

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PID KONTROLÖRÜN KARINCA KOLONİSİ /
GENETİK ALGORİTMA TABANLI OPTİMİZASYONU
VE GUNT RT 532 BASINÇ PROSESİNİN KONTROLÜ

Muhammet ÜNAL

YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR VE KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Hasan ERDAL
Yrd. Doç. Dr. Vedat TOPUZ

İSTANBUL, 2008

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PID KONTROLÖRÜN KARINCA KOLONİSİ /
GENETİK ALGORİTMA TABANLI OPTİMİZASYONU
VE GUNT RT 532 BASINÇ PROSESİNİN KONTROLÜ

Muhammet ÜNAL
(141100420050031)

YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR VE KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Hasan ERDAL
Yrd. Doç. Dr. Vedat TOPUZ

İSTANBUL, 2008

ÖNSÖZ VE TEŞEKKÜR

Teknolojinin gelişmesiyle birlikte, doğada var olan birçok sistem yeniden keşfedilmekte ve gerçekleştirilen çalışmalara esin kaynağı olmaktadır. Canlılar üzerinde var olan öğrenme, akıl yürütme ve karar verme gibi özelliklerden esinlenerek geliştirilen yapay zeka yöntemlerinin, kontrol uygulamalarında kullanılması oldukça yaygınlaşmış ve sunduğu çözümlerden dolayı bir tercih nedeni olmuştur.

Yapılan bu tez çalışmasında öncelikle Gunt RT 532 Basınç Proses Sisteminin Yapay Sinir Ağı (YSA) modeli oluşturulmuştur. Pnömatik oransal valf ve basınç tankından oluşan bu sistemin basınç kontrolü yörünge takipli olarak klasik PID kontrolör ile gerçekleştirilmiştir. PID kontrolörün K_p , K_i ve K_d katsayıları esnek programlama yöntemi olarak bilinen genetik ve karınca kolonisi algoritması ile optimize edilmiştir. Yapılan bu çalışmanın modelleme, optimizasyon ve kontrol üzerine çalışma yapacak kişiler için bir kaynak niteliğinde faydalı olacağına inanıyorum.

Bana, tez konusunda ve yapılan çalışmaların her aşamasında sabırlı ve değerli vaktini ayırarak yardımcı olan tez danışmanlarım Yrd. Doç. Dr. Hasan Erdal ve Yrd. Doç. Dr. Vedat TOPUZ 'a, sürekli desteğini esirgemeyen Türker Tekin Ergüzel' e, uzun süre bana tahammül eden aileme ve arkadaşlarıma teşekkür ederim.

İÇİNDEKİLER

	<u>SAYFA NO</u>
ÖNSÖZ VE TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
ÖZET.....	ix
ABSTRACT	x
SEMBOLLER	xi
KISALTMALAR.....	xiii
ŞEKİLLER	xiv
TABLolar	xvii
BÖLÜM I	1
GİRİŞ ve AMAÇ	1
BÖLÜM II.....	3
SİSTEMİN YAPISI.....	3
II.1. LİTERATÜR İNCELEMESİ.....	3
II.2. GUNT RT 532 BASINÇ PROSES KONTROL SİSTEMİ.....	7
II.3. SİSTEMİN GENEL YAPISI	8
II.3.1. Kontrolör.....	8
II.3.2. Grafik Çizici	9
II.3.3. Basınç Tankı	9
II.3.4. Pnömatik Oransal Valf (I/P)	9
II.3.5. Manometre	11
II.3.6. Basınç Regülâtör Ünitesi	11
II.3.7. Tahliye Vanası	11
II.3.8. Basınç Transdüseri (P/I)	12
II.4. SİSTEM ÜZERİNDE YAPILAN DEĞİŞİKLİKLER.....	13

II.4.1. Sistemde Kullanılan Filtre	14
II.4.2. Sistemin Basınç Eğrileri	17
II.4.3. Sistemin Çalışma Prensibi	18
II.5. ÇÖZÜM İÇİN YAKLAŞIMLAR	19
II.6. PID KONTROLÖR.....	20
II.7. YÖRÜNGE TAKİBİ.....	21
II.7.1. Sistemde Kullanılan Yörünge Denklemi	22
BÖLÜM III	24
ESNEK PROGRAMLAMA(SC)	24
III.1. YAPAY SİNİR AĞLARI(YSA) ile SİSTEM TANIMLAMA	25
III.1.1. Giriş.....	25
III.1.2. Biyolojik Sinir Ağları.....	26
III.1.3. Yapay Sinir Ağı Hücresi(Perceptron)	28
III.1.3.1. Aktivasyon Fonksiyonu	29
III.1.4. Yapay Sinir Ağları Modelleri.....	31
III.1.4.1. Mimari Yapılarına Göre YSA.....	31
III.1.4.2. Öğrenme Yapılarına Göre YSA.....	32
III.1.5. Hatanın Geri Yayılımı Öğrenme Algoritması.....	34
III.1.6. YSA ile Sistem Modelleme.....	36
III.1.6.1. İleri Modelleme (Seri-paralel modelleme).....	36
III.2. GENETİK ALGORİTMA (GA)	39
III.2.1. Basit Genetik Algoritma(SGA).....	39
III.2.2. Genetik Algoritma Operatörleri	41
III.2.2.1. Kodlama	41
III.2.2.2. İlk Popülasyonun Oluşturulması.....	42
III.2.2.3. Popülasyon Büyüklüğü	43
III.2.2.4. Üreme.....	43

III.2.2.5. Seçim Yöntemleri	43
III.2.2.6. Çaprazlama	45
III.2.2.7. Mutasyon(Değişim)	46
III.2.3. Kısıtlı Optimizasyon Problemlerinin Çözümü	47
III.2.3.1. Düzenleyici Algoritma	47
III.2.3.2. GENOCOP	48
III.2.3.3. Ceza fonksiyonu	48
III.2.4. GA ile Optimizasyon	48
III.3. KARINCA KOLONİSİ OPTİMİZASYONU(KKA)	50
III.3.1. Gerçek Karıncalar	50
III.3.2. Yapay Karıncalar	52
III.3.3. Karınca Kolonisi Algoritması	53
III.3.3.1. Feromen Buharlaşması	54
BÖLÜM IV	55
TEZ ÇALIŞMALARI	55
IV.1. SİSTEMİN YSA ile MODELLENMESİ	56
IV.1.1. Kullanılan YSA Modeli	56
IV.1.2. Eğitim Verisinin Toplanması	56
IV.1.3. Verinin Normalleştirilmesi	58
IV.1.4. YSA Mimarisinin Seçilmesi	58
IV.1.5. YSA 'nın Eğitimi	62
IV.2. GA ile PID KONTROLÖR TASARIMI	66
IV.2.1. Sistem İçin Tasarlanan Kontrolörün Genel Yapısı	66
IV.2.2. PID Parametrelerin Genetik Olarak Kodlanması	67
IV.2.3. PID Parametrelerin GA Tarafından Öğrenilmesi	68
IV.2.4. Sistem Çıkışının GA Uygunluk Değerine Dönüştürülmesi	69
IV.2.4.1. Maksimum Aşım miktarı (ym)	69

IV.2.4.2. Gecikme Zamanı (td)	69
IV.2.4.3. Yükselme Zamanı (tr)	69
IV.2.4.4. Yerleşme Zamanı (ts).....	70
IV.2.4.5. Kalıcı Hal Hatası (ess)	70
IV.2.5. Kullanılan Genetik Operatörler	70
IV.3. GENETİK-PID KONTROLÖRÜN PERFORMANSI.....	72
IV.3.1. Basamak Girişlere Sistemin Davranışı.....	72
IV.3.2. Basamak Girişler Arasında Geçişe Sistemin Davranışı	74
IV.3.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı.....	76
IV.3.4. Genetik-PID Kontrolörün Yörünge Takip Cevapları	80
IV.4. KKA ile PID KONTROLÖR TASARIMI	86
IV.4.1. Kullanılan KKA Parametreleri	89
IV.5. KARINCA-PID KONTROLÖRÜN PERFORMANSI	92
IV.5.1. Basamak Girişlere Sistemin Davranışı.....	92
IV.5.2. Basamak Girişler Arasında Geçişe Sistemin Davranışı	94
IV.5.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı.....	96
IV.5.4. Karınca-PID Kontrolörün Yörünge Takip Cevapları	99
IV.6. ZIEGLER- NICHOLS(ZN) YÖNTEMİYLE PID KATSAYILARININ BULUNMASI	105
IV.6. ZN-PID KONTROLÖRÜN PERFORMANSI.....	107
IV.6.1. Basamak Girişlere Sistemin Davranışı.....	107
IV.6.2. Basamak Girişler Arasında Geçişe Sistemin Davranışı	109
IV.6.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı.....	111
IV.6.3. ZN-PID Kontrolörün Yörünge Takip Cevapları	113
BÖLÜM – V	117
SONUÇLAR	117
V.1. TARTIŞMA VE DEĞERLENDİRME	120

REFERANSLAR.....	123
ÖZGEÇMİŞ.....	126

ÖZET

PID KONTROLÖRÜN KARINCA KOLONİSİ / GENETİK ALGORİTMA TABANLI OPTİMİZASYONU ve GUNT RT 532 BASINÇ PROSESİNİN KONTROLÜ

Bu tez, Marmara Üniversitesi Elektronik-Bilgisayar Eğitimi Bölümü Sayısal Kontrol Sistemleri Laboratuvarında bulunan Gunt RT 532 Basınç Proses Kontrol sisteminin denetiminde kullanılan PID kontrolör katsayılarının optimal değerlerinin, genetik algoritma (GA) ve karınca kolonisi algoritması(KKA) ile bulunması ve sistemin bu katsayılar ile gerçek zamanlı kontrolü üzerinedir.

Kontrol edilecek prosesin dinamik modeli yapay sinir ağı(YSA) kullanılarak elde edilmiştir. Dinamik modelinin oluşturulması sırasında, sisteme belli giriş değerleri verilmiş ve bu değerlere karşılık sistem çıkışları elde edilerek giriş-çıkış veri seti oluşturulmuştur. Bu veri seti kullanılarak proses için farklı katman ve nöron sayılı modeller oluşturulmuştur. Bu modeller farklı YSA performans fonksiyonlarına göre değerlendirilmiş ve hata değeri en küçük model, prosesin dinamik modeli olarak seçilmiştir.

Seçilen model kullanılarak PID kontrolör KKA, GA ve Ziegler-Nichols (ZN) ile optimize edilmiştir. KKA ve GA teknikleri ile elde edilen sonuçlar, klasik teknik olan ZN ile elde edilen aşım, yükselme zamanı, oturma zamanı kriterlerine ve yörünge takibindeki karekök ortalama (Root Mean Square-RMS) hatasına göre karşılaştırılmıştır. GA ve KKA' nın performanslarının ZN tekniğine göre daha iyi olduğu gözlenmiştir.

ABSTRACT

OPTIMIZATION OF PID CONTROLLER USING ANT COLONY / GENETIC ALGORITHMS AND CONTROL OF THE GUNT RT 532 PRESSURE PROCESS

This thesis describes, a real time control algorithm, using genetic algorithm (GA) and ant colony optimization (ACO) algorithm for optimizing PID controller parameters developed for Gunt RT 532 Pressure Process Control System in the Digital Control Systems Laboratory of Technical Education Faculty at Marmara University.

The dynamic model of the process to be controlled was obtained using Artificial Neural Network (ANN). In development of the model, the system was run with different input and output values and, these were taken as the input-output data set. Using this data set, models with varying number of layers and neurons were constructed for the process. The model was evaluated on the basis of their performance functions. The model with minimum error was chosen as the dynamic model of the process.

Using the chosen model, the parameters of PID controller were optimized with ACO, GA and Ziegler-Nichols (ZN) techniques. The performances of these three techniques were compared with each other using the criteria of overshoot, rise time, settling time and root mean square (RMS) error of the trajectory. It was observed that the performances of GA and ACO are better than that of ZN technique.

January, 2008

Muhammet ÜNAL

SEMBOLLER

K_p	: Oransal kontrolör katsayısı
K_i	: İntegral kontrolör katsayısı
K_d	: Türev kontrolör katsayısı
u_k	: Sayısal kontrol işareti
py	: Yörünge değeri
t	: Zaman
t_s	: Oturma zamanı
t_o	: Gecikme zamanı
P_o	: Başlangıç noktası
P_s	: İstenilen referans noktası
x_{min}	: Gerçel sayının minimum değeri
x_{max}	: Gerçel sayının maksimum değeri
t_{max}	: Yörünge süresi
Y	: Sistem çıkışı
R	: İstenen yörünge çıkışı
a	: Yapay sinir ağının(YSA) çıkış çektörü
W	: YSA 'nın ağırlık matrisi
IW	: YSA' nın giriş katmanının ağırlık matrisi
LW	: YSA' nın gizli katmanının ağırlık matrisi
f	: Yapay sinir ağının aktivasyon fonksiyonu
m	: YSA' nın katman sayısı
s	: YSA' nın her bir katmanının duyarlık değeri
α	: YSA' nın öğrenme oranı
b	: YSA' nın bias(eşik) değeri
D	: YSA' nın delay(Zaman gecikmesi)
e	: Hata vektörü
t	: Hedef vektörü
p	: Giriş vektörü
k	: Giriş sayısı

$\hat{y}$	: Tahmin edilen YSA çıkışı
N	: Genetik algortmada (GA) popölasyon büyüklüğü
p_i	: GA' da bireyin sečilme olasılığı
P_c	: GA' da bir bireyin çaprazlama olasılığı
p_k	: GA' da bir bireyin yaşama olasılığı
r_u	: GA' da rulet tekerleğı yönteminde bir bireyin sečilme şansı
q	: GA' da turnuva büyüklüğü
Φ	: GA' da uygunluk değeri
τ	: Karınca Kolonisi Algoritmasında(KKA) feromen miktarı
λ	: KKA' da buharlaşma katsayısı
c	: KKA' da RMS hatası
Δ	: KKA' da her tur sonunda feromen miktarını tutan matris

KISALTMALAR

PID	: Oransal Integral Türevsel
YSA	: Yapay Sinir Ağı
GA	: Genetik Algoritma
KKA	: Karınca Kolonisi Optimizasyonu
SGA	: Basit Genetik Algoritma
ZN	: Ziegler-Nichols
SC	: Esnek Programlama
GMVC	: Genelleştirilmiş Minimum Değişken Kontrolü
RMS	: Karekök Ortalama (Root Mean Square)
MSE	: Karesel Hata Ortalaması (Mean Square Error)
MSEREG	: Düzenleştirilmiş Karesel Hata Ortalaması (Mean Squared Error with Regularization)
SSE	: Karesel Hataların Toplamı (Sum Square Error)
ADC	: Analog Sayısal Çevirici
DAC	: Sayısal Analog Çevirici
FL	: Bulanık Mantık
MLP	: Çok Katmanlı Perceptron
SL	: Eğitici Öğrenme
UL	: Eğitici Öğrenme
L	: Gecikme Zamanı
T	: Zaman Sabiti

ŞEKİLLER

SAYFA NO

Şekil II. 1 Sistemin genel görünüşü.....	8
Şekil II. 2 Basınç tankı	9
Şekil II. 3 Pnömatik oransal valf	10
Şekil II. 4 Pnömatik oransal valf' e ait gerilim - açıklık grafiği.....	10
Şekil II. 5 Analog manometre	11
Şekil II. 6 Basınç Regülâtörü	11
Şekil II. 7 Cerabar PMC 133 basınç transdüseri	12
Şekil II. 8 Basınç transdüserine ait gerilim-basınç grafiği	12
Şekil II. 9 Tasarlanan kontrol sisteminin genel yapısı	13
Şekil II. 10 Basınç transduserinden alınan gürültülü veri	14
Şekil II. 11 Toplanan verinin frekans spektrumunda gösterimi	15
Şekil II. 12 Filtre karakteristiği	15
Şekil II. 13 Toplanan ilk veri ve filtrelenen verinin karşılaştırılması.....	16
Şekil II. 14 Ham veri ile filtreli veri arasındaki hata.....	16
Şekil II. 15 Sistemin 0-10V basamak girişe karşılık dolum eğrileri	17
Şekil II. 16 Kontrol edilen sistemin blok diyagramı	18
Şekil II. 17 PID kontrolörün blok şeması.....	20
Şekil II. 18 (A) noktadan noktaya (B) yol takibi (C) yörünge takibi	22
Şekil II. 19 Çalışmada kullanılan yörünge grafiği.....	23
Şekil III. 1 Esnek programlamayı oluşturan programlama teknikleri.....	24
Şekil III. 2 Doğal sinir hücresi	27
Şekil III. 3 Sinir sisteminde bilgi akışı.....	27
Şekil III. 4 Biyolojik ve yapay sinir hücresi.....	28
Şekil III. 5 Doğrusal aktivasyon fonksiyonu.....	29
Şekil III. 6 Sigmoid fonksiyon	30
Şekil III. 7 Hiperbolik tanjant fonksiyonu	30
Şekil III. 8 İleri beslemeli YSA modeli.....	31
Şekil III. 9 Geri beslemeli YSA yapısı.....	32
Şekil III. 10 YSA' nın eğiticili öğrenme ile eğitilmesi.....	33
Şekil III. 11 İleri beslemeli çok katmanlı sinir ağı.....	35
Şekil III. 12 YSA ile modellemede kullanılan temel ileri tip blok şema	37
Şekil III. 13 Seri – paralel ileri tip modelleme blok şeması.....	38
Şekil III. 14 Basit genetik algoritmanın akış diyagramı.....	40

