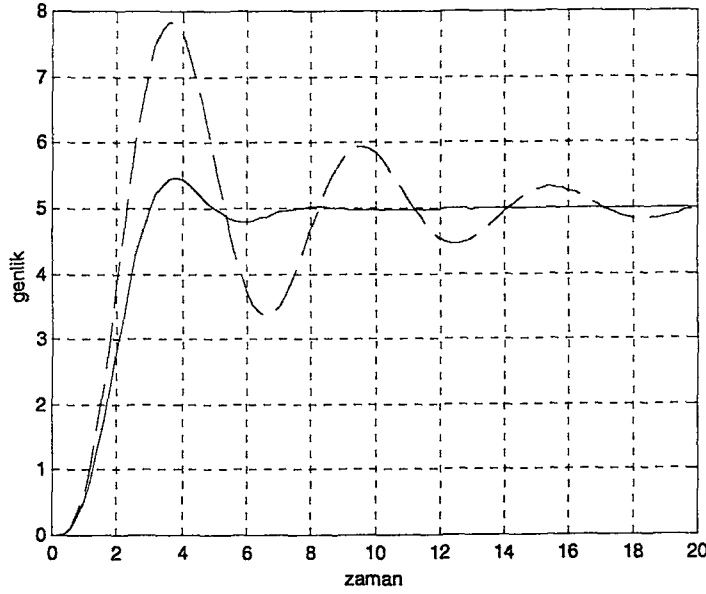
Şekil 4.42 ve Şekil 4.44' den görüldüğü üzere ölü zamansız sistem için bağlı hız gözleyicisi yoluyla parametreleri uyarlanan bulanık PID tipi kontrolöre ait gerçek sistem ve simülasyon yanıtları benzer elde edilmekte ancak ölü zaman elemanının eklenmesi ile elde edilen dört parametrelili model ve gerçek sistem yanıtları arasında tasarlanan kontrolör için Şekil 4.46 ve Şekil 4.48' den de görüldüğü gibi farklılıklar meydana gelmektedir. Gerçek sisteme ait yanıtla simülasyona ait yanıt arasında



Şekil 4.59 Ölü zamansız gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: $K_f = 22$ ve $K_{fd} = 0.21$ katsayıları için sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)



Şekil 4.60 Ölü zamanlı gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: $K_f = 6.45$ ve $K_{fd} = 1$ katsayıları için sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)



Şekil 4.61 Ölü zamanlı gerçek sistem için özayarlamalı ve özayarlamasız bulanık PID tipi kontrolörlere ait sistem yanıtları (düz: $K_f = 7.8$ ve $K_{fd} = 0.25$ katsayıları için sistem yanıtı, kesikli: özayarlamasız bulanık kontrolöre ait sistem yanıtı)

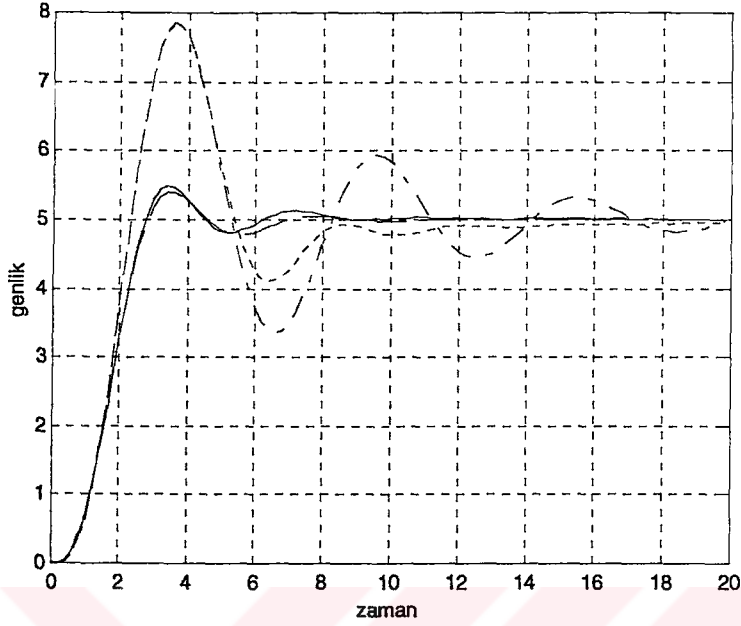
Ölü zamansız gerçek sistem yanıtlarında olduğu gibi, Şekil 4.60 ve Şekil 4.61' de görüldüğü üzere, ölü zamanlı gerçek sistem yanıtları için de benzer sonuçlar ortaya çıkmaktadır. Her iki durumda da aşım ve yerleşme zamanı kriterleri için iyileştirmeler sağlanmasına karşın, oransal kontrol şiddetinin korunması durumunda daha iyi bir yükselme zamanı değeri, oransal kontrol şiddetinin korunamaması durumunda ise daha iyi bir yerleşme zamanı değeri elde edilmektedir. Bu durumda aşım değerlerinde bir değişiklik meydana gelmemektedir.

Dikkat edilecek bir nokta ise, kontrolör tasarımı K_f ve K_{fd} olmak üzere bir yada en fazla iki parametre değerinin ayarlanması zaman ve uğraş açısından önemli bir avantaj sağlamaktadır.

4.3.4.4 Özayarlama Yöntemlerinin Karşılaştırılması

Şekil 4.62' de ölü zamanlı gerçek sisteme tasarlanan ölçekleme faktörleri $K_e = 0.9$, $K_{de} = 0.7$, $K_i = 0.4$, $K_d = 1$ olan özayarlamasız bulanık PID tipi kontrolöre ait sistem yanıtı ile aynı sisteme eşik değerine sahip aşım gözleyicisi, pozitif katsayıların $a_1 = 0.28$, $a_2 = 0.45$, $b_1 = 0.14$ ve $b_2 = 2.2$ olduğu fonksiyon ayarlayıcısı ve parametrelerin $K_f = 6.45$ ve $K_{fd} = 1$ olduğu bağıl hız gözleyicisi yöntemleri ayrı

ayrı kullanılarak tasarlanan özayarlama bulanık PID tipi kontrolöre ait sistem yanıtları beraber görülmektedir.



Şekil 4.62 Özayarlama yöntemlerinin karşılaştırılması (Kesikli noktalı : Ölçekleme faktörleri ayarlanamayan bulanık PID tipi kontrolöre ait yanıt, nokta nokta : Eşik değerli aşım gözleyicisi kullanılan kontrolöre ait yanıt, kesikli kesikli : fonksiyon ayarlayıcısı kullanılan kontrolöre ait yanıt, düz : bağıl hız gözleyicisi kullanılan kontrolöre ait yanıt)

Aşım gözleyicisi yöntemi ile ölçekleme faktörleri düzgün olarak ayarlanamayan kontrolöre ait sistem yanıtı ilk tepe oluşma anına kadar korunmuş, daha sonra ise oluşan tepe mutlak değerlerine bağlı olarak uyarlanan ölçekleme faktörleri ile daha kısa yerleşme zamanı elde edilmiş ve sistem sönümü artırılmıştır. Ancak ilk tepe anına kadar sistem yanıtında herhangi bir iyileştirme yapılamaması önemli bir olumsuzluk olarak göze çarpmıştır. Bu yöntemden daha iyi bir sonuç ise fonksiyon ayarlayıcısı yöntemi ile elde edilmiştir. Bu yöntem ile aşım gözleyicisi yöntemine oranla çok daha küçük bir aşım değeri ve kısa bir yerleşme zamanına ulaşılmıştır. Ancak ölçekleme faktörlerinin ayarlanmasına ilişkin algoritma nedeniyle tasarlanan kontrolöre ait oransal kontrol şiddeti sabit değerde tutulamamıştır ve dolayısıyla da sistemin hataya karşı olan hızlı tepkisi her zaman korunamamıştır. Ayrıca algoritmada mevcut iki adet fonksiyonun sahip olduğu dört pozitif katsayı değerinin tespiti oldukça güçtür. Bağıl hız gözleyicisi yöntemi kullanılarak elde edilen kontrolöre ait sistem yanıtı, fonksiyon ayarlayıcısı yöntemi ile elde edilen kontrolöre

ait sistem yanıtına oldukça benzemektedir. Ancak aynı başarıyı bu yöntem ile dört parametre yerine sadece bir parametrenin ayarlanması ile ve kontrolöre ait oransal kontrol şiddeti sabit tutulup sistemin hataya karşı olan hızlı tepkisi her zaman korunarak elde edilmiştir.