Şekil III. 15 İkili kodlama	42
Şekil III. 16 Rulet Tekerleği seçim yöntemi	44
Şekil III. 17 Çaprazlama operatörü	46
Şekil III. 18 Aritmetik çaprazlama	46
Şekil III. 19 Mutasyon operatörü	47
Şekil III. 20 Karıncaların izlediği yol	51
Şekil III. 21 Karıncaların bir engelle karşılaşması	51
Şekil III. 22 Engelle karşılaşan karıncaların seçimi	51
Şekil III. 23 Karıncaların kısa yolu bulmaları	52
Şekil IV. 1 YSA eğitimi için uygulanan basamak girişler ve sistemin cevabı	57
Şekil IV. 2 YSA eğitimi için uygulanan rastgele girişler ve sistemin cevabı	57
Şekil IV. 3 Kullanılan YSA modelin blok diyagramı	60
Şekil IV. 4 Gizli katmanındaki nöronların bağlantı şekli	61
Şekil IV. 5 Eğitim boyunca MSE hatasının değişim eğrisi	62
Şekil IV. 6 Hedef veriler ile YSA Model çıkışının regresyon eğrisi	63
Şekil IV. 7 Sistem ile YSA Modeli çıkış grafiği	64
Şekil IV. 8 Açık çevrimde sistem ile YSA çıkışının karşılaştırılması	65
Şekil IV. 9 Çevrim dışı çalışmada kullanılan sistemin kapalı genel yapısı	66
Şekil IV. 10 GA ile PID kontrolörün optimizasyonu	67
Şekil IV. 11 PID Kontrolör için geliştirilen kromozom yapısı	68
Şekil IV. 12 Tipik bir birim basamak girişinde sistem cevabı	69
Şekil IV. 13 1bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları	72
Şekil IV. 14 2bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları	73
Şekil IV. 15 3bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları	73
Şekil IV. 16 4bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları	74
Şekil IV. 17 Basamak geçişler sırasında gerçek sistemin davranışı	75
Şekil IV. 18 %25 bozucu uygulanan sistemin cevabı	77
Şekil IV. 19 %40 bozucu uygulanan sistemin cevabı	78
Şekil IV. 20 %50 bozucu uygulanan sistemin cevabı	79
Şekil IV. 21 Eğimi 6 olan referans yörüngesinin sistem tarafından takibi	80
Şekil IV. 22 Eğimi 10 olan referans yörüngesinin sistem tarafından takibi	81
Şekil IV. 23 Eğimi 15 olan referans yörüngesinin sistem tarafından takibi	82
Şekil IV. 24 Eğimi 20 olan referans yörüngesinin sistem tarafından takibi	83
Şekil IV. 25 Eğimi 30 olan referans yörüngesinin sistem tarafından takibi	84
Şekil IV. 26 Yörüngeler arası geçiş sistemin cevabı	85
Şekil IV. 27 KKA ile PID Kontrolör optimizasyonu	86
Şekil IV. 28 PID katsayıları için yapay karıncanın izleyebileceği yollar	87

Şekil IV. 29 KKA ile PID kontrolör tasarımı için kullanılan algoritma.....	90
Şekil IV. 30 1bar referans girişe ait kontrol, YSA model ve gerçek sistem çıkışları	92
Şekil IV. 31 2bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları	93
Şekil IV. 32 3bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları	93
Şekil IV. 33 4bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları	94
Şekil IV. 34 Basamak geçişler sırasında sistemin davranışı.....	95
Şekil IV. 35 %25 bozucu uygulanan sistemin cevabı.....	96
Şekil IV. 36 %40 bozucu uygulanan sistemin cevabı.....	97
Şekil IV. 37 %50 bozucu uygulanan sistemin cevabı.....	98
Şekil IV. 38 Eğimi 6 olan referans yörüngenin sistem tarafından takibi.....	99
Şekil IV. 39 Eğimi 10 olan referans yörüngenin sistem tarafından takibi	100
Şekil IV. 40 Eğimi 15 olan referans yörüngenin sistem tarafından takibi	101
Şekil IV. 41 Eğimi 20 olan referans yörüngenin sistem tarafından takibi	102
Şekil IV. 42 Eğimi 30 olan referans yörüngenin sistem tarafından takibi	103
Şekil IV. 43 Yörüngeler arası geçişe sistemin cevabı	104
Şekil IV. 44 Sabit genlikli basamak girişe sistemin cevabı.....	105
Şekil IV. 45 Basamak girişe sistemin cevabı	106
Şekil IV. 46 1bar referans girişe ait kontrol ve gerçek sistem çıkışları.....	107
Şekil IV. 47 2bar referans girişe ait kontrol ve gerçek sistem çıkışları.....	108
Şekil IV. 48 3bar referans girişe ait kontrol ve gerçek sistem çıkışları.....	108
Şekil IV. 49 4bar referans girişe ait kontrol ve gerçek sistem çıkışları.....	109
Şekil IV. 50 Basamak geçişler sırasında gerçek sistemin davranışı.....	110
Şekil IV. 51 %15 bozucu uygulanan sistemin cevabı.....	111
Şekil IV. 52 %25 bozucu uygulanan sistemin cevabı.....	112
Şekil IV. 53 Eğimi 6 olan referans yörüngenin sistem tarafından takibi.....	113
Şekil IV. 54 Eğimi 10 olan referans yörüngenin sistem tarafından takibi	114
Şekil IV. 55 Eğimi 15 olan referans yörüngenin sistem tarafından takibi	114
Şekil IV. 56 Eğimi 20 olan referans yörüngenin sistem tarafından takibi	115
Şekil IV. 57 Eğimi 30 olan referans yörüngenin sistem tarafından takibi	115
Şekil IV. 58 Yörüngeler arası geçişe sistemin cevabı	116
Şekil V.1 Farklı değerlerde basamak girişlere sistemin cevabı	118
Şekil V.2 Farklı yörünge değerleri için sistem cevapları.....	119

TABLolar

SAYFA NO

Tablo III. 1 Bireylerin uygunluk deęerlerinin bilgisayar ortamına evrilmesi.....	45
Tablo IV. 1 YSA mimarisi iin yapılan deney sonuları	59
Tablo IV. 2 Ziegler-Nichols 1.yöntem katsayı tablosu	105
Tablo V. 1 Elde edilen PID katsayıları.....	117
Tablo V. 2 PID kontrolörlerin kontrol karakteristikleri	118
Tablo V.3 Yörünge deęerleri iin RMS hataları	119

BÖLÜM I

GİRİŞ ve AMAÇ

Yapılan bu çalışma, bir basınç kontrol sisteminin yapay sinir ağı (YSA) ile modellenerek, PID kontrolör ile yörünge takipli kontrolü ve bu amaçla tasarlanan PID kontrolör katsayılarının genetik algoritma (GA), karınca kolonisi algoritması (KKA) ve Ziegler-Nichols (ZN) ile bulunması ve bu yöntemlerin performanslarının karşılaştırılması üzerinedir.

Basınç kontrol sisteminde, havanın sıkıştırılabilirliği, hava sıcaklığı ile basıncın değişmesi, ortamın değişken sıcaklıkta olması ve bozucu etkiler gibi durumlardan sistem, doğrusal olmayan bir davranış göstermektedir. Bu sebeplerden dolayı basınç sistemin modeli çıkartılırken matematiksel model yerine, YSA modeli kullanılmıştır. Modelleme sırasında, basınç sistemine farklı değerlerde girişler verilerek sistemin davranışı kaydedilmiş ve bu giriş-çıkış verileri YSA' ya eğitim verisi olarak uygulanıp, basınç sisteminin dinamik modeli elde edilmiştir.

Elde edilen dinamik YSA modeli üzerinden, PID kontrolörün katsayılarının optimal değerleri, GA ve KKA ile bulunmuştur. Referans ile çıkış arasındaki toplam mutlak hatanın minimize edilmesi sistemin amaç/optimallik kriteri olarak seçilmiştir. Bulunan PID katsayıları ile sistem gerçek zamanlı kontrol edilmiştir. Sistemin gerçek zamanlı davranışını test etmek için sisteme farklı basamak ve yörünge giriş değerleri verilmiştir. Elde edilen sistem davranışları birbirleri ve ZN yöntemi ile ayarlanan PID kontrolör ile karşılaştırılmıştır.

Bölüm II de literatür taraması yapılarak basınç sistemleri üzerinde yapılan klasik ve Esnek Programlama(SC) uygulamaları verilmiştir. Basınç tankını oluşturan donanım ve yazılım yapıları, PID kontrolör ve sisteme verilen referans yörünge fonksiyonu anlatılmıştır.

Bölüm III 'de esnek programlama tekniklerinden olan YSA, GA ve KKA tanımları ve teorik bilgi verilmiştir.

Bölüm IV 'de YSA ile basınç tankının dinamik modeli çıkarılmış, GA, KKA ve ZN ile YSA modeli üzerinden çevrimdışı çalışmada PID kontrolör için optimum katsayılar bulunmuştur. Tasarlanan Genetik-PID, Karınca-PID ve ZN-PID kontrolörün performansları klasik ve yörünge takipli deneyler ile ortaya konulmuştur.

Bölüm V' de elde edilen deneysel sonuçlar ışığı altında kontrolörlerin performansları değerlendirilip, öneri ve temenniler de bulunulmuştur.

BÖLÜM II

SİSTEMİN YAPISI

Bu bölümde öncelikle literatür incelemesi yapılacaktır. Literatür incelemesinde, basınç sistemlerinin kontrol uygulamaları, Genetik-PID ve Karınca-PID sistemler incelenecektir. İkinci olarak basınç prosesi detaylı olarak ele alınacaktır. Üçüncü olarak da sistemin kontrolünde kullanılan PID kontrolör ve referans olarak verilen yörünge fonksiyonu anlatılacaktır.

II.1. LİTERATÜR İNCELEMESİ

Literatürde, basınç sistemleri ile ilgili yapılmış kontrol uygulamaları ve kontrol aşamasında kullanılan YSA, GA, Genetik-PID Kontrolör, KKA ve Karınca-PID kontrolörlerle ilgili pek çok araştırma ve çalışma mevcuttur. Bir sistemin kontrolünü gerçekleştirmek için deneysel yöntemler bulunsa da özellikle simülasyon tabanlı çalışmalarda sistemin matematiksel modeline ihtiyaç duyulmaktadır. Dinamik davranışa ait parametreleri bilinen doğrusal yapıdaki sistemlerin matematiksel modellerinin elde edilmesi kolaydır. Doğrusal olmayan yapıdaki sistemlerin matematik modellerinin elde edilmesi ise oldukça güç ve zaman alıcıdır. Böyle durumlarda sistemin giriş-çıkış davranışı kullanılarak elde edilmesine dayalı yöntemler kullanılmaktadır. Bu yöntemlerden en çok tercih edilen yöntem ise YSA ile modellemedir.

Ancak bazı sistemlerde, doğrusal olmayan sistem değişkenlerinden ve/veya sistemin karmaşık yapısından dolayı sistemin giriş/çıkış davranışının YSA ile modellenmesi tercih edilen yöntemlerin arasındadır.

H.Mok ve arkadaşları zaman gecikmesine sahip, doğrusal olmayan davranışlı bir basınç kontrol sistemin sayısal PI kontrolör ile kontrol etmişlerdir. Sistemin modeli zaman gecikmesine sahip birinci dereceden bir transfer fonksiyonu olarak çıkartılmış ve bu model üzerinden kontrolör, ZN yöntemi kullanılarak açık çevrim frekans cevabına göre optimize edilmiştir[1].

R.Namba ve arkadaşları, PID kontrolörün katsayılarının düzgün şekilde bulunmasının, modelleme hatasından ve sistemin zamanla değişim göstermesinden doğan problemlerden dolayı zor olacağı için güçlendirilmiş kararlık teorisine(robust stability teory) göre PID kontrolörün parametrelerini belirleyerek hava basınç ünitesinin kontrolünü gerçekleştirmişlerdir[2].

C.Guo ve arkadaşları ısıtma, havalandırma ve iklimlendirme (HVAC) sisteminde hava destek ünitesinin kontrolünü PI kontrolör destekli YSA ile gerçekleştirmişlerdir. Sistem çalışırken oluşan gürültüleri elimine etmek için çevrimiçi eğitilen YSA kullanılmışlardır. Bu sistemde, PI kontrolör sayesinde araştırmacılar, kontrol sırasında oluşan hata sıfır iken dahi sisteme kontrol sinyali gönderebilmiş ve böylece sistem istenilen referans seviyesinde tutulabilmişlerdir[3].

W.Jian ve arkadaşının yaptığı çalışmada iklimlendirme sistemi için hava basınç ünitesinin kontrolünü gerçekleştirmişlerdir. Sistemin PID kontrolör ile çalışması sırasında, sistemin çıkışı verilen tolerans içinde düzgün bir şekilde devam etmektedir. Ancak sistemin zamanla değişen parametreleri ile sistem, doğrusal olmayan davranış göstermeye başlamıştır. Bu durumda PID kontrolör, sistemi kontrol etmekte zayıf kalmış ve bu zayıflığı gidermek için araştırmacılar doğrusal olmayan kontrolörler geliştirilmişlerdir. Bu amaçla sistemin ikinci dereceden ve zaman gecikmesi içeren bir modeli çıkartılmıştır. Bu model üzerinden PID ve bulanık kontrolör araştırmacılar tarafından, sinirsel-bulanık kontrolör ise en küçük kareler ortalamasına göre optimize edilmiş ve sistem hazırlanan üç kontrolör ile kontrol edilmiştir. Elde edilen sonuçlar içinde sinirsel-bulanık kontrolörün daha başarılı olduğu gösterilmiştir[4].

M. Ünal ve arkadaşları, transfer fonksiyonu bilinen ISAC (Intelligent Soft Arm Control) adı verilen robot kolunun simülasyon ortamında, referans olarak verilen bir yörüngeyi takip etmesi için GA ile PID kontrolör tasarlamışlardır. Robot koluna verilen yörünge GA tarafından optimize edilen PID kontrolör ile başarılı bir şekilde takip edebileceğini göstermişlerdir[5].

M. Salami ve arkadaşı verilen transfer fonksiyonu üzerinden PID kontrolörün kazanç değerini ve sistemin toplam hata değerinin küçülmesini sağlayan PID katsayılarının optimizasyonunu gerçekleştirmişlerdir. GA için gerekli parametreler kodlanırken ikili kodlama yöntemini kullanmışlardır[6].

A.H. Jones ve arkadaşı yaptıkları çalışmada kontrol edilecek sistem olarak laboratuvarın sıcaklık kontrolünü seçmişler ve ilk olarak sistemin modelini

oluşturmuşlardır. İkinci aşamada ise elde edilen model üzerinden çevrimdışı çalışmada sistemin kontrolü için gerekli olan PID kontrolörün katsayılarını GA ile toplam hatanın minimize edilmesi prensibine göre bulmuşlardır. Son olarak Genetik algoritma ile bulunan PID kontrolörün katsayılarının performansını Åstrom-Relay Auto Tuning metodu ile karşılaştırmışlardır ve GA'nın bulduğu katsayıların daha iyi sonuçlar verdiğini ortaya koymuşlardır[7].

Ö. Gündoğdu yaptığı çalışmada, bir dişli kutusuyla tahrik edilen ataletli bir kütlenin kontrolünde kullanılacak PID kontrolörün optimum ayarlanmasına ait bir çalışma yapmıştır. Yapılan çalışmada, sistem çıkışının kararlı haldeki değeri ile referans değerinin RMS(Root Mean Square) hatasını elde ederek, GA ile optimum kontrolör parametrelerini bulunmuştur. Çalışma, popülasyonun oluşturulacağı kontrolör parametre aralığını, ZN yönteminin önerdiği değerlerin kullanımını önermekte ve GA ile ZN'tan elde edilen sonuçlar sonuçları karşılaştırılmıştır. Genetik algoritmalar kullanıldığında daha iyi sonuçlar elde edildiği gösterilmiştir[8].

B.Dandıl ve arkadaşları yaptıkları çalışmada, doğrusal ve doğrusal olmayan çeşitli sistemler için GA ile model referans PID kontrolör tasarımını gerçekleştirmişlerdir. Önerilen denetim yönteminde, doğrusal ve doğrusal olmayan birçok sistemi, arzu edilen denetim kriterlerini (en büyük aşma, yükselme zamanı ve yerleşme zamanı gibi) sağlayacak şekilde denetleyen en uygun PID parametrelerini bulmuşlardır[9].

H.Alli ve arkadaşları yaptıkları çalışmada iki dinamik sistem için modellemiş ve sistemlerin kontrolü için PID kontrol katsayılarını değişik performans indekslerine göre GA kullanarak optimize etmişlerdir. Ayrıca bu performans indekslerinden en uygununun, zamansal mutlak hatanın integrali (the Integral of the Time Absolute Error, ITAE) ile yapılan hesaplamanın olduğunu göstermişlerdir[10].

Y.Mitsukura ve arkadaşları yaptıkları çalışmada, bilinmeyen veya zamanla değişen sistem parametrelerinin optimize edildiği, PID kontrolör optimizasyonunu tasarlamışlardır. Çalışmada, PID kontrolör ve genelleştirilmiş minimum değişen kontrol(Generalized Minimum Variance Control) kurallarına dayanan optimizasyon sistemi gerçekleştirilmiştir. Kullanıcı tarafından değişen ve GMVC kriterlerini içeren parametrelerin GA'nın yinelemeli araması ile bulunduğu gösterilmiştir. Optimizasyonda uygunluk değeri hesaplanırken en küçük kareler yöntemi kullanılmıştır. Aynı zamanda GA'nın arama uzayının, kullanıcı bilgisi ile küçültülmesinin optimizasyonun arama süresini kısalttığını göstermişlerdir[11].

H.Duan ve arkadaşlarının yörünge takibini gerçekleştirmek için yaptıkları çalışmada, KKA tarafından bulunan PID kontrolörün katsayıları ile uçuş simülatörü çalıştırılmış ve verilen referans ile çıkış arasındaki fark alınarak PID kontrolörün optimizasyonu yapılmıştır[12].

Y.Hsiao ve arkadaşları yaptıkları çalışmada verilen transfer fonksiyonu üzerinden bilinen matematiksel teknikler ile PID kontrolörün katsayılarının optimizasyonunu gerçekleştirmişler ve KKA ile bulunan PID katsayılarının performansını karşılaştırmışlardır. KKA ile optimizasyon sırasında bulunan PID katsayılarının performansını hesaplamak için maksimum aşım, oturma zamanı, yükselme zamanı ve referans ile sistem çıkışı arasındaki toplam mutlak hata değerlerine bakılmıştır. Öncelikli olarak toplam mutlak hata değerine bakılmıştır. Aynı zamanda geçici cevaptaki davranışı diğer üç performans kriteri ile kontrol altına almışlardır[13].

T. Ergüzel ve arkadaşının yaptığı çalışmada, KKA ile bulanık kontrolün aksiyon değerleri ve PID kontrolörün katsayı değerleri bulunmuştur. Kontrolörler optimize edilirken verilen referans ile sistem çıkışının RMS(Root Mean Square) hatasına bakılmış ve RMS hatası minimize edilmeye çalışılmıştır[14].

H.Varol ve arkadaşının yaptığı çalışmada, KKA ile PID kontrolör optimize edilirken verilen transfer fonksiyonu üzerinden toplam mutlak hata ve toplam karesel hata ile hesaplanan maliyet fonksiyonu tasarlanmıştır. KKA ile optimize edilen PID kontrolör ile daha kısa oturma süresine sahip, çok az veya hiç aşım olmayan sonuçlar elde edilmiştir. Elde edilen sonuçlar ZN, Internal Model Control (IMC) and Iterative Feedback Tuning (IFT) metotları ile karşılaştırılarak performansı ölçülmüştür[15].

Yapılan çalışmada, yukarıda belirtilen çalışmalar ışığında baktığımızda bazı farklılıklar gözükür. Bunlar:

Proses (basınç tankı) kontrolü için PID kontrolörün katsayılarının optimizasyonu YSA modeli üzerinden GA ve KKA tarafından bulunmuştur. Gerçek zamanlı çalışmada ise, oluşturulan bu PID kontrolörün performansı izlenmiştir.

PID kontrolörün parametrelerinin GA ve KKA tarafından bulunması sırasında, verilen referans ile PID kontrolörlü modelin çıkışı arasında oluşan hatanın RMS değerinin minimum olacak şekilde bulmasını sağlayan yöntem incelenmiştir. Bu şekilde tasarlanan PID kontrolörün, farklı giriş seviyelerinde istenilen performans kriterinde uygun çalışması sağlanmaya çalışılmıştır.

II.2. GUNT RT 532 BASINÇ PROSES KONTROL SİSTEMİ

Bu bölümde proje çalışmasında kullanılmış olan Gunt RT 532 Basınç Proses Kontrol sisteminin genel yapısı incelenecektir. Ayrıca sisteme bu tez çalışması kapsamında yapılan ilavelerle oluşturulmuş yeni sistemin yapısı, çalışma prensibi ve sistemin farklı girişlerde gösterdiği davranışları incelenecektir.

RT 532 Basınç Proses Kontrol Sistemi, basınç kontrol kullanarak, proses kontrol ünitelerinin kapsamlı bir şekilde eğitim ve öğretimini sağlayan bir endüstriyel proses kontrol ünitesidir. Yapısının ve kullanımının kolaylığı ile sistem, mesleki eğitim veren okullarda ve üniversitelerde yapılan deneyler için çok kullanışlıdır. Bu nedenle günümüzde bu sistem eğitim ve deneysel amaçlı kullanılmaktadır. Bu deney seti ile yapılan bazı deneyler;

- Kontrol döngüsü davranışları,
- ON-OFF Kontrol,
- P Kontrol,
- PI Kontrol,
- PD Kontrol,
- PID Kontrol,
- Basamak girişe verilen cevapların kaydedilmesi,
- Bozucu değişkenlerin sisteme etkisinin tespiti;
- Sistemin kontrolünde optimizasyonu sağlamak

gibi aşamaları kapsar.

II.3. SİSTEMİN GENEL YAPISI

RT 532 Basınç Proses Kontrol Sisteminin yapısını incelemek için sistem sekiz bölüme ayrılmıştır. Bu bölümler Şekil II.1 'de ayrıntılı olarak gösterilmektedir.



Şekil II. 1 Sistemin genel görünüşü

Bu bölümler;

1. Kontrolör
2. İki kanallı Çizici
3. Basınç Regülâtör Ünitesi
4. Pnömatrik Oransal Valf (I/P)
5. Basınç Transdüseri (P/I)
6. Manometre
7. Tahliye Vanası
8. Basınç Tankı

II.3.1. Kontrolör

Sistem üzerinde üretici firma tarafından konulmuş, mikroişlemci tabanlı bir Honeywell marka UDC 5000 kontrolör vardır. Bütün ölçümler sayısaldir, sayısal olarak işlenir ve analog sinyale çevrilerek aktuatör elemana gönderilir. Kontrolörün

giriş ve çıkış değerleri 4-20mA arasındadır. Kontrolör parametreleri, kontrol paneli üzerindeki menüden basmalı butonlar yardımı seçilerek değiştirilebilir. UDC 5000 ile seçilen kontrolör tipine göre (PID, P, PI, PD, ON-OFF) sistem kontrol edilebilir.

Yapılan çalışmada, PC tabanlı kontrolör kullanıldığından dolayı bu kontrolör devre dışı bırakılmıştır.

II.3.2. Grafik Çizici

Kontrol değişkenlerini kaydetmek için kullanılır. İki adet analog giriş kanalı bulunur. Bunlar kontrol ve sistem çıkış sinyalidir. Yapılan çalışmada grafik işlemleri bilgisayar tabanlı yapıldığı için bu çizici kullanılmamıştır.

II.3.3. Basınç Tankı

Basınç tankı 10 lt hacme ve maksimum 16 bar basınca sahiptir. Kontrol edilen değişken bu tanktaki havanın basıncıdır. Amacımız bu tanka giren hava miktarı ile tank içindeki hava basıncı kontrol etmektir. Şekil II.2’de basınç tankı görülmektedir.



Şekil II. 2 Basınç tankı

II.3.4. Pnömatik Oransal Valf (I/P)

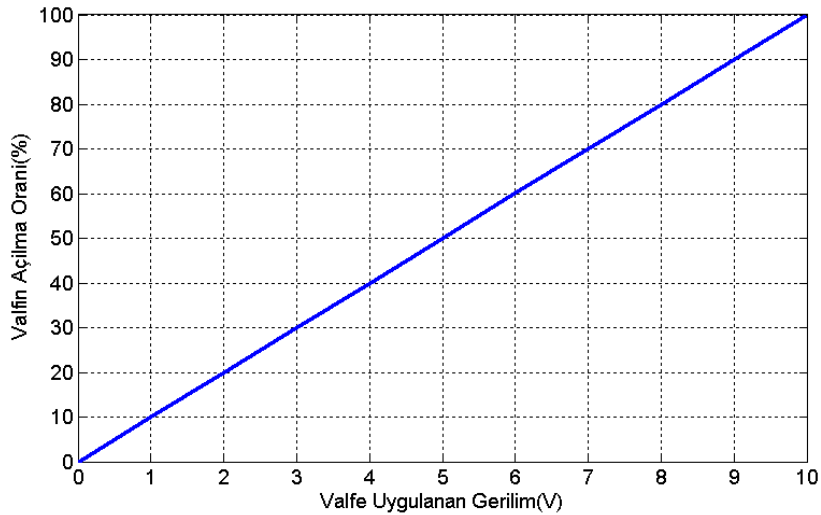
Kullanmış olduğumuz prosesde Samson firmasının üretmiş olduğu tip 241-7 pnömatik oransal valf kullanılmaktadır. Bu pnömatik valf, 2 bar basınç değerinde bir hava ile beslenerek, 4-20 mA’lık bir kontrol sinyali ile kontrol edilmektedir.

Kontrol devresinde aktuatör olarak görev yapar. Tanka giren basınçlı hava, bu pnömatik valf ile ayarlanır. Şekil II.3’de gösterilen valfin kumanda devresi elektrikli(4-20mA), güç devresi ise pnömatiktir.



Şekil II. 3 Pnömatik oransal valf

Pnömatik valfin sistemdeki görevi; uygulanan kontrol sinyaline göre oransal açılıp kapanarak tanka giren havanın debisini ayarlamaktır. Pnömatik valf 4-20mA arası lineer/doğrusal olarak çalışmaktadır. Sistemin kontrolünde kullanılan veri toplama kartı gerilim çıkışlı olduğu için veri toplama kartından gönderilen gerilimi akımı dönüştürmek için gerilim-akım dönüştürücü modül(Knick DC Isolation Amplifier) kullanılmıştır. Bu modül ile veri toplam kartından çıkan 0-10V'luk gerilim, 4-20mA'lık akıma dönüştürülerek pnömatik oransal valfe uygulanmaktadır. Valfin 0–10V arası çalışması Şekil II.4'de gösterilmektedir. Şekil II.4'den anlaşılacağı gibi pnömatik oransal valf 0 volttan sonraki gerilimlerde açılmaya başlamakta ve 10 voltta maksimum açıklığa ulaşmaktadır.



Şekil II. 4 Pnömatik oransal valf'e ait gerilim - açıklık grafiği

II.3.5. Manometre

Tank içindeki hava basıncının analog olarak görüntülenmesini sağlar. Şekil II.5’de görüldüğü gibi manometrenin ölçme aralığı 0-10 bar ve çapı 63mm’ dir.



Şekil II. 5 Analog manometre

II.3.6. Basınç Regülâtör Ünitesi

Şekil II.6’da gösterildiği gibi biri oransal pnömatik valfe diğeri sistem girişine bağlı iki adet basınç regülâtörü bulunmaktadır. Kompresörden gelen basınçlı havadaki titreşimleri ve fazla basıncı, istenen basınç değerinde düzenler. Sistemin beslemesi için basınç değeri 6 bar, valf için 2 bardır.



Şekil II. 6 Basınç Regülâtörü

II.3.7. Tahliye Vanası

Hava tankın çıkışında bulunur ve sisteme bozucu etki ithalini sağlar. Yapılan çalışmada tankın girişinde kullanılan pnömatik valften bir tanede tankın çıkışında kullanılmıştır. Tahliye vanası maksimum açılarak tahliye işlemi, çıkıştaki pnömatik valfe 6V’luk gerilim uygulanarak %60 oranında sağlanmaktadır.

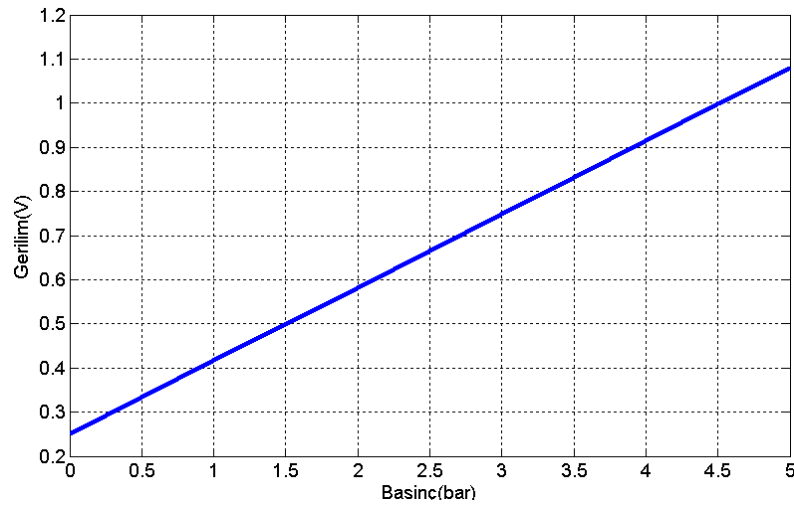
II.3.8. Basınç Transdüseri (P/I)

Endress+Hauser Firmasına ait transdüser Şekil II.7’de gösterilmektedir. Transdüser, basınç tankından aldığı basınç bilgisini 4–20 mA akım değerine çevirir.



Şekil II. 7 Cerabar PMC 133 basınç transdüseri

Şekil II.8’de bu transdüserin 64 Ω ’ luk bir direnç üzerinde oluşturduğu gerilim değerleri 0–5 bar için gösterilmiştir. Grafikten de anlaşılağı gibi tanktaki basınç değeri 0bar iken yaklaşık 0.25V’ luk gerilim değeri okunmakta ve hava basıncının artmasıyla basınç transdüserinden gelen gerilim doğrusal olarak artmaktadır.



Şekil II. 8 Basınç transdüserine ait gerilim-basınç grafiğı

Basınç transdüserinden okunan gerilim, basınç değerine dönüştürölmektedir. Böylece basınç transdüserinden okunan gerilim değeri basınç bilgisine çevrilip, sistemden geri besleme olarak alınmaktadır.

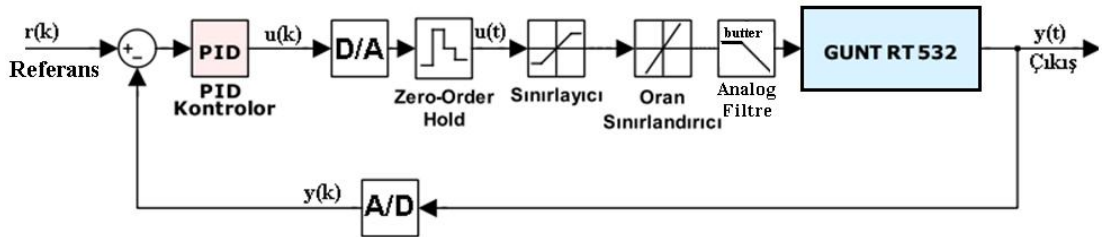
II.4. SİSTEM ÜZERİNDE YAPILAN DEĞİŞİKLİKLER

Sistem bilgisayarla kontrol edileceğinden dolayı sistem üzerinde bazı değişiklikler yapılması gerekmiştir. Bu değişiklikler sistemin temel çalışmasını değiştirmemekte, sadece kontrolörün çalışmasını düzenlemektedir. Sistemde yapılan en önemli değişiklik sistemde bulunan UDC 5000 kontrolörün devre dışı bırakılmasıdır.