5. SONUÇLAR VE TARTIŞMA

Bu çalışmada klasik ve bulanık kontrol algoritmaları, PCS 327 Proses Kontrol Simülatörü deney seti üzerinde PLC cihazının kontrolör olarak kullanılması ile başarılı bir şekilde uygulanmıştır.

Basit bir yapıya sahip olmasından ötürü endüstride en fazla PID kontrolü kullanılmaktadır. Ancak sistemlerin birtakım doğrusal olmayan özelliklere ve karmaşık bir yapıya sahip olması durumunda PID kontrolörlerinin etkili bir başarımları sağlayamadıkları bilinmektedir. Bu gibi durumlarda bulanık kontrol algoritmaları alternatif bir çözüm haline gelmektedir. Endüstride kullanılan çoğu bulanık kontrolör PI yada PID kontrolörlere benzer yapılara sahiptir. Bulanık kurallar ve üyelik fonksiyonlarının kullanılması, doğrusal olmayan özelliklerin kontrolöre eklenmesini kolaylaştırmaktadır. Dolayısıyla hem doğrusal olan hem de doğrusal olmayan sistemler için etkili ve basit kullanımı, bulanık PID kontrolörlerinin endüstride geniş bir alanda kullanılmasını sağlamaktadır.

Literatürde değişik yapılarda bulanık PID kontrolörleri olmasına karşın, bu tezde yapıları giriş - çıkış bilgisi açısından bir PID kontrolörüne benzeyen, kontrol sinyalinin çıkarım mekanizması ve bilgi tabanı tarafından üretildiği, hata ve hatanın değişimi olmak üzere iki adet giriş ve bir kural tabanına sahip olan bulanık PID tipi kontrolörler ele alınmıştır. Burada kontrolörün sahip olduğu dört adet ölçekleme faktörü tasarım başlangıcında güncel olmayan (off-line) şekilde ayarlandığı için bu tip kontrolörler özayarlamasız bulanık PID tipi kontrolörlerdir. PCS 327 Proses Kontrol Simülatöründe oluşturulan sistemlerin, basamak yanıtı modelleme yöntemlerinden üç ve dört adet parametreye sahip yapılar ile modellenmesi sonucu elde edilen sistem modellerine kontrolör tasarımı neticesinde, ölçekleme faktörlerinin düzgün ayarlanması durumunda özayarlamasız bulanık PID tipi kontrolörlerin PID kontrolörlerine nazaran daha iyi bir sistem yanıtına sahip olduğu görülmüştür.

Bu tip kontrolörün başarımı da, literatürde mevcut olan ve bulanık PID tipi kontrolörlere ait dört adet ölçekleme faktöründen iki tanesinin, aşım gözleyicisi, fonksiyon ayarlayıcısı ve bağıl hız gözleyicisi olmak üzere üç adet ayarlama algoritmasının ayrı ayrı kullanılmasıyla uyarlandığı, özayarlamalı bulanık PID tipi kontrolörler ile artırılmıştır. Tasarlanan kontrolörler neticesinde, aşım gözleyicisi ile parametre uyarlamalı yöntem ile ilk tepe oluşma anına kadar sistem yanıtı korunmuş, ancak daha sonra oluşan tepe zamanlarında tepe değerine bağlı olarak ayarlanan ölçekleme faktörleri ile sistem yanıtının daha küçük aşım ve yerleşme zamanı değerlerine sahip olması sağlanmıştır. Ancak bu ayarlama yönteminden kaynaklanan, ilk tepe anına kadar olan zaman diliminde sistem yanıtında iyileştirme yapılamamasına dair olumsuzluk, fonksiyon ayarlayıcı yoluyla parametre uyarlamalı yöntemin uygulanması ile giderilmiştir. Bu yöntemin kullanıldığı kontrolör ile bir önceki yöntemle kıyasla, sistem yanıtında daha küçük bir aşım ve yerleşme zamanı değerine ulaşılmıştır. Buna karşın ayarlama algoritmasından ötürü kontrolöre ait oransal kontrol şiddeti sabit değerde tutulmadığından, sistemin hataya karşı olan hızlı reaksiyonu her zaman korunamamıştır. Ayrıca sistem yanıtında iyileştirme yapılabilmesi amacıyla ayarlama algoritmasında kullanılan iki adet deneysel fonksiyonun sahip olduğu dört pozitif katsayının genetik arama algoritması yoluyla ayarlanması zaman ve emek açısından önemli bir olumsuzluk oluşturmuştur.

Fonksiyon ayarlayıcısı yoluyla parametre uyarlamalı yöntem ile elde edilen küçük aşım değeri ve hızlı yükselme kriterleri, bağıl hız gözleyicisi yoluyla parametre uyarlamalı yöntem ile ve sadece iki parametreden bir tanesi "1" değerinde sabit tutularak, diğeri uyarlanarak elde edilmiş ve böylelikle oransal kontrol şiddeti değiştirilmeyerek hataya karşı olan hızlı reaksiyon her zaman korunmuştur. Bu kriterlere ilave olarak oransal kontrol şiddetinin değiştirilerek iki adet parametrenin uyarlanması ile sistem yanıtında fonksiyon ayarlayıcı yöntemine nazaran daha da küçük bir yerleşme zamanı elde edilmiştir. Bu yöntem ile uyarlanacak parametre sayısında önemli bir avantaj sağlanmıştır. Dolayısıyla gerek kontrolöre ait oransal kontrol şiddetinin korunması gerekse korunmaması durumlarında, yükselme zamanı, aşım değeri ve yerleşme zamanı gibi sistem yanıtı kriterleri için bağıl hız gözleyicisi yoluyla parametre uyarlamalı yöntem en kullanışlı ayarlama mekanizmasıdır.

Bu konularda daha çok çalışmanın yapılabileceği aşıkardır. Bu çalışmalar; mevcut kontrol algoritmalarının daha fazla fiziksel özellik gösteren ısı kontrol setine, çift

tank seviye kontrol deney setine veya daha başka proseslere uygulanması olarak sıralanabilir. Ayrıca elde edilen sonuçların başarımını daha fazla artırmak maksadıyla, mevcut üç adet ayarlama algoritması üzerinde iyileştirmeler yapılabilir yada bunlara ilave olarak değişik uyarlama mekanizmaları geliştirilebilir.



KAYNAKLAR

- Aström, K. And Höggund, T., 1995. PID Controllers: Theory, Design and Tuning, Instrument Society of America, New York.
- Bellman, R. E. and Zadeh, L. A., 1970. Decision-Making in a fuzzy environment, Management Science, 17.
- Eksin, I., Güzelkaya, M. and Gürleyen, F., 2001. A new methodology for deriving the rule-base of a fuzzy logic controller with a new internal structure, Engineering Applications of Artificial Intelligence, 14(5), 617-628.
- Galichet, S. and Foulloy, L., 1995. Fuzzy Controllers: Synthesis and equivalences, IEEE Trans on Fuzzy Systems, 3, 140-148.
- Golob, M., 2001. Decomposed fuzzy proportional-integral-derivative controllers, Applied Soft Computing, 1, 201-204.
- Güzelkaya, M., Eksin, I. and 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.
- Güzelkaya, M., Eksin, I. and Yesil, E., 2003. Self-tuning of PID-type fuzzy logic controller coefficients via relative rate observer, Applications of Artificial Intelligence, 16, 227-236.
- Hu, B., Mann, G. K. I. and Gosine, R. G., 1999. A new methodology for a analytical and optimal design of fuzzy PID controllers, IEEE Trans on Fuzzy Systems, 7(5), 521-539.
- Kuo, B. C., 1992. Digital Control Systems, University of Illinois, Urbana-Champaign.
- Kurtulan, S., 2001. PLC ile endüstriyel otomasyon, Birsen Yayınevi, İstanbul.
- Kwok, D. P., Tam, P., Li, C. K. and Wang, P., 1990. Linguistic PID controllers, Proceedings of the 11th World Congress, Tallin, Estonia, 7, 192-197.
- Li, H. X. and Gatland, H. B., 1996. Conventional fuzzy control and its enhancement, IEEE Trans on Systems, Man, Cybernetics-Part(B), 26(5), 791-797.
- Mac Vicar-Whelan, P. J., 1976. Fuzzy sets for man-machine interaction, Int. J. Man-Machine Studies, 8, 687-697.
- Mamdani, E. H., 1974. Application of fuzzy algorithms for control of simple dynamic plant, Proc. Inst. Elect. Eng. Contr. Sci., vol 121, 1585-1588.
- Mamdani, E. H. and Assilian, S., 1975. An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man-Machine Stud., 7, 1-13.
- Mizumoto, M., 1990. Fuzzy controls by product-sum gravity method, International Academic Publishers, C. 1. 1-C. 1. 4.