Basınç kontrol ünitesi için gerçekleştirilen kontrol sisteminin genel yapısı Şekil-II.10'da görülmektedir. Sistemin temel elemanı, Pentium IV 1.6GHz işlemcili bir bilgisayardır ve kontrol işlemi Matlab programı ile oluşturulmuş yazılımdır. Bilgisayar içerisinde dijital/analog, analog/ dijital dönüştürme işlemi yapan veri toplama kartı (Advantech PCI1711) bulunmaktadır.

Sistemin basınç bilgisi basınç transdüserinden akım olarak alınmaktadır. ADC(Analog Digital Converter) kartına girişin gerilim olması gerektiğinden dolayı basınç transdüserinden alınan akım, 64Ω 'luk bir direnç üzerinden gerilime dönüştürülmüştür. Ayrıca pnömatisal oransal kontrol valfi akım kontrollü olduğu için DAC(Digital Analog Converter) tan gelen gerilim, Knick marka gerilim- akım dönüştürücü modül ile akım bilgisine dönüştürülerek pnömatisal oransal valfin kontrolü sağlanmıştır.

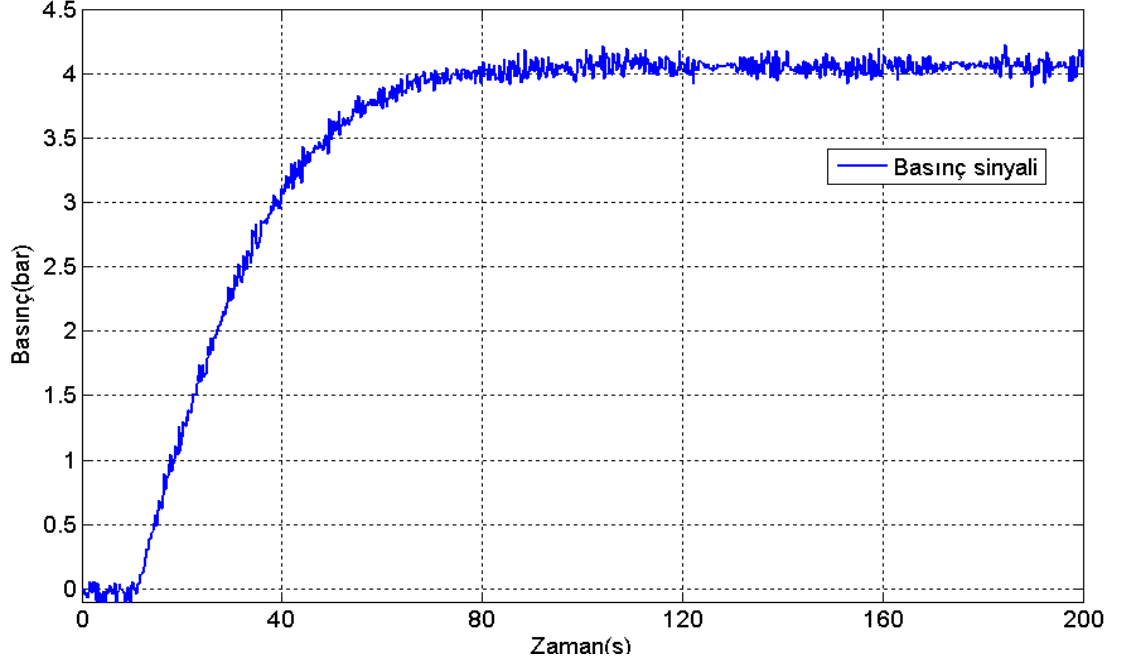
Sistemin güvenli çalışması için sistemin kontrolünde çeşitli eklentiler yapılmıştır. Şekil II.14'de gösterildiği gibi, PID kontrolörün çıkışı 10V ile sınırlandırılmış ve sistemin kontrolü sırasında pnömatisal valfin çok hızlı değişim göstermemesi için oran sınırlayıcı kullanılarak valfin hareketi yavaşlatılmıştır.



Şekil II. 9 Tasarlanan kontrol sisteminin genel yapısı

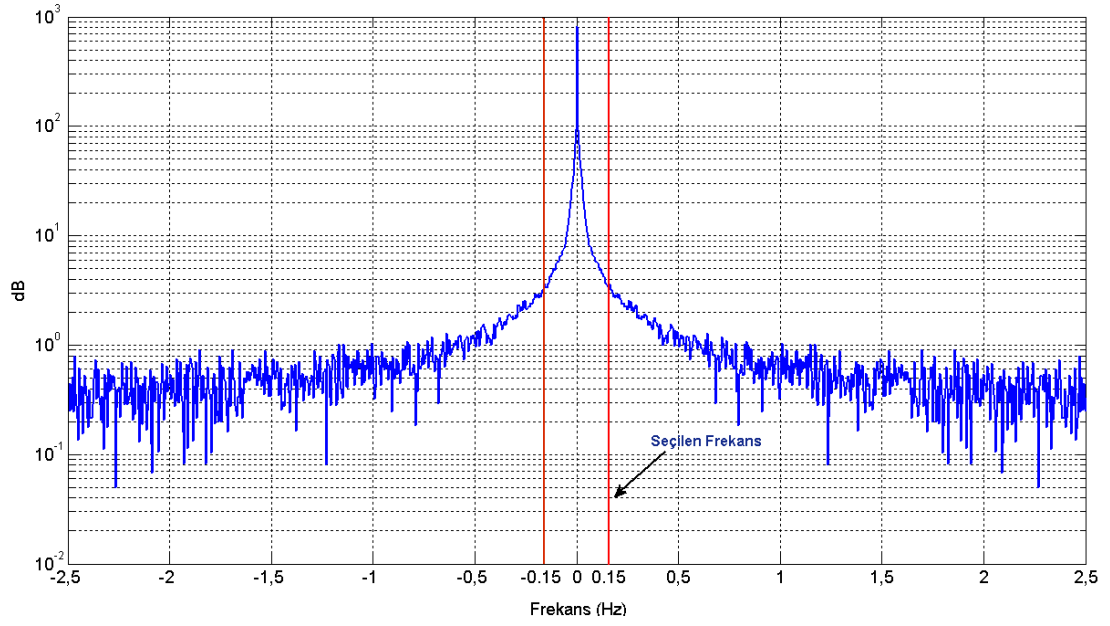
II.4.1. Sistemde Kullanılan Filtre

Basınç transduserinden veri alınırken, oluşan gürültülerin bastırılması için filtre kullanılmıştır. Basınç transduserinden alınan DC gerilim değeri, basınca çevrilip Şekil II.10’ da gösterilmiştir. Şekil II.10’ dan anlaşılabacağı gibi basınç transduserinden alınan veri DC gerilim olmasına rağmen çok titreşimli bir sinyaldir.



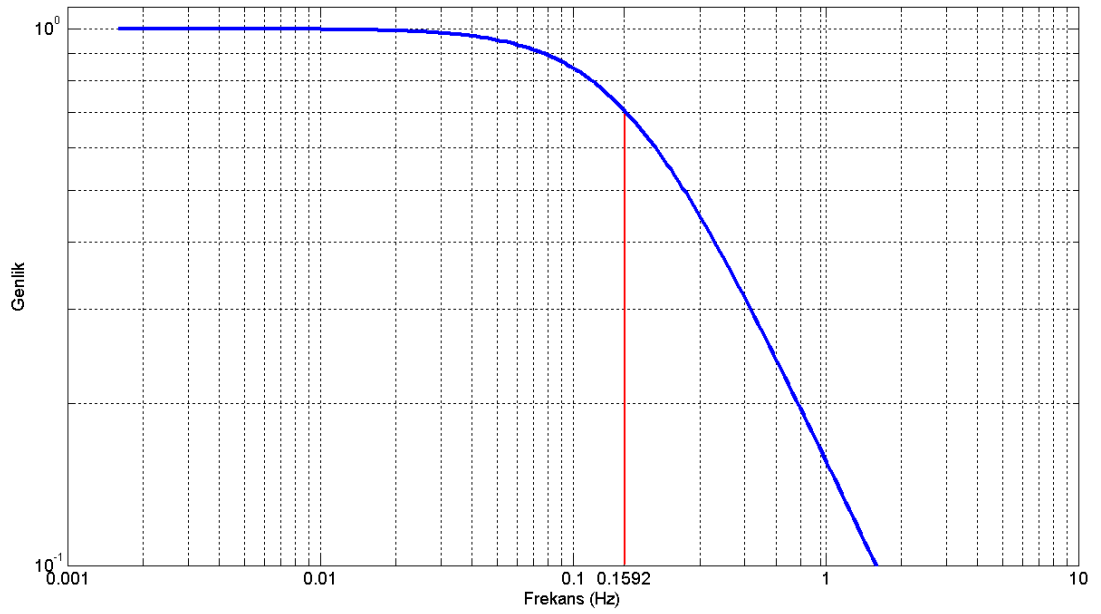
Şekil II. 10 Basınç transduserinden alınan gürültülü veri

Alınan verilerdeki titreşimleri bastırmak için öncelikle toplanan veri, frekans spektrumunda incelenmiştir. Basınç verisinin frekans spektrumundaki görünümü Şekil II.11’de gösterilmiştir. Şekil II.11’ de gösterilen frekans filtre için köşe frekansıdır ve köşe frekansı 0.1592Hz olarak seçilmiştir. Diğer frekanslar ise sistem için bozucu olarak kabul edilip, filtrelenen frekanslardır.



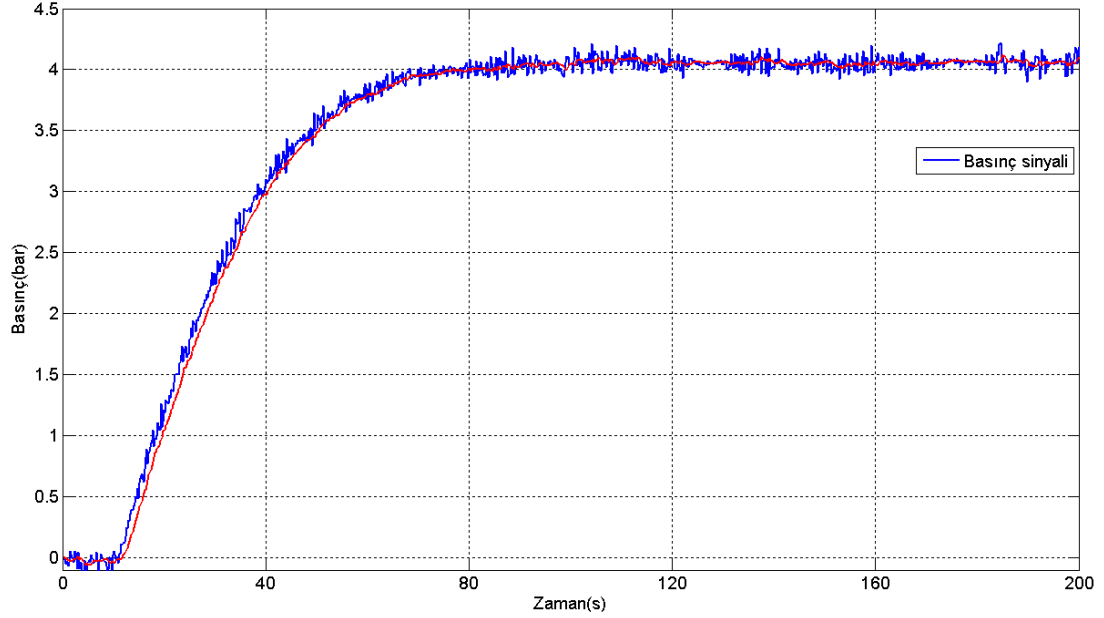
Şekil II. 11 Toplanan verinin frekans spektrumunda gösterimi

Sistem için 0.1592Hz’lik frekansı sinyalleri geçirip diğer frekansları geçirmeyen birinci dereceden, köşe frekansı 0.1592Hz olan butterworth tipinde bir analog filtre tasarlanmış ve tasarlanan filtrenin karakteristiği Şekil II.12’ de gösterilmiştir.



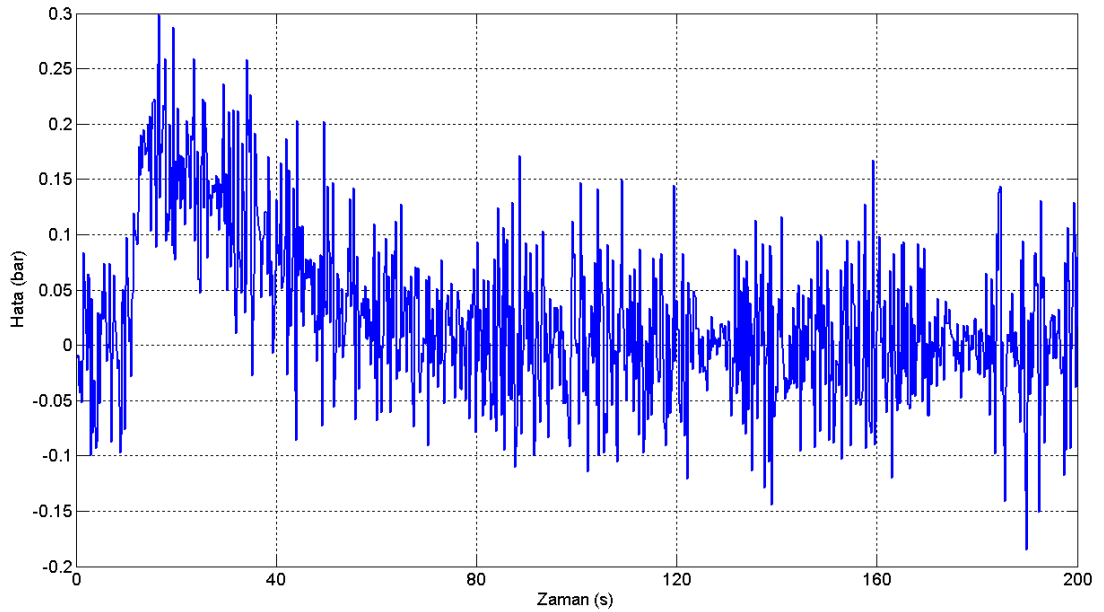
Şekil II. 12 Filtre karakteristiği

Basınç transduserinden alınan verilere, tasarlanan filtre uygulandığı zaman Şekil II.13’ de gösterilen çıkışlar elde edilmektedir. Şekil II.13’den anlaşılabacağı gibi toplanan verilerdeki gürültüler kırılmıştır. Böylece sistemin modellenmesi ve kontrolü sırasında basınç transduserinden daha gürültüsüz veriler alınmaktadır.



Şekil II. 13 Toplanan ilk veri ve filtrelenen verinin karşılaştırılması

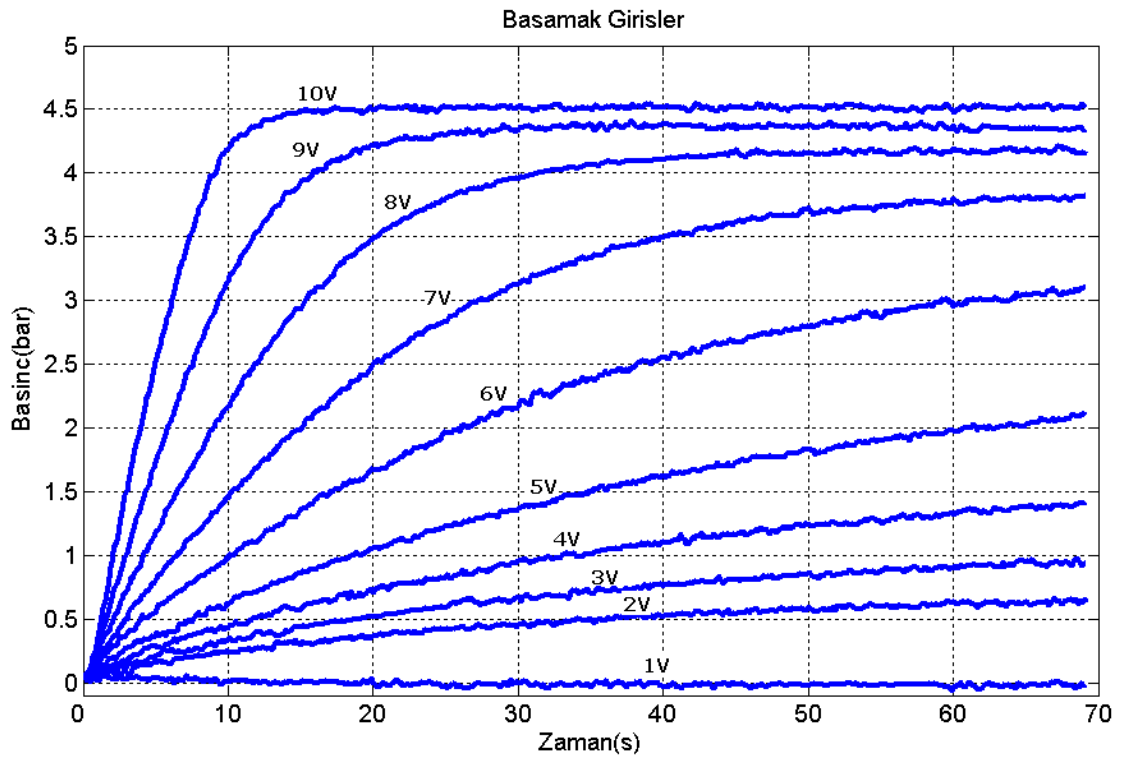
Ham veri ile filtreli veri arasında oluşan hataların grafiği Şekil II.14’de gösterilmektedir. Şekil II.14’den anlaşılabacağı gibi tasarlanan filtre ile toplanan verideki gürültüler temizlenmiştir.



Şekil II. 14 Ham veri ile filtreli veri arasındaki hata

II.4.2. Sistemin Basınç Eğrileri

GUNT RT 532 basınç kontrol ünitesinin PID kontrolü tasarımının gerçekleştirilmesi için öncelikle sistemin davranışının iyi tanınması gerekmektedir. Bu sebepten dolayı sistem üzerinde bulunan pnömatik kontrol valfine çeşitli gerilim sinyalleri uygulanarak, tanktaki basınç değişimleri incelenmiştir. Bu işlemin sonucunda sistemin açık çevrim cevabı elde edilmiştir. Elde edilen verilerden, sistem değişkenleri olarak kontrol girişi ve basınçtaki değişim ele alınmıştır. Pnömatik oransal valfin çalışma aralığı 4-20mA'dır. Ancak kullanılan akım gerilim dönüştürücü ile 0-10V çalışma aralığı elde edilmiştir. Çeşitli kontrol sinyalleri için sistemin filtreli davranışı Şekil II.15' de görülmektedir.

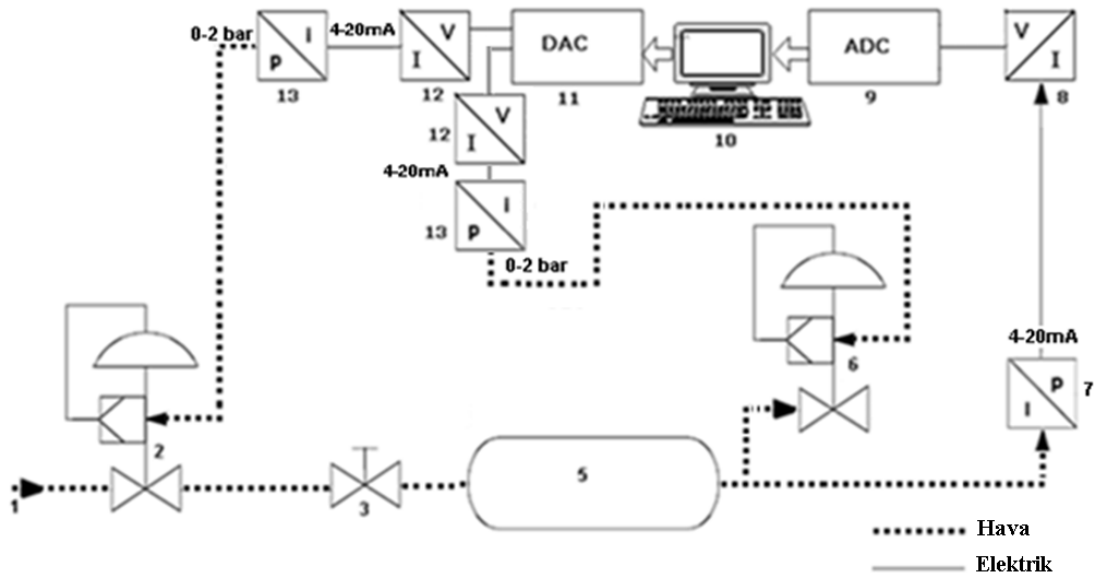


Şekil II. 15 Sistemin 0-10V basamak girişe karşılık dolum eğrileri

Verilen kontrol sinyallerinden 5V'un üzerinde olan girişlerde sistem sürekli rejime ortalama bir dakikada sürede geçtiği görülmektedir. Ancak tahliye vanasındaki sürekli hava çıkışından dolayı kontrol sinyalinin 5V ve altındaki değerlerinde, sistem davranışının yavaşladığı ve sürekli rejim cevabının çok geç gerçekleştiği görülmektedir.

II.4.3. Sistemin Çalışma Prensibi

Sistemin blok diyagramı Şekil II.16 'da verilmiştir. Sistemdeki amaç; basınç tankı(5) içerisindeki havanın istenilen referans değerde tutulmasıdır. Referans basınç değeri, bilgisayardan(10) sayısal analog dönüştürücü(11) üzerinden verilmektedir. Kontrolör, basınç transdüserinden (7) gelen bilgi ile basıncın o andaki değerini okur ve referans değeri ile karşılaştırarak gerekli olan kontrol sinyali akım/ basınç dönüştürücüye(13) buradan da pnömatik kontrol valfine(2) gönderir. Basınç tankındaki(5) hava, hava girişinden(1) sistemin çalışmasıyla birlikte sürekli olarak pnömatik oransal kontrol valfinin girişine gönderilmektedir. Pnömatik kontrol valfi, kontrolörden gelen kontrol sinyale göre girişinde bulunan havanın debisini değiştirerek, basınç tankına gönderir. Bu arada basınç tankının yan tarafındaki tahliye valfi (6), bozucu giriş olarak daimi hava çıkışını sağlar. Bu sebepten dolayı basınç seviyesi, referans değere ulaştığında dahi, pnömatik kontrol valf tam kapalı duruma gelmeyecektir. Valf gelen hava ile giden havayı dengede tutacak şekilde bir açıklıkta kalacaktır.



Şekil II. 16 Kontrol edilen sistemin blok diyagramı

- | | |
|--------------------------------------|-------------------------------------|
| 1. Hava girişi | 8. Akım gerilim dönüştürücü(I/V) |
| 2. Pnömatik Oransal Valf (Giriş) | 9. Analog sayısal dönüştürücü(ADC) |
| 3. Manüel vana | 10. Bilgisayar |
| 4. Manometre | 11. Sayısal analog dönüştürücü(DAC) |
| 5. Basınç tankı | 12. Gerilim akım dönüştürücü(V/I) |
| 6. Tahliye vanası (Pnömatik Oransal) | 13. Akım basınç dönüştürücü(I/P) |
| 7. Basınç transdüseri(P/I) | |

II.5. ÇÖZÜM İÇİN YAKLAŞIMLAR

Sistemin kontrolü, PID kontrolör ile gerçekleştirilmektedir. PID kontrolörün başarısı, katsayılarının(K_p , K_i ve K_d) uygun değerlerde seçilmiş olmasına bağlıdır. Bu katsayılar uygun değerde olmazsa sistem istenildiği gibi kontrol edilemez hatta sistem kararsızlığa doğru gider. Bu katsayıların sistem üzerindeki etkisi özet olarak;

- K_p artarsa aşım artar, cevap süresi azalır.
- K_d artarsa sönüm katsayısı artar, aşım ve salınım azalır.
- K_i artarsa sönüm katsayısı küçülür, salınım ve aşım artar[9, 16].

PID katsayıları çeşitli tekniklerle bulunabilmektedir. Döngüsel şekillendirmek(shaping), analitik metod(root locus tekniği) [4], kutup yerleştirmesi(pole placement) ve tekrarlı geri beslemeli optimizasyon[5] bu tekniklerden bazılarıdır. Ancak bu tekniklerin kullanılabilmesi için sistemin matematiksel modelinin bulunması gerekir. Sistemin davranışı doğrusal değilse matematiksel modelinin çıkartılması doğrusal davranış gösteren sistemlere nazaran zordur. Doğrusal davranış göstermeyen sistemlerde kontrolörün optimizasyonun daha kolay yapılabilmesi için YSA gibi yapay zekâya dayalı modeller kullanılmaktadır[17].

PID katsayılarının tespitinde en çok kullanılan yöntemlerden biri deneme-yanılma (hand-tuning) metodudur. Ancak bu yöntem zaman alıcı olması ve arama uzayının çok geniş olması gibi nedenlerle her sistem için uygun değildir. Bu yöntem yerine, GA, KKA, tabu arama, benzetilmiş tavlama gibi algoritmalar yaygın olarak kullanılmaktadır. Fakat bu yöntemlerde sistemin istenilen PID katsayılarını bulması sırasında ya sistemin gerçek zamanda çalıştırılmasına(çevrimiçi) ya da sistemin bir giriş-çıkış modeline(çevrimdışı) ihtiyaç duyulur.

Bu çalışmada, GUNT RT 532 Basınç Proses sisteminin YSA kullanılarak dinamik modelinin çıkartılması ve farklı referanslarda kontrol edilmesi için gerekli olan PID katsayılarının GA, KKA ve ZN ile bulunması gerçekleştirilmiştir. GA ve KKA ile farklı referanslarda kontrolör için gerekli katsayılar diğer yöntemlere nazaran daha kısa sürede ve daha sağlıklı bulunabilmektedir.

II.6. PID KONTROLÖR

Yeni teknolojilerin gelişmesi ile endüstriyel kontrol teknikleri gelişmektedir. Eksiklikleri giderilmeye çalışılmakla birlikte, basit yapısı ve kullanışlılığı ile PID kontrolör endüstriyel kontrol sistemlerinde halen kullanılmaya devam etmektedir[1, 18].

Oransal(Proportional), integral(Integral) ve türev(Derivative) kontrollerin toplamından oluşan PID kontrol, geri beslemeli kontrol sistemleri arasında popüler olarak kullanılan bir kontrol yöntemidir. Bu kontrol yönteminde hatanın miktarına, hatanın belli bir zaman aralığındaki integral değerine ve hatanın türevine bakılır. Ne kadar düzeltme sinyali uygulanacağı bilgisiyile birlikte aynı zamanda ne kadar süreyle uygulanacağı da belirlenmiş olur. Bu üç farklı değer belli değerlerde katsayılarıyla çarpılarak toplanır ve kontrol için proses girişine uygulanacak olan kontrol sinyali elde edilir.

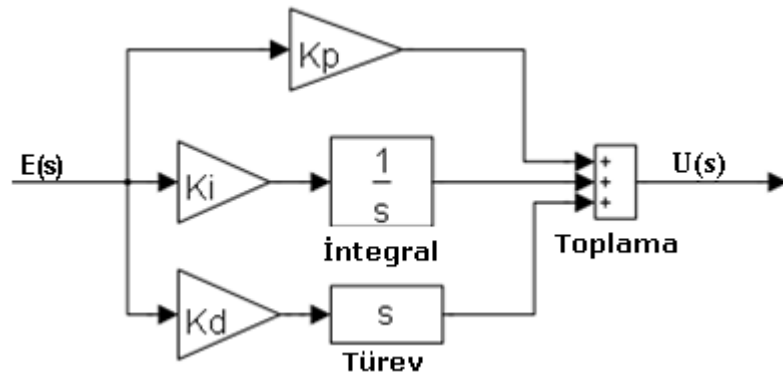
PID kontrolörün sürekli zaman formunda ifadesi Denklem II.1'deki gibidir.

$$u(t) = K_p \cdot e(t) + K_d \frac{de(t)}{dt} + K_i \int e(t) dt \quad (II.1)$$

Burada K_p orantı sabitini, K_i integral sabitini ve K_d türev sabitini, $u(t)$ kontrol sinyalini ve $e(t)$ hata değerini göstermektedir.

PID kontrolörün s uzayındaki ifadesi, Denklem II.2' de verilmiştir. PID kontrolörün blok şeması da Şekil II.17'de gösterilmektedir.

$$U(s) = E(s) \left(K_p + K_d s + \frac{K_i}{s} \right) \quad (II.2)$$



Şekil II. 17 PID kontrolörün blok şeması

Sistem kontrolü yapılırken ayırık düzlemde, belirli örnekleme zamanı(T_s) ile çalışan sayısal kontrolör kullanılır. Bu durumda PID kontrolörün formülü

$$U(z) = E(z)Kp + E(z)Ki \frac{1}{1 - z^{-1}} + E(z)Kd(1 - z^{-1}) \quad (\text{II.3})$$

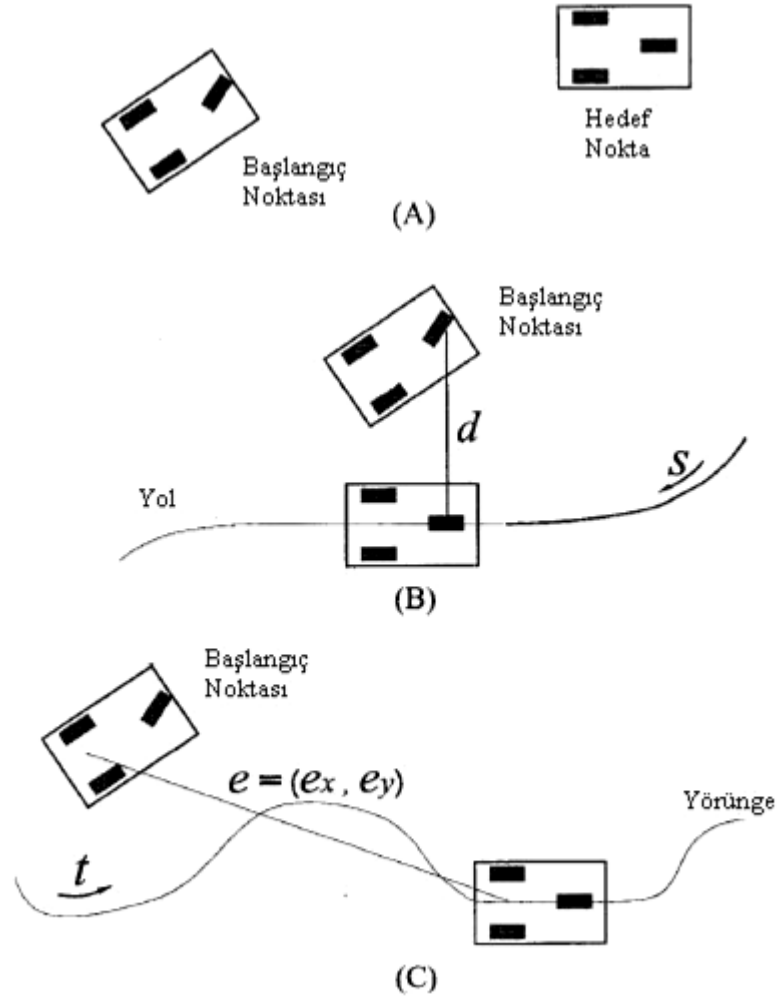
şeklinde olur. Burada $U(z)$ kontrol sinyalinin, $E(z)$ ise referans değeri ile proses geri besleme değeri arasındaki farkı göstermektedir.

PID kontrol, Şekil II.17' de gösterildiği gibi üç temel kontrol etkisini birleştiren bir kontrolördür. Oransal etki sistemin cevap hızını artırır, integral etki sistemde ortaya çıkabilecek kalıcı durum hatasını sıfırlar ve türev etki de sistemin cevabındaki titreşimleri bastırır, aşımı azaltır. Buna göre PID kontrol, sistemde sıfır sürekli rejim hatası olan, aşısız hızlı bir cevap sağlar.