- Mizumoto, M.**, 1992. Realization of PID controllers by fuzzy control methods, *Fuzzy Sets and Systems*, 70, 171-182.
- Ollero, A. and Garcia-Crezo, A. J.**, 1989. Direct digital control, auto-tuning and supervision using fuzzy logic, *Fuzzy Sets and Systems*, 30(2), 135-153.
- Palm, R. and Driankov, D.**, 1996. *Model Based Fuzzy Control*, Springer.
- Peng, X. T.**, 1990. Generating rules for fuzzy logic controllers by functions, *Fuzzy Sets and Systems*, 36(2), 83-89.
- Qiao, W. Z. and Mizumoto, M.**, 1996. PID type fuzzy controller and parameters adaptive method, *Fuzzy Sets and Systems*, 78, 23-35.
- Qin, S. J. and Borders, G.**, 1994. A multiregion fuzzy logic controller for nonlinear process control, *IEEE Trans on Fuzzy Systems*, 2, 74-81.
- Takagi, T. and Sugeno, M.**, 1985. Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans Systems, Man, Cybern.*, vol. SMC-15, pp. 116-132.
- Tang, M. L. and Mulholland, R. J.**, 1987. Comparing fuzzy logic with classical controller designs, *IEEE Trans on Systems, Man, Cybernetics*, 17(6), 1085-1087.
- Vega, P., Prada, C. and Aleixander, V.**, 1991. Self-tuning predictive PID controller, *IEEE Proc. D.*, 138(3), 303-311.
- Woo, Z. W., Chung, H. Y. and Lin, J. J.**, 2000. A PID-type fuzzy controller with self-tuning scaling factors, *Fuzzy Sets and Systems*, 115, 321-326.
- Xu, J. X., Liu, C. and Hang, C. C.**, 1998. Tuning and analysis of a fuzzy PI controller based on gain and phase margins, *IEEE Trans on Systems, Man, Cybernetics-Part(A)*, 28(5), 685-691.
- Xu, J. X., Hang, C. C. and Liu, C.**, 2000. Parallel structure and tuning of a fuzzy PID controller, *Automatica*, 36, 673-684.
- Zadeh, L. A.**, 1965. Fuzzy sets, *Information and Control*, 8, 338-353.

EK A

Deney seti üzerinde oluşturulan ölü zamansız ve ölü zamanlı sistemlerin modellenmesinde basamak yanıtların elde edilmesi amacıyla PLC' de yazılan program aşağıda verilmiştir.

// ANA PROGRAM

NETWORK 1

LD SM0.1

MOVW +16000, VW4040

MOVB 200, SMB35

ATCH 0, 11

ENI

LD I0.0

EU

MOVW VW4040, AQW0

LD I0.0

ED

MOVW +0, AQW0

LD I0.0

TON T37, +200

LD Q0.0

ED

MOVW +0, AQW0

// KESME ALT PROGRAMI

NETWORK 1

LD I0.1

EU

MOVD &VB0, VD2004

NETWORK 2

LD I0.0

AN T37

MOVW AIW0, *VD2004

+D +2, VD2004

= Q0.0

EK B

Aşağıda Eşitlik 3.4 ile verilen algoritmadaki gibi girişin hata olması durumunda oran, integral ve türev bloklarının ayrı ayrı hesaplanıp daha sonra bu blokların toplanarak kontrol işaretinin elde edilmesine ilişkin PID kontrolör programı verilmiştir.

// ANA PROGRAM

```
NETWORK 1
LD SM0.1
MOVB 200, SMB34
ATCH 0, 10
ENI
CALL 0
LD I0.0
TON T37, +200
LD I0.0
ED
MOVW +0, AQW0
LD T37
EU
MOVW +0, AQW0
```

// 0 NO' LU ALT PROGRAM

```
NETWORK 1
LD SM0.0
MOVR 0.2, "TS"
MOVR 32000.0, "UMAX"
MOVR 0.0, "UMIN"
MOVR 1.0, "KP"
MOVR 0.7124, "KI"
MOVR 0.1572, "KD"
MOVW +16000, "REF"
MOVR 0.0, "E_1"
MOVR 0.0, "U_1"
```

// KESME ALT PROGRAMI

```
NETWORK 1
LD I0.1
EU
```

```
MOVD &VB1700, VD3510
MOVD &VB2000, VD200
LD I0.0
AN T37
MOVW AIW0, "Y"
XORD AC0, AC0
MOVW "REF", AC0
-I "Y", AC0
LDW>= AC0, +0
JMP 0
NOT
ORD 16#FFFF0000, AC0
LBL 0
LD I0.0
AN T37
DTR AC0, AC0
MOVR AC0, "E"
*R "KP", AC0
MOVR AC0, "UP"
MOVR "E", AC0
+R "E_1", AC0
*R "TS", AC0
*R "KI", AC0
/R 2.0, AC0
+R "U_1", AC0
MOVR AC0, "UI"
MOVR "E", AC0
-R "E_1", AC0
*R "KD", AC0
/R "TS", AC0
MOVR AC0, "UD"
MOVR 0.0, AC0
MOVR "UP", AC0
+R "UI", AC0
+R "UD", AC0
MOVR AC0, "US"
LDR>= "US", "UMAX"
```

```
MOVR "UMAX", "US"  
LDR<= "US", "UMIN"  
MOVR "UMIN", "US"  
LD I0.0  
AN T37  
MOVR "E", "E_1"  
MOVR "UI", "U_1"  
MOVR "US", AC0  
TRUNC AC0, AC0  
MOVW AC0, AQW0  
MOVW AIW0, *VD3510  
+D +2, VD3510  
MOVW AC0, *VD200  
+D +2, VD200
```



EK C1

Aşağıda özayarlanmasız bulanık PID tipi kontrolöre ait PLC programı verilmiştir.

// ANA PROGRAM

```
NETWORK 1
LD SM0.1
MOVB 200, SMB34
ATCH 0, 10
ENI
CALL 0
LD I0.0
TON T37, +200
LD I0.0
ED
MOVW +0, AQW0
LD T37
EU
MOVW +0, AQW0

// 0 NO' LU ALT PROGRAM
```

```
NETWORK
LD SM0.0
MOVR 0.0, "E_1"
MOVR 0.0, "U_1"
MOVR 1.0, "U1"
MOVR 0.0, "U2"
MOVR 0.0, "U3"
MOVR -1.0, "U4"
MOVR 19200.0, "KI"
MOVR 32000.0, "KD"
MOVR 32000.0, "UMAX"
MOVR 0.0, "UMIN"
MOVW +16000, "REF"
MOVR 2.8125E-005, "KE"
MOVR 1.875E-005, "KDE"
```