II.7. YÖRÜNGE TAKİBİ

Sistemin kontrolü yapılırken referans olarak basamak giriş, rampa giriş gibi sabit referanslar verilebilir. Ancak sistemin ilk konumundan itibaren istenilen basınç değerlerinde seyretmesi için yörünge şeklinde referans verilebilir. Yörünge referansı ile sistem, ilk konumdan son konuma ve ara değerlerde hangi konumda olması gerektiğini bilecektir.

Başlangıç pozisyonundan sonuç pozisyonu kadar hareketi esnasında yer değiştirme yollarını belirleyen noktalar kümesine yörünge denir[19]. Hareketin aktivitesine göre, yörünge planlaması, noktadan noktaya hareket, yol takibi veya yörünge takibi şeklindedir. Şekil II.18'de bu üç yörünge planlamasının çalışması görülmektedir. Noktadan noktaya harekette, başlangıçta seçilen bir noktadan istenilen hedefteki noktaya ulaşılmasıdır. Yol takibinde ise, verilen başlangıç ve bitiş noktaları arasındaki geometrik yolu sistemin takip ederek ulaşmasıdır. Yörünge takibi, kartezyen koordinat sistemi içinde başlangıç noktasından verilen başlangıç konfigürasyona göre bir yörünge takip etmektedir[20].



Şekil II. 18 (A) noktadan noktaya (B) yol takibi (C) yörünge takibi

Yörünge tanımında birçok matematiksel fonksiyon kullanılabilir. Kübik fonksiyon, hareket eğrisinin üretilmesinde kullanılmaktadır. Bu eğri, düşük dereceli eğriler yerine polinom eğrilerinin ara değer kestirimi için daha tahmin edilebilir durumdadır[21].

II.7.1. Sistemde Kullanılan Yörünge Denklemi

Basınç Proses Kontrol sisteminde, basınç tankının içindeki hava basıncının kontrolü gerçekleştirilmektedir. Burada istenen, tankın içindeki hava basıncının ilk değerinden istenilen basınç değerine, Şekil II.19’da görülen, üçüncü dereceden denklemin çizdiği yolu takip ederek çıkmasıdır. Aynı zamanda bu eğrinin, istenilen noktaya istenilen gecikme zamanı ile ulaşması istenmektedir. Sistemde kullanılan üçüncü dereceden yörünge denkleminin matematiksel ifadesi Denklem II.4 de verilmiştir [22].

$$\begin{aligned}
p_y &= p_0 & t < t_0 \\
p_y &= f_3 t^3 + f_2 t^2 + f_1 t + f_0 & t_0 < t < t_s \\
p_y &= p_s & t > t_s
\end{aligned} \tag{II.4}$$

Burada p_y , yörünge değeri, t_0 gecikme zamanı, f_0 ve f_3 katsayılar, t ise zamandır. Katsayılar sırasıyla;

$$t_s = 3\epsilon \frac{P_s - P_0}{P_s + P_0}$$

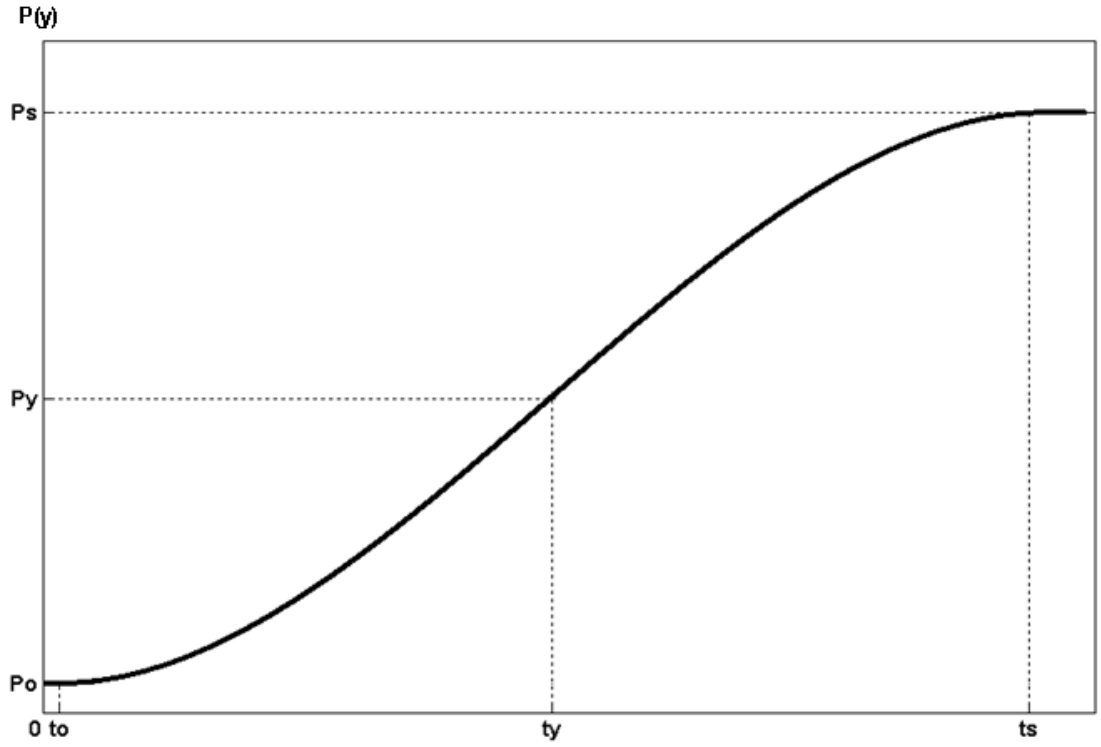
$$f_0 = P_0$$

$$f_1 = 0$$

$$f_2 = -3 \frac{P_0 - P_s}{t_s^2}$$

$$f_3 = 2 \frac{P_0 - P_s}{t_s^3}$$

Burada; t_s oturma zamanı, ϵ eğim değeri, P_s istenilen referans noktası ve P_0 başlangıç noktasıdır. Yapılan çalışmada kullanılan yörünge grafiği Şekil II.19'da gösterilmektedir.

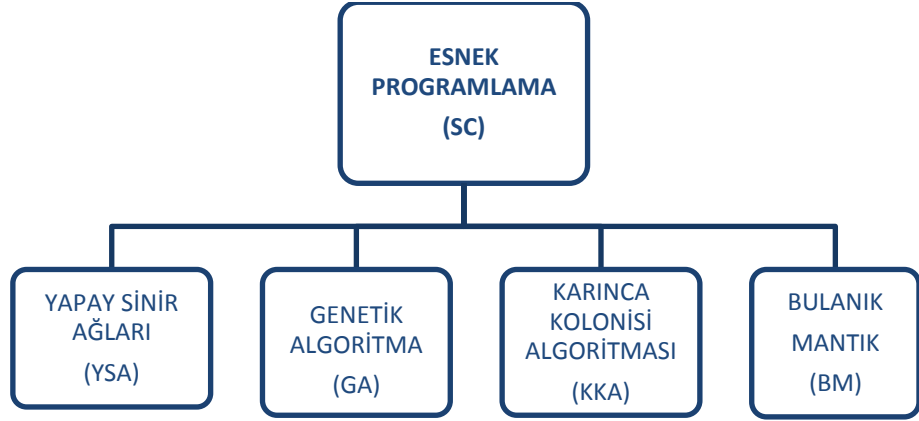


Şekil II. 19 Çalışmada kullanılan yörünge grafiği

BÖLÜM III

ESNEK PROGRAMLAMA(SC)

YSA, GA ve KKA gibi yapay zekâ tekniklerinin birlikte kullanılması SC'yi oluşturmaktadır. Bu yöntemlerde, GA doğal genetiği, YSA sistemin öğrenme yapısını, bulanık mantık(FL) sistemin karar verme yapısını, KKA ise var olan çözümlerden en kısa olanının bulunmasını amaçlamıştır. Şekil III.1'de SC' yi oluşturan yöntemler görülmektedir [23].



Şekil III. 1 Esnek programlamayı oluşturan programlama teknikleri

III.1. YAPAY SİNİR AĞLARI(YSA) ile SİSTEM TANIMLAMA

III.1.1. Giriş

İnsanlığın doğayı araştırma ve taklit etme çabalarının en son ürünlerinden bir tanesi, YSA teknolojisidir. YSA, basit biyolojik sinir sisteminin çalışma şekli simüle edilerek tasarlanan programlama yaklaşımıdır. Simüle edilen sinir hücreleri (nöronlar) içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA' lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik (synaptic) bağlantıların ayarlanması ile olur. Yani, insanlar doğumlarından itibaren yaşayarak öğrenme süreci içerisine girerler. Bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. Bu durum YSA için de geçerlidir. Öğrenme, eğitme yoluyla örnekler kullanarak olur; başka bir deyişle, gerçekleşme girdi/çıktı verilerinin işlenmesiyle, yani eğitme algoritmasının bu verileri kullanarak bağlantı ağırlıklarını (weights of the synapses) bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur.

YSA'lar, belirli katsayılar ile ağırlıklandırılarak birbirlerine bağlanmış birçok işlem biriminden (nöronlar) oluşan matematiksel sistemlerdir. Bir işlem birimi, aslında sık sık transfer fonksiyonu olarak anılan bir denklemdir. Bu işlem birimi, diğer nöronlardan sinyalleri alır; bunları birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkarır. Genelde, işlem birimleri kabaca gerçek nöronlara karşılık gelirler ve bir ağ içinde birbirlerine bağlanırlar; bu yapı da sinir ağlarını oluşturmaktadır.

Sinirsel (neural) hesaplamanın merkezinde dağıtılmış, adaptif ve doğrusal olmayan işlem kavramları vardır. YSA' lar, geleneksel işlemcilerden farklı şekilde işlem yapmaktadırlar. Geleneksel işlemcilerde, tek bir merkezi işlem birimi her hareketi sırasıyla gerçekleştirir. YSA' lar ise her biri büyük bir problemin bir parçası ile ilgilenen, çok sayıda basit işlem birimlerinden oluşmaktadır. En basit şekilde, bir işlem birimi, bir girdiyi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir

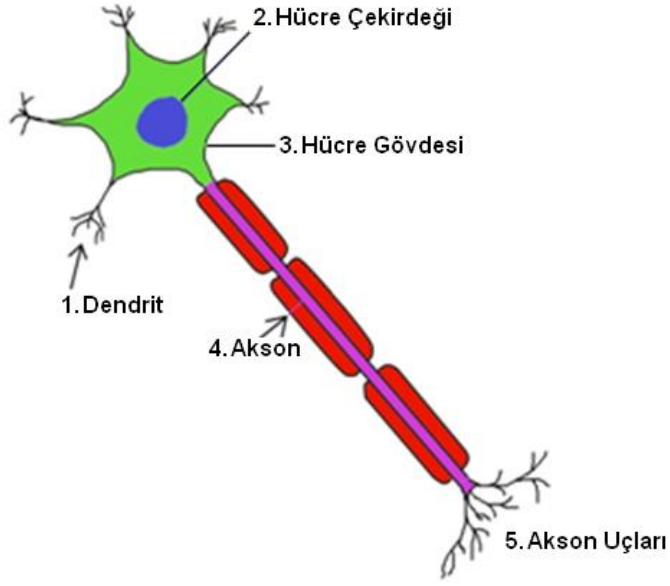
şekilde dönüşümünü sağlar ve bir çıktı değeri oluşturur. İlk bakışta, işlem birimlerinin çalışma şekli yanıltıcı şekilde basittir. Sinirsel hesaplamanın gücü, toplam işlem yükünü paylaşan işlem birimlerinin birbirleri arasındaki yoğun bağlantı yapısından gelmektedir.

Çoğu YSA' da, benzer karakteristiğe sahip nöronlar tabakalar halinde yapılandırılırlar ve transfer fonksiyonları eş zamanlı olarak çalıştırılırlar. Hemen hemen tüm ağlar, veri alan nöronlara ve çıktı üreten nöronlara sahiptirler.

YSA' nın ana ögesi olan matematiksel fonksiyon, ağın mimarisi tarafından şekillendirilir. Daha açık bir şekilde ifade etmek gerekirse, fonksiyonun temel yapısını ağırlıkların büyüklüğü ve işlem elemanlarının işlem şekli belirler. YSA' ların davranışları, yani girdi veriyi çıktı veriye nasıl ilişkilendirdikleri, ilk olarak nöronların transfer fonksiyonlarından, nasıl birbirlerine bağlandıklarından ve bu bağlantıların ağırlıklarından etkilenir[24].

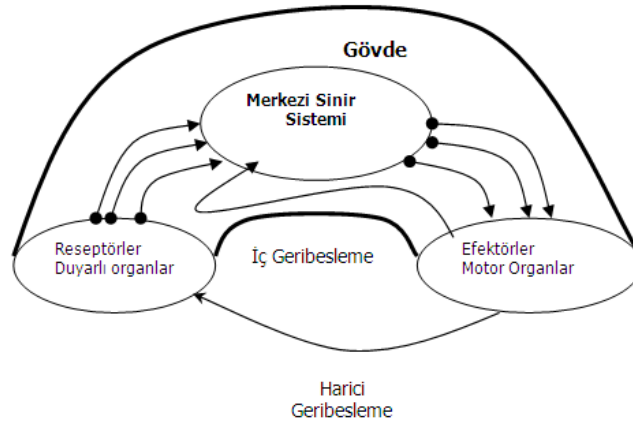
III.1.2. Biyolojik Sinir Ağları

Biyolojik sinir ağının temel inşaa bloğu olan basit bir sinir hücresi nöron olarak adlandırılır. Şekil III.2'de biyolojik sinir hücresi görülmektedir. Nöron ya da sinir hücresi, sinir sisteminin temel fonksiyonel birimidir. Çeşitli biçim ve büyüklüklerde olabilir. Sinirsel uyarıları elektriksel ve kimyasal yolla iletir. Aşağıdaki şekilde şematik diyagramı gösterilen tipik sinir hücresi üç ana bölümden oluşur. Hücre gövdesi büyüktür ve çekirdek burada yer alır. Bunun dışında, akson ve dendrit denilen uzantıları vardır. Başka hücrelerden gelen uyarılar(giriş) dendritlerin uçlarından alınır, soma olarak adlandırılan hücre gövdesi(akson ve dendritler) tarafından işlenir ve sinyalleri taşıyan uzun bir sinirsel bağlantı halindeki akson ile uçlarından diğer hücelere iletilir. Akson dendrit bağlantısı ise sinaps olarak adlandırılır. Sinaps nöronlar arasında elektrokimyasal bağlantıyı sağlamaktadır.



Şekil III. 2 Doğal sinir hücresi

Nöronlar birbirleriyle elektriksel sinyaller aracılığıyla haberleşmektedir. Ayrıca, nöronlar kimyasal bir ortamda çok yoğun beyinsel faaliyetleri yerine getirmektedirler. Böylece beyin, biyokimyasal işlemlerin gerçekleştiği son derece yoğun bir elektriksel ağ gibi düşünülebilir. Büyük sinir ağları çok karmaşık ve ayrıntılı bir yapıyla birbirine bağlıdır. Ağa giriş duyarlı algılayıcılar (reseptörler) ile sağlanır. Reseptörler uyarıyı gövdeye götürürler. Uyarı elektriksel sinyaller biçimindedir. Nöron ağının içine bilgi taşınması ve merkezi sinir sisteminde bilginin işlenmesi sonucu efektörler kontrol edilir. Bundan sonra insan cevabını çeşitli eylemler şeklinde verir. Şekil III.4’ de gösterildiği gibi sinir sisteminde bilgi akışı üç ana kısımdan oluşmaktadır: reseptörler, sinir ağı ve efektörler[21].