// KESME ALT PROGRAMI

```
NETWORK 1
LD I0.1
EU
MOVD &VB1700, VD3510
MOVD &VB2000, VD200
LD I0.0
AN T37
MOVW AIW0, "Y"
XORD AC0, AC0
MOVW "REF", AC0
-I "Y", AC0
LDW>= AC0, +0
JMP 0
LDW<= AC0, +0
ORD 16#FFFFF0000, AC0
LBL 0
DTR AC0, AC0
MOVR AC0, "E1"
*R "KE", AC0
MOVR AC0, "E"
LD I0.0
AN T37
MOVR "E1", AC0
-R "E_1", AC0
*R "KDE", AC0
MOVR AC0, "DELTAE"
LDR>= "E", 1.0
XORD "E", "E"
MOVR 1.0, "E"
JMP 20
LDR<= "E", -1.0
XORD "E", "E"
MOVR -1.0, "E"
LBL 20
LDR>= "DELTAE", 1.0
XORD "DELTAE", "DELTAE"
MOVR 1.0, "DELTAE"
```

```

JMP 21
LDR<= "DELTAE", -1.0
XORD "DELTAE", "DELTAE"
MOVR -1.0, "DELTAE"
LBL 21
LD I0.0
AN T37
XORD AC0, AC0
MOVR "E", AC0
+R 1.0, AC0
/R 2.0, AC0
MOVR AC0, "W1"
XORD AC0, AC0
MOVR "DELTAE", AC0
+R 1.0, AC0
/R 2.0, AC0
MOVR AC0, "W2"
XORD AC0, AC0
MOVR 1.0, AC0
-R "E", AC0
/R 2.0, AC0
MOVR AC0, "W3"
XORD AC0, AC0
MOVR 1.0, AC0
-R "DELTAE", AC0
/R 2.0, AC0
MOVR AC0, "W4"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "W1", AC0
*R "W2", AC0
MOVR AC0, "B1"
*R "U1", AC0
MOVR AC0, "U11"
XORD AC0, AC0
MOVR "W1", AC0
*R "W4", AC0
MOVR AC0, "B2"
*R "U2", AC0
MOVR AC0, "U22"
XORD AC0, AC0
MOVR "W3", AC0
*R "W2", AC0
MOVR AC0, "B3"
*R "U3", AC0
MOVR AC0, "U33"
XORD AC0, AC0
MOVR "W3", AC0
*R "W4", AC0

```

```

MOVR AC0, "B4"
*R "U4", AC0
MOVR AC0, "U44"
XORD AC0, AC0
MOVR "B1", AC0
+R "B2", AC0
+R "B3", AC0
+R "B4", AC0
MOVR AC0, "TOPB"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "U11", AC0
+R "U22", AC0
+R "U33", AC0
+R "U44", AC0
MOVR AC0, "U"
/R "TOPB", AC0
MOVR AC0, "U"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "U", AC0
*R "KD", AC0
MOVR AC0, "UD"
XORD AC0, AC0
MOVR "U", AC0
*R "KI", AC0
+R "U_1", AC0
MOVR AC0, "UI"
XORD AC0, AC0
MOVR "UI", AC0
+R "UD", AC0
MOVR AC0, "UT"
LDR>= "UT", "UMAX"
MOVR "UMAX", "UT"
LDR<= "UT", "UMIN"
MOVR "UMIN", "UT"
LD I0.0
AN T37
MOVR "UT", AC0
TRUNC AC0, AC0
MOVR "E1", "E_1"
MOVR "UI", "U_1"
MOVW AC0, AQW0
MOVW AIW0, *VD3510
+D +2, VD3510
MOVW AC0, *VD200
+D +2, VD200

```

EK C2

Aşağıda eşik değerine sahip olmayan aşım gözleyicisi yöntemi kullanılarak tasarlanan özayarlamalı bulanık PID tipi kontrolöre ait PLC programı verilmiştir.

// ANA PROGRAM

```
NETWORK 1
LD SM0.1
MOVB 200, SMB34
ATCH 0, 10
ENI
CALL 0
LD I0.0
TON T37, +400
LD I0.0
ED
MOVW +0, AQW0
LD T37
EU
MOVW +0, AQW0
```

// 0 NO' LU ALT PROGRAM

```
NETWORK 1
LD SM0.0
MOVR 0.0, "E_1"
MOVR 0.0, "U_1"
MOVR 1.0, "U1"
MOVR 0.0, "U2"
MOVR 0.0, "U3"
MOVR -1.0, "U4"
MOVR 19200.0, "KIS"
MOVR 32000.0, "KD"
MOVR 32000.0, "UMAX"
MOVR 0.0, "UMIN"
MOVW +16000, "REF"
MOVR 2.8125E-005, "KE"
MOVR 1.875E-005, "KDES"
MOVR 0.0, "E_2"
MOVR 16000.0, "MP_1"
MOVR 1.875E-005, "KDE"
MOVR 19200.0, "KI"
```

```
MOVR 19200.0, "KI_1"
MOVR 1.875E-005, "KDE_1"
```

// KESME ALT PROGRAMI

```
NETWORK 1
LD SM0.0
XORD "E", "E"
XORD "E1", "E1"
XORD "MP", "MP"
XORD "MPN", "MPN"
XORD "FARK1", "FARK1"
XORD "FARK2", "FARK2"
XORD "W1", "W1"
XORD "W2", "W2"
XORD "W3", "W3"
XORD "W4", "W4"
XORD "B1", "B1"
XORD "B2", "B2"
XORD "B3", "B3"
XORD "B4", "B4"
XORD "DELTA E", "DELTA E"
XORD "U11", "U11"
XORD "U22", "U22"
XORD "U33", "U33"
XORD "U44", "U44"
XORD "TOPB", "TOPB"
XORD "U", "U"
XORD "UU", "UU"
XORD "UT", "UT"
LD I0.1
EU
MOVD &VB1600, VD1002
MOVD &VB2700, VD200
MOVD &VB2100, VD3600
LD I0.0
```

```

AN T37
MOVW AIW0, "Y"
XORD AC0, AC0
MOVW "REF", AC0
-I "Y", AC0
LDW>= AC0, +0
JMP 0
LDW<= AC0, +0
ORD 16#FFFF0000, AC0
LBL 0
DTR AC0, AC0
MOVR AC0, "E1"
*R "KE", AC0
MOVR AC0, "E"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "E1", AC0
-R "E_1", AC0
MOVR AC0, "FARK1"
XORD AC0, AC0
MOVR "E_1", AC0
-R "E_2", AC0
MOVR AC0, "FARK2"
LDR= "E_1", 0.0
AR= "E_2", 0.0
MOVR "MP_1", "MP"
JMP 5
LD I0.0
AN T37
LDR>= "E1", 0.0
JMP 2
LDR<= "E1", 0.0
JMP 3
LBL 2
LDR<= "FARK1", 0.0
AR>= "FARK2", 0.0
XORD "MP", "MP"
MOVR "E1", "MP"
XORD AC0, AC0
MOVR "MP", AC0
/R 16000.0, AC0
MOVR AC0, "MPN"
XORD AC0, AC0
MOVR "KDES", AC0
/R "MPN", AC0
MOVR AC0, "KDE"
XORD AC0, AC0
MOVR "KIS", AC0
MOVR "KIS", AC0
*R "MPN", AC0

```

```

MOVR AC0, "KI"
JMP 5
NOT
XORD "MP", "MP"
MOVR "MP_1", "MP"
XORD AC0, AC0
MOVR "MP", AC0
/R 16000.0, AC0
MOVR AC0, "MPN"
MOVR "KDE_1", "KDE"
MOVR "KI_1", "KI"
JMP 5
LBL 3
LDR>= "FARK1", 0.0
AR<= "FARK2", 0.0
XORD "MP", "MP"
XORD AC0, AC0
MOVR "E1", AC0
*R -1.0, AC0
MOVR AC0, "MP"
/R 16000.0, AC0
MOVR AC0, "MPN"
XORD AC0, AC0
MOVR "KDES", AC0
/R "MPN", AC0
MOVR AC0, "KDE"
XORD AC0, AC0
MOVR "KIS", AC0
*R "MPN", AC0
MOVR AC0, "KI"
JMP 5
NOT
XORD "MP", "MP"
MOVR "MP_1", "MP"
XORD AC0, AC0
MOVR "MP", AC0
/R 16000.0, AC0
MOVR AC0, "MPN"
MOVR "KDE_1", "KDE"
MOVR "KI_1", "KI"
JMP 5
LBL 5
LD I0.0
AN T37
XORD "MP_1", "MP_1"
MOVR "MP", "MP_1"
XORD AC0, AC0
MOVR "E1", AC0
-R "E_1", AC0
*R "KDE", AC0