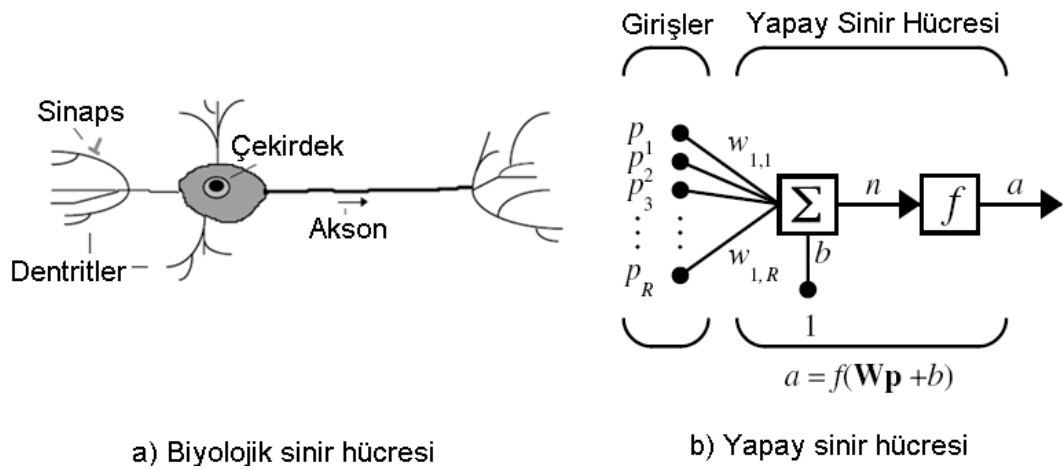
Şekil III. 3 Sinir sisteminde bilgi akışı

Şekil III.4’ de gösterilen sistemde bilgi işlenmekte, değerlendirilmekte ve merkezi sinir sisteminde depolanan bilgiyle karşılaştırılmaktadır. Motor organlar

eylemi doğrulayan geri beslemeli bağlantılarla merkezi sinir sistemini yönetir ve denetlerler. İç ve dış geri beslemeli kontrolün ikisi de komutlarla gerçekleştirilir. Tüm sinir sisteminin yapısı kapalı-çevrim bir kontrol sistemini andırmaktadır.

III.1.3. Yapay Sinir Ağı Hücresi(Perceptron)

Tek katmandan oluşan ve basit bir aktivasyon fonksiyonuna sahip olan yapay sinir ağı hücresine perceptron denir. Temel bir yapay sinir ağı hücresi biyolojik sinir hücresine göre çok daha basit bir yapıya sahiptir. Biyolojik sinir hücresi Şekil III.4(a)'da, en temel nöron modeli Şekil III.4(b)' de görülmektedir[25].



Şekil III. 4 Biyolojik ve yapay sinir hücresi

Yapay sinir ağı hücresinde temel olarak dış ortamdan ya da diğer nöronlardan alınan veriler yani girişler($\mathbf{p}$), ağırlıklar($\mathbf{w}$), toplama fonksiyonu, aktivasyon fonksiyonu(f) ve çıkışlar($\mathbf{a}$) bulunmaktadır. Dış ortamdan alınan veri, ağırlıklar aracılığıyla nörona bağlanır ve bu ağırlıklar ilgili girişin etkisini belirler. Toplam fonksiyonu ise $net(n)$ girişi hesaplar, $net(n)$ giriş, girişlerle bu girişlerle ilgili ağırlıkların çarpımının bir sonucudur. Aktivasyon fonksiyonu(f) işlem süresince net çıkışını hesaplar ve bu işlem aynı zamanda nöron çıkışını($\mathbf{a}$) verir. Genelde aktivasyon fonksiyonu doğrusal olmayan (lineer olmayan) bir fonksiyondur. Şekil III.4 (b)' de görülen $\mathbf{b}$ (bias) bir sabittir, bias veya aktivasyon fonksiyonunun eşik değeri olarak adlandırılır.

Ağın giriş sayısı R ile, aktivasyon fonksiyonu f ile ve ağın çıkışı da a ile gösterilirse;

$$\mathbf{a} = f(\mathbf{W} \cdot \mathbf{p} + \mathbf{b}) \quad (\text{III.1})$$

Burada ;

$\mathbf{W} = w_1, w_2, w_3, \dots, w_R$ ağırlık matrisi,

$\mathbf{p} = p_1, p_2, p_3, \dots, p_R$ girişler vektörü,

$\mathbf{b} = b_1, b_2, b_3, \dots, b_R$ bias vektörüdür.

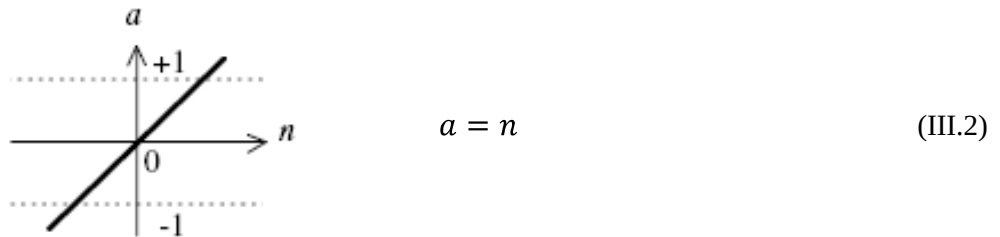
III.1.3.1. Aktivasyon Fonksiyonu

Transfer fonksiyonu olarak da geçen aktivasyon fonksiyonu, birleştirme fonksiyonundan elde edilen net girdiyi bir işlemde geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur. Bir yapay sinir ağının davranışı ağırlıklarının ve aktivasyon fonksiyonunun değerine bağlıdır. Hücre modellerinde, hücrenin gerçekleştireceği işleve göre çeşitli tipte aktivasyon fonksiyonları kullanılabilir. Aktivasyon fonksiyonları sabit parametrelili ya da uyarlanabilir parametrelili seçilebilir. Aktivasyon fonksiyonunun seçimi büyük ölçüde yapay sinir ağının verilerine ve ağın neyi öğrenmesinin istendiğine bağlıdır.

Aktivasyon fonksiyonları bir YSA' da nöronun çıkış genliğini, istenilen değerler arasında sınırlar. Bu değerler genellikle $[0,1]$ veya $[-1,1]$ arasındadır. Eğer türeve dayalı bir öğrenme algoritması kullanılacak ise YSA' da kullanılacak aktivasyon fonksiyonlarının türevi alınabilir olması ve süreklilik arz etmesi gereklidir. Lineer veya doğrusal olmayan transfer fonksiyonlarının kullanılması YSA' ların karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır.

1.a. Doğrusal Aktivasyon Fonksiyonu

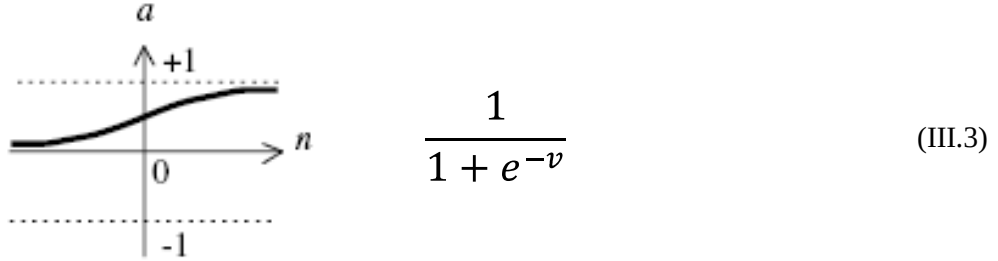
Doğrusal bir problemi çözmek amacıyla kullanılan doğrusal hücre ve YSA' da ya da genellikle katmanlı YSA' nın çıkış katmanında kullanılan doğrusal fonksiyon, hücrenin net girdisini doğrudan hücre çıkışı olarak verir. Doğrusal aktivasyon fonksiyonu Şekil III.5'te gösterildiği gibi girişe uygulanan değeri aynen çıkış olarak verir.



Şekil III. 5 Doğrusal aktivasyon fonksiyonu

1.b. Sigmoid Aktivasyon Fonksiyonu

Sigmoid Aktivasyon Fonksiyonu, türevi alınabilir, sürekli ve doğrusal olmayan bir fonksiyon olması nedeniyle uygulamada en çok kullanılan aktivasyon fonksiyonudur. Bu fonksiyon, girdinin her değeri için sıfır ile bir arasında bir değer üretir. Şekil III.6’da sigmoid aktivasyon fonksiyonun grafiği görülmektedir.

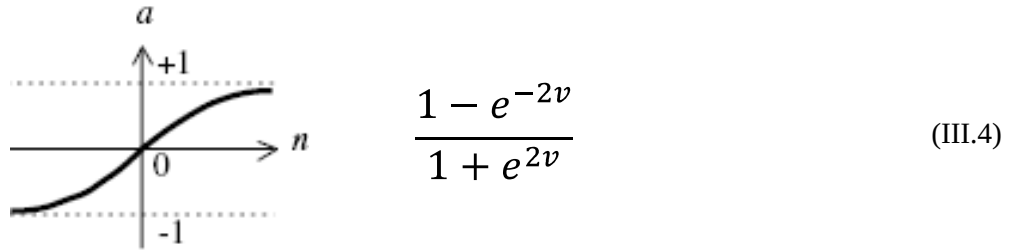


Şekil III. 6 Sigmoid fonksiyon

şeklinde. Burada v sabit bir sayıdır ve fonksiyonun eğimini belirler.

2.c. Tanjant Hiperbolik

Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonunun biraz farklı şeklidir. Giriş uzayının genişletilmesinde etkili bir aktivasyon fonksiyonudur. Sigmoid fonksiyonun çıkış aralığı 0 ve 1 olurken, hiperbolik tanjant fonksiyonunun çıkışı -1 ve 1 aralığında oluşmaktadır ve Şekil III.7’de gösterilmektedir.



Şekil III. 7 Hiperbolik tanjant fonksiyonu

Yukarıda anlatılan aktivasyon fonksiyonlarından başka;

- Basamak Fonksiyonu
- Kutuplamalı Basamak Fonksiyonu
- Parçalı Doğrusal Fonksiyon

aktivasyonları kullanılabilir[26].

III.1.4. Yapay Sinir Ağları Modelleri

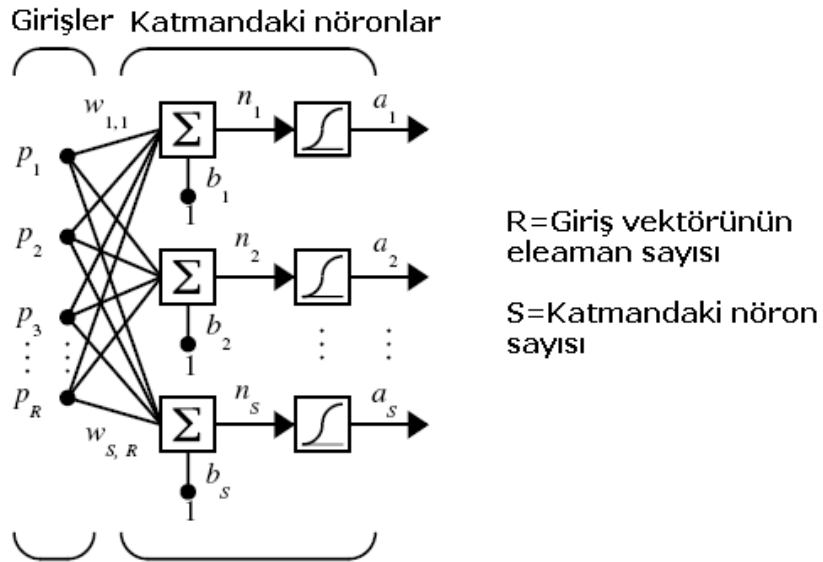
YSA modellerini sınıflandırarak iki kısımda inceleyebiliriz.

- A. Mimari yapılarına göre
 - İleri beslemeli
 - Geri beslemeli
- B. Öğrenme yaklaşımlarına göre
 - Danışmanlı öğrenme
 - Danışmansız öğrenme
 - Takviyeli öğrenme

III.1.4.1. Mimari Yapılarına Göre YSA

1.a. İleri Beslemeli Yapılar

En basit ve en genel yapay sinir ağları tek yönlü sinyal akışını kullanırlar. İleri beslemeli yapay sinir ağlarında gecikmeler yoktur, işlem girişlerden çıkışlara doğru ilerler. Çıkış değerleri öğreticiden alınan istenen çıkış değeriyle karşılaştırılarak bir hata sinyali elde edilerek ağ ağırlıkları güncellenir. Şekil III.8'de ileri beslemeli ağ modeli görülmektedir.



Şekil III. 8 İleri beslemeli YSA modeli

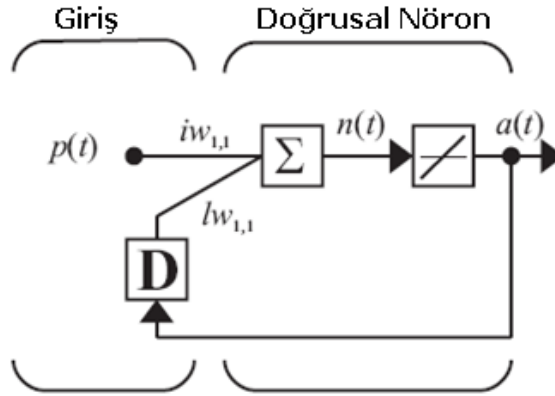
2.b. Geri Beslemeli Yapılar

Geri beslemeli yapay sinir ağlarında ise tıpkı kontrol uygulamalarında olduğu gibi gecikmeler söz konusudur. Geri beslemeli nöral ağ, çıkışlar girişlere bağlanarak ileri beslemeli bir ağdan elde edilir. Ağın t anındaki çıkışı $a(t)$ ise, $t + D$ anındaki çıkışı ise $a(t+D)$ ' dir.

$$a(0)=p(0),$$

$$a(t)=iw_{1,1}p(t)+lw_{1,1}a(t-1) \quad (III.5)$$

Bu formül Şekil III.9’da ifade edilmiştir. Burada $iw_{1,1}$ giriş ağırlık katsayısını, $lw_{1,1}$ ağın çıkışından gelen girişin ağırlık katsayısını ve D ise zaman gecikmesini göstermektedir. D , bir örnekleme zamanı öncesi bilgiyi hafızada tutar ve bir sonraki çıkışa giriş olarak uygulanır.



Şekil III. 9 Geri beslemeli YSA yapısı

Şekil III.9’da **D(delay)** harfi ile gösterilen zaman gecikmesi, bir örnekleme zamanı önceki bilgiyi hafızada tutar ve bir önceki çıkışı, YSA’ nın girişi olarak uygular.

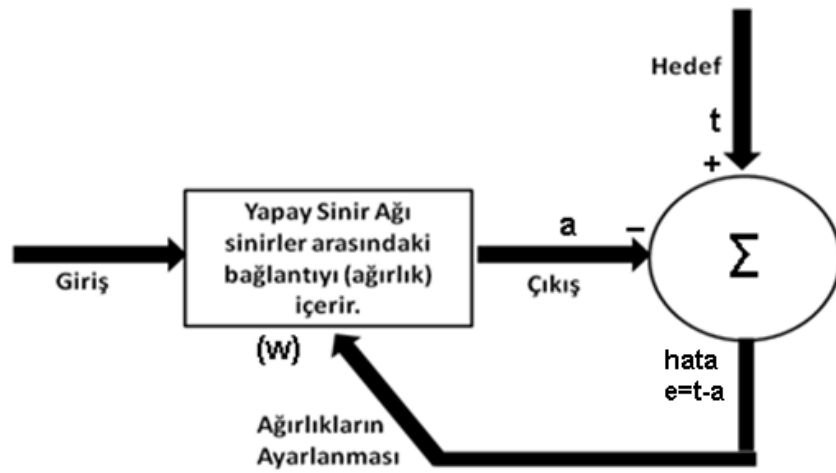
III.1.4.2. Öğrenme Yapılarına Göre YSA

İnsan beyni doğumdan sonraki gelişme sürecinde çevresinden duyu organlarıyla algıladığı davranışları yorumlar ve bu bilgileri diğer davranışlarında kullanır. Yaşadıkça beyin gelişir ve tecrübelenir. Artık olaylar karşısında nasıl tepki göstereceğini çoğu zaman bilmektedir. Fakat hiç karşılaşmadığı bir olay karşısında yine tecrübesiz kalabilir. Yapay sinir ağlarının öğrenme sürecinde de, tıpkı dış ortamdan gözle veya vücudun diğer organlarıyla uyarıların alınması gibi dış ortamdan girişler alınır, bu girişlerin beyin merkezine iletilerek burada değerlendirilip tepki verilmesi gibi yapay sinir ağında da aktivasyon fonksiyonundan geçirilerek bir tepki çıkışı üretilir.

1990’lı yıllardan bugüne gelinceye kadar birçok öğrenme algoritması geliştirilmiştir. Öğrenme algoritmaları temelde üç grupta toplanmaktadır. Eğitici öğrenme, eğitici öğrenme ve takviyeli öğrenme algoritmaları.