```

```

MOVR AC0, "DELTAE"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "E", AC0
+R 1.0, AC0
/R 2.0, AC0
MOVR AC0, "W1"
XORD AC0, AC0
MOVR "DELTAE", AC0
+R 1.0, AC0
/R 2.0, AC0
MOVR AC0, "W2"
XORD AC0, AC0
MOVR 1.0, AC0
-R "E", AC0
/R 2.0, AC0
MOVR AC0, "W3"
XORD AC0, AC0
MOVR 1.0, AC0
-R "DELTAE", AC0
/R 2.0, AC0
MOVR AC0, "W4"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "W1", AC0
*R "W2", AC0
MOVR AC0, "B1"
*R "U1", AC0
MOVR AC0, "U11"
XORD AC0, AC0
MOVR "W1", AC0
*R "W4", AC0
MOVR AC0, "B2"
*R "U2", AC0
MOVR AC0, "U22"
XORD AC0, AC0
MOVR "W3", AC0
*R "W2", AC0
MOVR AC0, "B3"
*R "U3", AC0
MOVR AC0, "U33"
XORD AC0, AC0
MOVR "W3", AC0
*R "W4", AC0
MOVR AC0, "B4"
*R "U4", AC0
MOVR AC0, "U44"
XORD AC0, AC0

```

```

MOVR "B1", AC0
+R "B2", AC0
+R "B3", AC0
+R "B4", AC0
MOVR AC0, "TOPB"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "U11", AC0
+R "U22", AC0
+R "U33", AC0
+R "U44", AC0
MOVR AC0, "U"
/R "TOPB", AC0
MOVR AC0, "UU"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "UU", AC0
*R "KD", AC0
MOVR AC0, "UD"
XORD AC0, AC0
MOVR "UU", AC0
*R "KI", AC0
+R "U_1", AC0
MOVR AC0, "UI"
XORD AC0, AC0
MOVR "UI", AC0
+R "UD", AC0
MOVR AC0, "UT"
LDR>= "UT", "UMAX"
MOVR "UMAX", "UT"
LDR<= "UT", "UMIN"
MOVR "UMIN", "UT"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "UT", AC0
TRUNC AC0, AC0
XORD "E_2", "E_2"
MOVR "E_1", "E_2"
XORD "E_1", "E_1"
MOVR "E1", "E_1"
XORD "U_1", "U_1"
MOVR "UI", "U_1"
MOVR "KDE", "KDE_1"
MOVR "KI", "KI_1"
MOVW AC0, AQW0
MOVW AIW0, *VD1002
+D +2, VD1002

```

```
MOVD "MP", *VD200  
+D +4, VD200  
MOVW AC0, *VD3600  
+D +2, VD3600
```



EK C3

Aşağıda eşik değerine sahip olan aşım gözleyicisi yöntemi kullanılarak tasarlanan özayarlamalı bulanık PID tipi kontrolöre ait PLC programı verilmiştir.

// ANA PROGRAM

```
NETWORK 1
LD SM0.1
MOVB 200, SMB34
ATCH 0, 10
ENI
CALL 0
LD I0.0
TON T37, +400
LD I0.0
ED
MOVW +0, AQW0
LD T37
EU
MOVW +0, AQW0
```

// 0 NO' LU ALT PROGRAM

```
NETWORK 1
LD SM0.0
MOVR 0.0, "E_1"
MOVR 0.0, "U_1"
MOVR 1.0, "U1"
MOVR 0.0, "U2"
MOVR 0.0, "U3"
MOVR -1.0, "U4"
MOVR 19200.0, "KIS"
MOVR 32000.0, "KD"
MOVR 32000.0, "UMAX"
MOVR 0.0, "UMIN"
MOVW +16000, "REF"
MOVR 2.8125E-005, "KE"
MOVR 1.875E-005, "KDES"
MOVR 0.0, "E_2"
MOVR 16000.0, "MP_1"
MOVR 1.875E-005, "KDE"
```

```
MOVR 19200.0, "KI"
MOVR 19200.0, "KI_1"
MOVR 1.875E-005, "KDE_1"
```

// KESME ALT PROGRAMI

```
NETWORK 1
LD SM0.0
XORD "E", "E"
XORD "E1", "E1"
XORD "MP", "MP"
XORD "MPN", "MPN"
XORD "FARK1", "FARK1"
XORD "FARK2", "FARK2"
XORD "W1", "W1"
XORD "W2", "W2"
XORD "W3", "W3"
XORD "W4", "W4"
XORD "B1", "B1"
XORD "B2", "B2"
XORD "B3", "B3"
XORD "B4", "B4"
XORD "DELTA E", "DELTA E"
XORD "U11", "U11"
XORD "U22", "U22"
XORD "U33", "U33"
XORD "U44", "U44"
XORD "TOPB", "TOPB"
XORD "U", "U"
XORD "UU", "UU"
XORD "UT", "UT"
LD I0.1
EU
MOVD &VB1600, VD1002
MOVD &VB2700, VD200
MOVD &VB2100, VD3600
```

```

LD I0.0
AN T37
MOVW AIW0, "Y"
XORD AC0, AC0
MOVW "REF", AC0
-I "Y", AC0
LDW>= AC0, +0
JMP 0
LDW<= AC0, +0
ORD 16#FFFF0000, AC0
LBL 0
DTR AC0, AC0
MOVR AC0, "E1"
*R "KE", AC0
MOVR AC0, "E"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "E1", AC0
-R "E_1", AC0
MOVR AC0, "FARK1"
XORD AC0, AC0
MOVR "E_1", AC0
-R "E_2", AC0
MOVR AC0, "FARK2"
LDR= "E_1", 0.0
AR= "E_2", 0.0
MOVR "MP_1", "MP"
JMP 5
LD I0.0
AN T37
LDR>= "E1", 0.0
JMP 2
LDR<= "E1", 0.0
JMP 3
LBL 2
LDR<= "FARK1", 0.0
AR>= "FARK2", 0.0
JMP 10
NOT
JMP 11
LBL 10
XORD "MP", "MP"
MOVR "E1", "MP"
LDR>= "MP", 800.0
JMP 12
NOT
JMP 11
LBL 12
XORD AC0, AC0

```

```

MOVR "MP", AC0
/R 16000.0, AC0
MOVR AC0, "MPN"
XORD AC0, AC0
MOVR "KDES", AC0
/R "MPN", AC0
MOVR AC0, "KDE"
XORD AC0, AC0
MOVR "KIS", AC0
*R "MPN", AC0
MOVR AC0, "KI"
JMP 5
LBL 11
XORD "MP", "MP"
MOVR "MP_1", "MP"
XORD AC0, AC0
MOVR "MP", AC0
/R 16000.0, AC0
MOVR AC0, "MPN"
MOVR "KDE_1", "KDE"
MOVR "KI_1", "KI"
JMP 5
LBL 3
LDR>= "FARK1", 0.0
AR<= "FARK2", 0.0
JMP 6
NOT
JMP 7
LBL 6
XORD "MP", "MP"
XORD AC0, AC0
MOVR "E1", AC0
*R -1.0, AC0
MOVR AC0, "MP"
LDR>= "MP", 800.0
JMP 8
NOT
JMP 7
LBL 8
/R 16000.0, AC0
MOVR AC0, "MPN"
XORD AC0, AC0
MOVR "KDES", AC0
/R "MPN", AC0
MOVR AC0, "KDE"
XORD AC0, AC0
MOVR "KIS", AC0
*R "MPN", AC0
MOVR AC0, "KI"
JMP 5

```

```

LBL 7
XORD "MP", "MP"
MOVR "MP_1", "MP"
XORD AC0, AC0
MOVR "MP", AC0
/R 16000.0, AC0
MOVR AC0, "MPN"
MOVR "KDE_1", "KDE"
MOVR "KI_1", "KI"
JMP 5
LBL 5
LD I0.0
AN T37
XORD "MP_1", "MP_1"
MOVR "MP", "MP_1"
XORD AC0, AC0
MOVR "E1", AC0
-R "E_1", AC0
*R "KDE", AC0
MOVR AC0, "DELTAE"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "E", AC0
+R 1.0, AC0
/R 2.0, AC0
MOVR AC0, "W1"
XORD AC0, AC0
MOVR "DELTAE", AC0
+R 1.0, AC0
/R 2.0, AC0
MOVR AC0, "W2"
XORD AC0, AC0
MOVR 1.0, AC0
-R "E", AC0
/R 2.0, AC0
MOVR AC0, "W3"
XORD AC0, AC0
MOVR 1.0, AC0
-R "DELTAE", AC0
/R 2.0, AC0
MOVR AC0, "W4"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "W1", AC0
*R "W2", AC0
MOVR AC0, "B1"
*R "U1", AC0
MOVR AC0, "U11"

```

```

XORD AC0, AC0
MOVR "W1", AC0
*R "W4", AC0
MOVR AC0, "B2"
*R "U2", AC0
MOVR AC0, "U22"
XORD AC0, AC0
MOVR "W3", AC0
*R "W2", AC0
MOVR AC0, "B3"
*R "U3", AC0
MOVR AC0, "U33"
XORD AC0, AC0
MOVR "W3", AC0
*R "W4", AC0
MOVR AC0, "B4"
*R "U4", AC0
MOVR AC0, "U44"
XORD AC0, AC0
MOVR "B1", AC0
+R "B2", AC0
+R "B3", AC0
+R "B4", AC0
MOVR AC0, "TOPB"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "U11", AC0
+R "U22", AC0
+R "U33", AC0
+R "U44", AC0
MOVR AC0, "U"
/R "TOPB", AC0
MOVR AC0, "UU"
LD I0.0
AN T37
XORD AC0, AC0
MOVR "UU", AC0
*R "KD", AC0
MOVR AC0, "UD"
XORD AC0, AC0
MOVR "UU", AC0
*R "KI", AC0
+R "U_1", AC0
MOVR AC0, "UI"
XORD AC0, AC0
MOVR "UI", AC0
+R "UD", AC0
MOVR AC0, "UT"
LDR>= "UT", "UMAX"

```

```
MOVR "UMAX", "UT"  
LDR<= "UT", "UMIN"  
MOVR "UMIN", "UT"  
LD I0.0  
AN T37  
XORD AC0, AC0  
MOVR "UT", AC0  
TRUNC AC0, AC0  
XORD "E_2", "E_2"  
MOVR "E_1", "E_2"  
XORD "E_1", "E_1"  
MOVR "E1", "E_1"
```

```
XORD "U_1", "U_1"  
MOVR "UI", "U_1"  
MOVR "KDE", "KDE_1"  
MOVR "KI", "KI_1"  
MOVW AC0, AQW0  
MOVW AIW0, *VD1002  
+D +2, VD1002  
MOVR "MP", *VD200  
+D +4, VD200  
MOVW AC0, *VD3600  
+D +2, VD3600
```



EK C4

Aşağıda fonksiyon ayarlayıcısı yöntemi kullanılarak tasarlanan özayarlamalı bulanık PID tipi kontrolöre ait PLC programı verilmiştir.

// ANA PROGRAM

NETWORK 1

```
LD SM0.1
MOVB 200, SMB34
ATCH 0, 10
ENI
CALL 0
LD I0.0
TON T37, +200
LD I0.0
ED
MOVW +0, AQW0
LD T37
EU
MOVW +0, AQW0
```

// 0 NO' LU ALT PROGRAM

NETWORK 1

```
LD SM0.0
MOVR 0.0, "E_1"
MOVR 0.0, "U_1"
MOVR 1.0, "U1"
MOVR 0.0, "U2"
MOVR 0.0, "U3"
MOVR -1.0, "U4"
MOVR 12800.0, "KIS"
MOVR 32000.0, "KD"
MOVR 32000.0, "UMAX"
MOVR 0.0, "UMIN"
MOVW +16000, "REF"
MOVR 2.8125E-005, "KE"
MOVR 2.1875E-005, "KDES"
MOVR 0.28, "A1"
MOVR 0.45, "A2"
MOVR 0.16, "K1"
MOVR 3.4, "K2"
```

// KESME ALT PROGRAMI

NETWORK 1

```
LD I0.1
EU
MOVD &VB400, VD700
MOVD &VB1000, VD1300
LD I0.0
AN T37
MOVW AIW0, "Y"
XORD AC0, AC0
MOVW "REF", AC0
-I "Y", AC0
LDW>= AC0, +0
JMP 0
LDW<= AC0, +0
ORD 16#FFFF0000, AC0
LBL 0
DTR AC0, AC0
MOVR AC0, "E"
LDR>= "E", 0.0
XORD AC0, AC0
MOVR "E", AC0
/R 32000.0, AC0
XORD "EN", "EN"
MOVR AC0, "EN"
JMP 1
NOT
XORD "EN", "EN"
XORD AC0, AC0
MOVR "E", AC0
*R -1.0, AC0
/R 32000.0, AC0
MOVR AC0, "EN"
LBL 1
XORD AC0, AC0
MOVR 1.0, AC0
```

```

-R  "EN", AC0
*R  "K1", AC0
+R  "K2", AC0
MOVR AC0, "GFONK"
XORD AC0, AC0
MOVR "EN", AC0
*R  "A1", AC0
+R  "A2", AC0
MOVR AC0, "FFONK"
XORD AC0, AC0
MOVR "E", AC0
*R  "KE", AC0
MOVR AC0, "EE"
LDR>= "EE", 1.0
XORD "EE", "EE"
MOVR 1.0, "EE"
JMP 2
LDR<= "EE", -1.0
XORD "EE", "EE"
MOVR -1.0, "EE"
JMP 2
LBL 2
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "E", AC0
-R  "E_1", AC0
*R  "GFONK", AC0
*R  "KDES", AC0
MOVR AC0, "DELTAE"
LDR>= "DELTAE", 1.0
XORD "DELTAE", "DELTAE"
MOVR 1.0, "DELTAE"
JMP 3
LDR<= "DELTAE", -1.0
XORD "DELTAE", "DELTAE"
MOVR -1.0, "DELTAE"
LBL 3
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "EE", AC0
+R  1.0, AC0
/R  2.0, AC0
MOVR AC0, "W1"
XORD AC0, AC0
MOVR "DELTAE", AC0
+R  1.0, AC0
/R  2.0, AC0
MOVR AC0, "W2"

```

```

XORD AC0, AC0
MOVR 1.0, AC0
-R  "EE", AC0
/R  2.0, AC0
MOVR AC0, "W3"
XORD AC0, AC0
MOVR 1.0, AC0
-R  "DELTAE", AC0
/R  2.0, AC0
MOVR AC0, "W4"
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "W1", AC0
*R  "W2", AC0
MOVR AC0, "B1"
*R  "U1", AC0
MOVR AC0, "U11"
XORD AC0, AC0
MOVR "W1", AC0
*R  "W4", AC0
MOVR AC0, "B2"
*R  "U2", AC0
MOVR AC0, "U22"
XORD AC0, AC0
MOVR "W3", AC0
*R  "W2", AC0
MOVR AC0, "B3"
*R  "U3", AC0
MOVR AC0, "U33"
XORD AC0, AC0
MOVR "W3", AC0
*R  "W4", AC0
MOVR AC0, "B4"
*R  "U4", AC0
MOVR AC0, "U44"
XORD AC0, AC0
MOVR "B1", AC0
+R  "B2", AC0
+R  "B3", AC0
+R  "B4", AC0
MOVR AC0, "TOPB"
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "U11", AC0
+R  "U22", AC0
+R  "U33", AC0
+R  "U44", AC0
MOVR AC0, "UU"

```

```

/R  "TOPB", AC0
MOVR AC0, "U"
LD  I0.0
AN  T37
XORD AC0, AC0
XORD "UD", "UD"
MOVR "U", AC0
*R  "KD", AC0
MOVR AC0, "UD"
XORD "UI", "UI"
XORD AC0, AC0
MOVR "U", AC0
*R  "KIS", AC0
*R  "FFONK", AC0
+R  "U_1", AC0
MOVR AC0, "UI"
XORD "U_1", "U_1"
MOVR "UI", "U_1"
+R  "UD", AC0
XORD "UT", "UT"
MOVR AC0, "UT"
LDR>= "UT", "UMAX"
MOVR "UMAX", "UT"
LDR<= "UT", "UMIN"
MOVR "UMIN", "UT"
LD  I0.0
AN  T37
MOVR "UT", AC0
TRUNC AC0, AC0

```

```

MOVW AC0, AQW0
MOVW AIW0, *VD700
+D  +2, VD700
MOVW AC0, *VD1300
+D  +2, VD1300
XORD "E_1", "E_1"
MOVR "E", "E_1"
XORD "UU", "UU"
XORD "U", "U"
XORD "E", "E"
XORD "W1", "W1"
XORD "W2", "W2"
XORD "W3", "W3"
XORD "W4", "W4"
XORD "U11", "U11"
XORD "U22", "U22"
XORD "U33", "U33"
XORD "U44", "U44"
XORD "B1", "B1"
XORD "B2", "B2"
XORD "B3", "B3"
XORD "B4", "B4"
XORD "TOPB", "TOPB"
XORD "EE", "EE"
XORD "DELTA E", "DELTA E"
XORD "GFONK", "GFONK"
XORD "FFONK", "FFONK"
XORD "EN", "EN"

```

EK C5

Aşağıda bağıl hız gözleyicisi yöntemi kullanılarak tasarlanan özayarlamalı bulanık PID tipi kontrolöre ait PLC programı verilmiştir.

// ANA PROGRAM

NETWORK 1

```
LD SM0.1
MOVB 200, SMB34
ATCH 0, 10
ENI
CALL 0
LD I0.0
TON T37, +200
LD I0.0
ED
MOVW +0, AQW0
LD T37
EU
MOVW +0, AQW0
```

// 0 NO' LU ALTPROGRAM

NETWORK 1

```
LD SM0.0
MOVR 0.0, "E1"
MOVR 0.0, "E2"
MOVR 1.0, "KFD"
MOVR 6.7, "KF"
MOVR 0.0, "U_1"
MOVR 1.0, "U1"
MOVR 0.0, "U2"
MOVR 0.0, "U3"
MOVR -1.0, "U4"
MOVR 12800.0, "KI"
MOVR 32000.0, "KD"
MOVR 32000.0, "UMAX"
MOVR 0.0, "UMIN"
MOVW +16000, "REF"
MOVR 2.8125E-005, "KE"
MOVR 2.1875E-005, "KDE"
MOVR 2.1875E-005, "KDES"
MOVR 12800.0, "KIS"
```

// KESME ALT PROGRAMI

NETWORK 1

```
LD I0.1
EU
MOVD &VB400, VD700
MOVD &VB1000, VD1300
MOVD &VB1600, VD2100
MOVD &VB2600, VD3100
LD I0.0
AN T37
MOVW AIW0, "Y"
XORD AC0, AC0
MOVW "REF", AC0
-I "Y", AC0
LDW>= AC0, +0
JMP 0
LDW<= AC0, +0
ORD 16#FFFF0000, AC0
LBL 0
DTR AC0, AC0
MOVR AC0, "E"
LDR>= AC0, 0.0
XORD "EN", "EN"
/R 32000.0, AC0
MOVR AC0, "EN"
JMP 7
NOT
XORD AC0, AC0
XORD "EN", "EN"
MOVR "E", AC0
*R -1.0, AC0
/R 32000.0, AC0
MOVR AC0, "EN"
LBL 7
LD I0.0
AN T37
XORD "DE", "DE"
XORD "DE1", "DE1"
```

```

XORD AC0, AC0
XORD "DDE", "DDE"
MOVR "E", AC0
-R "E1", AC0
MOVR AC0, "DE"
XORD AC0, AC0
MOVR "E1", AC0
-R "E2", AC0
MOVR AC0, "DE1"
*R "DE", AC0
LDR<= AC0, 0.0
MOVR 0.0, "DE1"
JMP 4
NOT
LBL 4
XORD AC0, AC0
MOVR "DE", AC0
-R "DE1", AC0
MOVR AC0, "DDE"
LDR>= "DDE", 0.0
XORD "DDEN", "DDEN"
MOVR "DDE", "DDEN"
JMP 5
NOT
XORD AC0, AC0
XORD "DDEN", "DDEN"
MOVR "DDE", AC0
*R -1.0, AC0
MOVR AC0, "DDEN"
LBL 5
LDR>= "DE", 0.0
XORD "DEN", "DEN"
MOVR "DE", "DEN"
JMP 1
NOT
XORD "DEN", "DEN"
XORD AC0, AC0
MOVR "DE", AC0
*R -1.0, AC0
MOVR AC0, "DEN"
LBL 1
LDR>= "DE1", 0.0
XORD "DE1N", "DE1N"
MOVR "DE1", "DE1N"
JMP 2
NOT
XORD "DE1N", "DE1N"
XORD AC0, AC0
MOVR "DE1", AC0
*R -1.0, AC0

```

```

MOVR AC0, "DE1N"
LBL 2
LDR>= "DEN", "DE1N"
XORD "BOL", "BOL"
MOVR "DE", "BOL"
JMP 3
NOT
XORD "BOL", "BOL"
MOVR "DE1", "BOL"
LBL 3
LDR<= "DDEN", 400.0
XORD "RV", "RV"
MOVR -1.0, "RV"
JMP 6
NOT
XORD AC0, AC0
XORD "RV", "RV"
MOVR "DDE", AC0
/R "BOL", AC0
MOVR AC0, "RV"
LBL 6
LD I0.0
AN T37
LDR>= "EN", 0.0
AR<= "EN", 0.333
XORD AC0, AC0
XORD "K1", "K1"
MOVR "EN", AC0
*R -1.0, AC0
/R 0.333, AC0
+R 1.0, AC0
MOVR AC0, "K1"
JMP 11
NOT
MOVR 0.0, "K1"
LBL 11
LDR>= "EN", 0.0
AR<= "EN", 0.667
XORD AC0, AC0
XORD "K2", "K2"
MOVR "EN", AC0
/R 0.333, AC0
-R 1.0, AC0
JMP 19
NOT
MOVR 0.0, "K2"
JMP 12
LBL 19
LDR>= AC0, 0.0
*R -1.0, AC0

```

```

+R 1.0, AC0
MOVR AC0, "K2"
JMP 12
NOT
+R 1.0, AC0
MOVR AC0, "K2"
LBL 12
LDR>= "EN", 0.333
AR<= "EN", 1.0
XORD AC0, AC0
XORD "K3", "K3"
MOVR "EN", AC0
/R 0.333, AC0
-R 2.0, AC0
JMP 20
NOT
MOVR 0.0, "K3"
JMP 13
LBL 20
LDR>= AC0, 0.0
*R -1.0, AC0
+R 1.0, AC0
MOVR AC0, "K3"
JMP 13
NOT
+R 1.0, AC0
MOVR AC0, "K3"
LBL 13
LDR>= "EN", 0.667
AR<= "EN", 1.0
XORD AC0, AC0
XORD "K4", "K4"
MOVR "EN", AC0
/R 0.333, AC0
-R 3.0, AC0
JMP 21
NOT
MOVR 0.0, "K4"
JMP 14
LBL 21
LDR>= AC0, 0.0
*R -1.0, AC0
+R 1.0, AC0
MOVR AC0, "K4"
JMP 14
NOT
+R 1.0, AC0
MOVR AC0, "K4"
LBL 14
LDR>= "RV", -1.0

```

```

AR<= "RV", 0.0
XORD AC0, AC0
XORD "K5", "K5"
MOVR "RV", AC0
*R -1.0, AC0
MOVR AC0, "K5"
JMP 15
NOT
MOVR 0.0, "K5"
LBL 15
LDR>= "RV", -1.0
AR<= "RV", 1.0
XORD AC0, AC0
XORD "K6", "K6"
MOVR "RV", AC0
JMP 23
NOT
MOVR 0.0, "K6"
JMP 16
LBL 23
LDR>= AC0, 0.0
*R -1.0, AC0
+R 1.0, AC0
MOVR AC0, "K6"
JMP 16
NOT
+R 1.0, AC0
MOVR AC0, "K6"
LBL 16
LDR>= "RV", 0.0
AR<= "RV", 1.0
XORD AC0, AC0
XORD "K7", "K7"
MOVR "RV", "K7"
JMP 17
NOT
MOVR 0.0, "K7"
LBL 17
LD I0.0
AN T37
XORD "S1", "S1"
XORD "S2", "S2"
XORD "S3", "S3"
XORD "S4", "S4"
XORD "S5", "S5"
XORD "S6", "S6"
XORD "S7", "S7"
XORD "S8", "S8"
XORD "S9", "S9"
XORD "S10", "S10"

```

```

XORD "S11", "S11"
XORD "S12", "S12"
XORD "D1", "D1"
XORD "D2", "D2"
XORD "D3", "D3"
XORD "D4", "D4"
XORD "D5", "D5"
XORD "D6", "D6"
XORD "D7", "D7"
XORD "D8", "D8"
XORD "D9", "D9"
XORD "D10", "D10"
XORD "D11", "D11"
XORD "D12", "D12"
XORD AC0, AC0
MOVR "K1", AC0
*R "K5", AC0
MOVR AC0, "S1"
*R 0.667, AC0
MOVR AC0, "D1"
XORD AC0, AC0
MOVR "K1", AC0
*R "K6", AC0
MOVR AC0, "S2"
*R 0.667, AC0
MOVR AC0, "D2"
XORD AC0, AC0
MOVR "K1", AC0
*R "K7", AC0
MOVR AC0, "S3"
*R 1.0, AC0
MOVR AC0, "D3"
XORD AC0, AC0
MOVR "K2", AC0
*R "K5", AC0
MOVR AC0, "S4"
*R 0.333, AC0
MOVR AC0, "D4"
XORD AC0, AC0
MOVR "K2", AC0
*R "K6", AC0
MOVR AC0, "S5"
*R 0.667, AC0
MOVR AC0, "D5"
XORD AC0, AC0
MOVR "K2", AC0
*R "K7", AC0
MOVR AC0, "S6"
*R 1.0, AC0
MOVR AC0, "D6"

```

```

XORD AC0, AC0
MOVR "K3", AC0
*R "K5", AC0
MOVR AC0, "S7"
*R 0.0, AC0
MOVR AC0, "D7"
XORD AC0, AC0
MOVR "K3", AC0
*R "K6", AC0
MOVR AC0, "S8"
*R 0.333, AC0
MOVR AC0, "D8"
XORD AC0, AC0
MOVR "K3", AC0
*R "K7", AC0
MOVR AC0, "S9"
*R 0.667, AC0
MOVR AC0, "D9"
XORD AC0, AC0
MOVR "K4", AC0
*R "K5", AC0
MOVR AC0, "S10"
*R 0.0, AC0
MOVR AC0, "D10"
XORD AC0, AC0
MOVR "K4", AC0
*R "K6", AC0
MOVR AC0, "S11"
*R 0.0, AC0
MOVR AC0, "D11"
XORD AC0, AC0
MOVR "K4", AC0
*R "K7", AC0
MOVR AC0, "S12"
*R 0.333, AC0
MOVR AC0, "D12"
LD I0.0
AN T37
XORD "TOPS", "TOPS"
XORD "TOPD", "TOPD"
XORD AC0, AC0
XORD "GAMA", "GAMA"
MOVR "S1", AC0
+R "S2", AC0
+R "S3", AC0
+R "S4", AC0
+R "S5", AC0
+R "S6", AC0
+R "S7", AC0
+R "S8", AC0

```

```

+R  "S9", AC0
+R  "S10", AC0
+R  "S11", AC0
+R  "S12", AC0
MOVR AC0, "TOPS"
XORD AC0, AC0
MOVR "D1", AC0
+R  "D2", AC0
+R  "D3", AC0
+R  "D4", AC0
+R  "D5", AC0
+R  "D6", AC0
+R  "D7", AC0
+R  "D8", AC0
+R  "D9", AC0
+R  "D10", AC0
+R  "D11", AC0
+R  "D12", AC0
MOVR AC0, "TOPD"
/R  AC0, "TOPS"
MOVR AC0, "GAMA"
*R  "KF", AC0
XORD "KC", "KC"
MOVR AC0, "KC"
XORD "KC1", "KC1"
*R  "KFD", AC0
MOVR AC0, "KC1"
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "E", AC0
*R  "KE", AC0
MOVR AC0, "EE"
LDR>= "EE", 1.0
XORD "EE", "EE"
MOVR 1.0, "EE"
JMP 40
LDR<= "EE", -1.0
XORD "EE", "EE"
MOVR -1.0, "EE"
LBL 40
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "DE", AC0
*R  "KDES", AC0
*R  "KC1", AC0
MOVR AC0, "DELTAE"
LDR>= "DELTAE", 1.0
XORD "DELTAE", "DELTAE"

```

```

MOVR 1.0, "DELTAE"
JMP 41
LDR<= "DELTAE", -1.0
XORD "DELTAE", "DELTAE"
MOVR -1.0, "DELTAE"
LBL 41
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "EE", AC0
+R  1.0, AC0
/R  2.0, AC0
MOVR AC0, "W1"
XORD AC0, AC0
MOVR "DELTAE", AC0
+R  1.0, AC0
/R  2.0, AC0
MOVR AC0, "W2"
XORD AC0, AC0
MOVR 1.0, AC0
-R  "EE", AC0
/R  2.0, AC0
MOVR AC0, "W3"
XORD AC0, AC0
MOVR 1.0, AC0
-R  "DELTAE", AC0
/R  2.0, AC0
MOVR AC0, "W4"
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "W1", AC0
*R  "W2", AC0
MOVR AC0, "B1"
*R  "U1", AC0
MOVR AC0, "U11"
XORD AC0, AC0
MOVR "W1", AC0
*R  "W4", AC0
MOVR AC0, "B2"
*R  "U2", AC0
MOVR AC0, "U22"
XORD AC0, AC0
MOVR "W3", AC0
*R  "W2", AC0
MOVR AC0, "B3"
*R  "U3", AC0
MOVR AC0, "U33"
XORD AC0, AC0
MOVR "W3", AC0

```

```

*R  "W4", AC0
MOVR AC0, "B4"
*R  "U4", AC0
MOVR AC0, "U44"
XORD AC0, AC0
MOVR "B1", AC0
+R  "B2", AC0
+R  "B3", AC0
+R  "B4", AC0
MOVR AC0, "TOPB"
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "U11", AC0
+R  "U22", AC0
+R  "U33", AC0
+R  "U44", AC0
MOVR AC0, "UU"
/R  "TOPB", AC0
MOVR AC0, "U"
LD  I0.0
AN  T37
XORD AC0, AC0
MOVR "U", AC0
*R  "KD", AC0
MOVR AC0, "UD"
XORD AC0, AC0
MOVR "U", AC0
*R  "KIS", AC0
/R  "KC", AC0
+R  "U_1", AC0
MOVR AC0, "UI"
+R  "UD", AC0
MOVR AC0, "UT"
LDR>= "UT", "UMAX"
MOVR "UMAX", "UT"
LDR<= "UT", "UMIN"
MOVR "UMIN", "UT"
LD  I0.0

```

```

AN  T37
MOVR "UT", AC0
TRUNC AC0, AC0
MOVW AC0, AQW0
MOVW AC0, *VD1300
+D  +2, VD1300
XORD "E2", "E2"
MOVR "E1", "E2"
XORD "E1", "E1"
MOVR "E", "E1"
XORD "U_1", "U_1"
MOVR "UI", "U_1"
XORD "E", "E"
XORD AC0, AC0
MOVW AIW0, *VD700
+D  +2, VD700
MOVR "RV", *VD2100
+D  +4, VD2100
MOVR "GAMA", *VD3100
+D  +4, VD3100
XORD "EE", "EE"
XORD "DELTA E", "DELTA E"
XORD "W1", "W1"
XORD "W2", "W2"
XORD "W3", "W3"
XORD "W4", "W4"
XORD "B1", "B1"
XORD "B2", "B2"
XORD "B3", "B3"
XORD "B4", "B4"
XORD "U11", "U11"
XORD "U22", "U22"
XORD "U33", "U33"
XORD "U44", "U44"
XORD "U", "U"
XORD "UD", "UD"
XORD "UI", "UI"
XORD "UT", "UT"
XORD "UU", "U"

```

ÖZGEÇMİŞ

Onur Karasakal, 05. 04. 1977 tarihinde Kocaeli' de doğdu. İlk ve orta öğrenimini Kocaeli' de tamamladıktan sonra 1992 yılında Deniz Lisesi' ne girdi. 1999 yılında İletişim Mühendisliği bölümünde lisans öğrenimini tamamlayıp, Deniz Harp Okulu'ndan deniz subayı olarak mezun oldu. İki yıl çeşitli bölgelerde görev yaptıktan sonra 2001 yılında İTÜ Elektrik / Elektronik Fakültesi, Kontrol ve Otomasyon Mühendisliği Programında lisansüstü öğrenimine başladı.