2.a. Eđiticiili ğrenme

Eđiticiili ğrenmede, giriřler ađa uygulanıp aktivasyon fonksiyonundan geilerek ıkıřlar elde edilir. Bu ıkıřlar, tecrbeyle verilen ıkıřla karřılařtırılarak hata bulunur. eřitli ğrenme algoritmalarıyla hata azaltılıp gerek ıkıřa yaklařılmaya alıřılır. Bu alıřma suresince yenilenen yapay sinir ađının ađırlıklarıdır. řekil III.10'da gsterildiđi gibi ađırlıklar her bir evrimde yenilenerek amaca ulařılmaya alıřılır. Amaca ulařmanın veya yaklařmanın ls de yine dıřarıdan verilen bir deđerdir. Ađırlıkların gncellenmesi suresince eđitici, dllendirme-cezalandırma řemasını ađa uygulayarak hatayı azaltır. Bu ğrenme modelinde giriř ve ıkıř rnekleri kmesi eđitim kmesi olarak adlandırılır[25].



řekil III. 10 YSA' nın eđiticiili ğrenme ile eđitilmesi

Eđer yapay sinir ađı verilen giriř-ıkıř iftleriyle amaca ulařmıř ise ađırlık deđerleri saklanır. Ađırlıkların srekli yenilenip istenilen sonuca ulařılana kadar geen zamana ğrenme adı verilir. Yapay sinir ađı ğrendikten sonra daha nce verilmeyen giriřler verilip, sinir ađı ıkıřıyla gerek ıkıřı yaklařımı incelenir. Eđer yeni verilen rneklerle de dođru yaklařıyorsa sinir ađı iři ğrenmiř demektir. Sinir ađına verilen rnek sayısı optimum deđerden fazla ise sinir ađı iři ğrenmemiř ezberlemiřtir. Genelde eldeki rneklerin yzde sekseni ađa verilip ađ eđitilir, daha sonra geri kalan yzde yirmilik kısım verilerek ađın davranıřı incelenir. Diđer bir deyiřle ađ test edilir.

2.b. Eđiticisiz ğrenme

Eđiticisiz ğrenmede, eđiticiili ğrenmedeki gibi arzu edilen y ıkıřları bilinmemektedir. Bu yzden kesin bir hata bilgisini ađın davranıřını deđiřtirmekte kullanmak mmkn deđildir. Cevabın dođruluđu veya yanlıřlıđı hakkında bilgi

sahibi olunmadığı için öğrenme, girişlerin verdiği cevaplar gözlenerek başarıya ulaşılır. Eğitici-siz öğrenme yaklaşımı daha çok sınıflandırma problemlerinin çözümünde kullanılır.

2.c. Takviyeli öğrenme

Takviyeli öğrenmede, ağıın davranışının uygun olup olmadığını belirten bir öz yetenek bilgisine ihtiyaç duyulur. Bu bilgiye göre ağırlıklar ayarlanır. Gerçek zamanda öğrenme yöntemi olup deneme-yanılma esasına göre sinir ağı eğitilmektedir[25].

III.1.5. Hatanın Geri Yayılımı Öğrenme Algoritması

Hatanın geri yayılımı öğrenme algoritması ileri beslemeli çok katmanlı ağların eğitilmesi için kullanılan en genel algoritmadır.

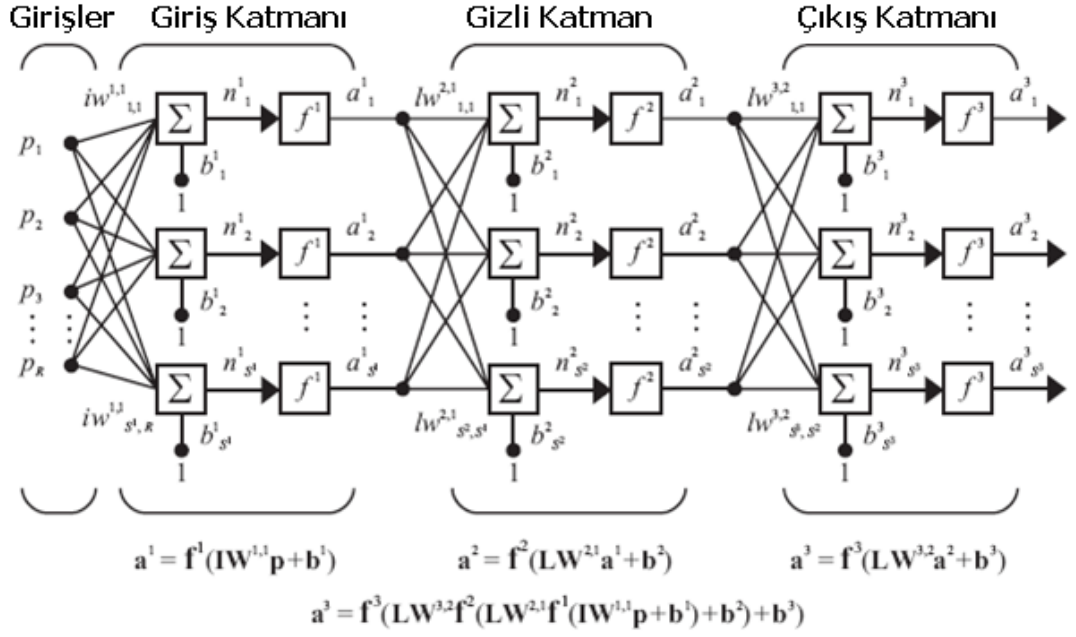
Öğrenme fazında, giriş örnekleri ağına belirli bir sırada sunulur. Her bir çalışma örneği çıktı örneği hesaplanana kadar katman katman ileri yayılır. Hesaplanan çıktı daha sonra hedef çıktı ile karşılaştırılarak hata sinyali bulunur. Hata yani $e(k)$, arzu edilen çıkış($y(k)$) ile sinir ağıının çıkışı ($a(k)$) arasındaki farktır.

$$e(k) = y(k) - a(k) \quad (III.6)$$

Hatalar, katman katman sinaptik ayarlamaların yapıldığı geri besleme bağlantılarına ‘girdiler’ olarak sunulmaktadır.

Geri yayılım öğrenme kullanıldığında, sonraki katmanların hataları kullanılarak gizli katmanın ağırlıkları ayarlanır. Böylece çıkış katmanında hesaplanan hatalar son gizli katman ile çıkış katmanı arasındaki ağırlıklar ayarlanır. Aynı biçimde, bu işlemler ilk gizli katmana kadar tekrarlanır. Bu yolla hatalar katman katman ilgili katmanın ağırlık düzeltmeleri yapılarak geriye doğru yayılır. Tamamlanan çalışma süresi içinde ‘toplam hata’ en aza indirilinceye kadar bu işlemler tekrarlanır[27].

Şekil-III.11’de birçok sinir hücresinin bir birine bağlandığı ileri yönlü çok katmanlı bir yapay sinir ağı görülmektedir[28]. Giriş katmanı ile çıkış katmanı arasındaki katman veya katmanlar gizli katman olarak adlandırılır. Sinir ağlarında kaç tane gizli katman kullanılacağı ve her bir gizli katmanda kaç nöron olacağı probleme göre değişen ve deneme-yanılma yoluyla bulunan niteliklerdir.



Şekil III. 11 İleri beslemeli çok katmanlı sinir ağı

Şekil III.11'deki çok katmanlı ağı düşünürsek ağı eğitilmesi 4 adımda gerçekleşir;

1) İleri Yön Değerleri Hesaplanır;

$$\begin{aligned}
 a^0 &= p && \text{; giriş katmanı için} \\
 a^{m+1} &= f^{m+1}(W^{m+1}a^m) && \text{; } m=1,2, \dots, M-1 \\
 a^0 &= a^M && \text{; çıkış katmanı için}
 \end{aligned} \tag{III.7}$$

Burada k=giriş sayısı, m=katman sayısıdır.

2) Hatanın bulunması;

$$e = t - a$$

Burada t hedef çıkış ve a ağı çıkışıdır.

3) Hatanın Geriye Yayılması;

$$s^M = -2F^M(n^M)e \quad \text{; çıkış katmanı için} \tag{III.8}$$

$$s^m = F^m(n^m)(w^{m+1})^T s^{m+1} \quad \text{; } m=M-1, \dots, 2, 1 \tag{III.9}$$

Burada s her bir katmanın duyarlılık değerleridir. F= her bir katman için aşağıdaki gibi tanımlanır.

$$F^m(n^m) = \begin{bmatrix} \frac{\partial f^m(n_1^m)}{(n_1^m)} & 0 & \dots & 0 \\ 0 & \frac{\partial f^m(n_2^m)}{(n_2^m)} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \frac{\partial f^m(n_s^m)}{(n_s^m)} \end{bmatrix} \tag{III.10}$$

4) Ağırlıkların ayarlanması;

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha s^M (a^{M-1})^T \quad (\text{III.11})$$

Burada α = öğrenme oranıdır ve 0 ile 1 arasında bir değerdir. Tüm öğrenme algoritmalarında şu adımlar izlenir. İlk önce (III.7) kullanılarak giriş ileri yönde yayılır, sonra (III.8) kullanılarak geri yayılır ve son olarak ağırlıklar (III.11) kullanılarak güncellenir[25].

III.1.6. YSA ile Sistem Modelleme

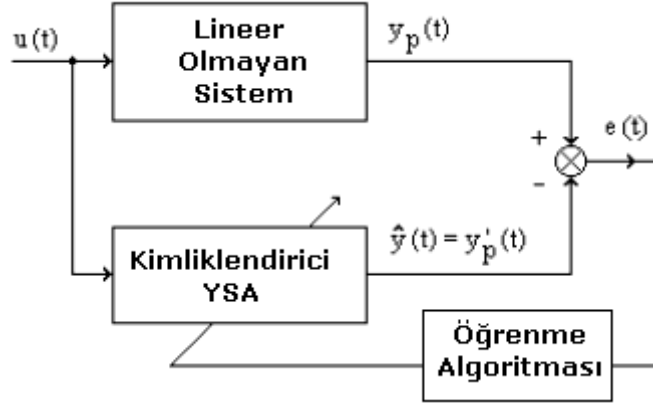
Modelleme ve ölçme, sistemlerin tasarımında ve denetimde çok büyük bir öneme sahiptir. Klasik modelleme anlayışında sistemlerin tüm elemanlarının ve bağlantı tarzlarının fiziksel kanunlara dayalı analitik ifadeleri bulunmalıdır. Gerçekte bu işlem, karmaşık sistemlerde ya hiç mümkün değil ya da çok zordur. Bu yüzden modern tanılama (modelleme) anlayışında, sisteme uygun bir model önerilip giriş-çıkış veri kümesi yardımı ile bu model öğretilir ve güncelleştirilir. Son yıllarda, özellikle, karmaşık ve lineer olmayan sistemlerin modellenmesinde yapay sinir ağları (YSA) yaygın olarak kullanılmaya başlanmıştır. YSA' nın lineer olmayan yapısı öğretilen sistemlerinki ile iyi bir uyum sağlamaktadır.

Genel olarak; sistem modelinin elde edilmesi; giriş, çıkış ve durum değişkenlerinin ölçülmesi tabanında fiziksel sistemin yapısının öğrenilmesi anlamına gelir. Durum değişkenlerinden biri veya birkaçı sıklıkla sistem çıkışı olabilir. Bir sistem modeli, matematiksel ifade, giriş-çıkış data seti veya dilsel kurallar kullanılarak belirtilebilir.

Yapay sinir ağları (YSA) lineer ve lineer olmayan yapıdaki sistemlerin modellenmesinde kullanılabilir.

III.1.6.1. İleri Modelleme (Seri-paralel modelleme)

Zamanla değişen, dinamik bir sistemin giriş ve çıkış, sırasıyla u ve y_p olarak gösterilir. Eğer model ile sistem aynı sistem girişi u ile çalıştırılıyorsa; modelin ürettiği çıkış($\hat{y}$)'ın y_p ye yaklaşması gerekmektedir. Şekil III.13' de bu metoda ait blok şema görülmektedir.



Şekil III. 12 YSA ile modellemede kullanılan temel ileri tip blok şema

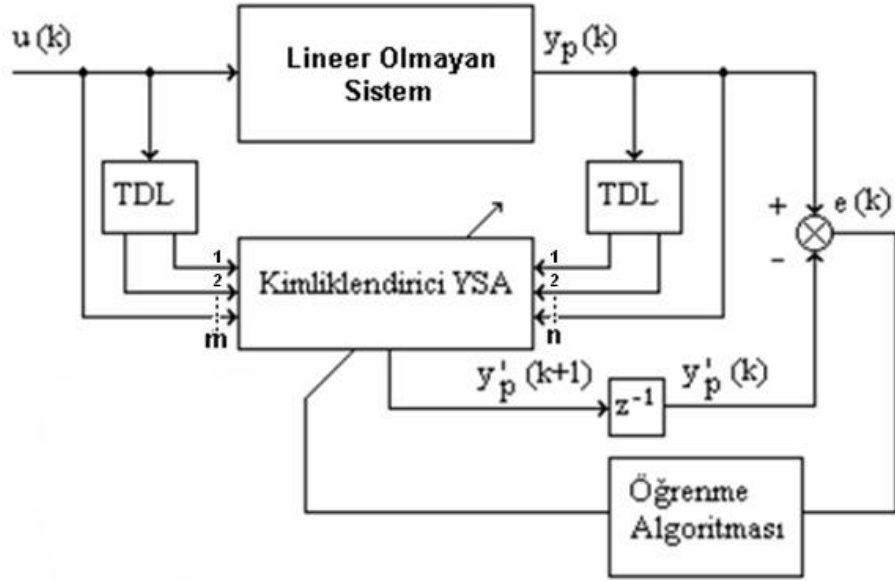
Şekil III.13’de Kimliklendirici Yapay Sinir Ağı (YSA) bloğu, lineer olmayan sisteme paralel olarak yerleştirilmiştir. Sistem çıkışı y_p ile YSA çıkışı $\hat{y}(t)$ arasındaki hata (tahmin hatası) $e(t) = y_p - \hat{y}$, YSA’nın öğretilmesinde kullanılmaktadır. Çok katmanlı ileri beslemeli YSA modelinde, hatanın geri yayılım algoritması kullanılır[29]. Buna göre YSA’ nın sistem dinamiğinin öğrenmesinde eğitici öğrenme prosedürü kullanılmaktadır. Sistemin aşağıda verilen lineer olmayan ayrık zamanlı fark denklemi ile belirtildiği farz edilsin.

$$y_p(k+1) = F[y_p(k), \dots, y_p(k-n+1); u(k), \dots, u(k-m+1)] \quad (\text{III.12})$$

Burada, $y_p(k+1)$, sistem çıkışının $(k+1)$ örnekleme zamanındaki değeri, $F(.)$ ise, n ve m pozitif tamsayılar olmak üzere geçmiş n sistem çıkışına ve geçmiş m girişe bağlı olan çok boyutlu lineer olmayan bir sistem tanımlanmasıdır. YSA’ nın giriş-çıkış yapısının seçiminde, giriş-çıkış yapısının sisteminki ile aynı olduğu farz edilir[30]. Bu yüzden, YSA çıkışı şu şekilde belirtilebilir.

$$\hat{y}(k+1) = y_p'(k+1) = N[y_p(k), \dots, y_p(k-n+1); u(k), \dots, u(k-m+1)] \quad (\text{III.13})$$

Burada $N(.)$, YSA’ nın lineer olmayan tanımlanmasıdır ve $\hat{y}$ tahmin edilen YSA çıkışıdır. Buradan anlaşıldığı gibi, YSA çıkışı, geçmiş n adet sistem çıkışına ve m adet girişe bağlıdır. YSA’ nın geri beslemesinin olmamasından dolayı YSA’ nın geçmiş modlarına bağlı değildir.



Şekil III. 13 Seri – paralel ileri tip modelleme blok şeması

Şekil III.13, (III.12) ve (III.13) nolu bağıntıları kullanarak oluşturulan kimliklendirme algoritmasının blok diyagramını göstermektedir. Bu seri-paralel kimliklendirme sisteminde; ana ilke olarak, sistem çıkışı ve onun geriye kaymış modları YSA modeline giriş olarak sağlanmaktadır. Şekil III.13’ de TDL isimli bloklar zaman gecikme döngülerini ifade etmektedir[17, 30, 31].

III.2. GENETİK ALGORİTMA (GA)

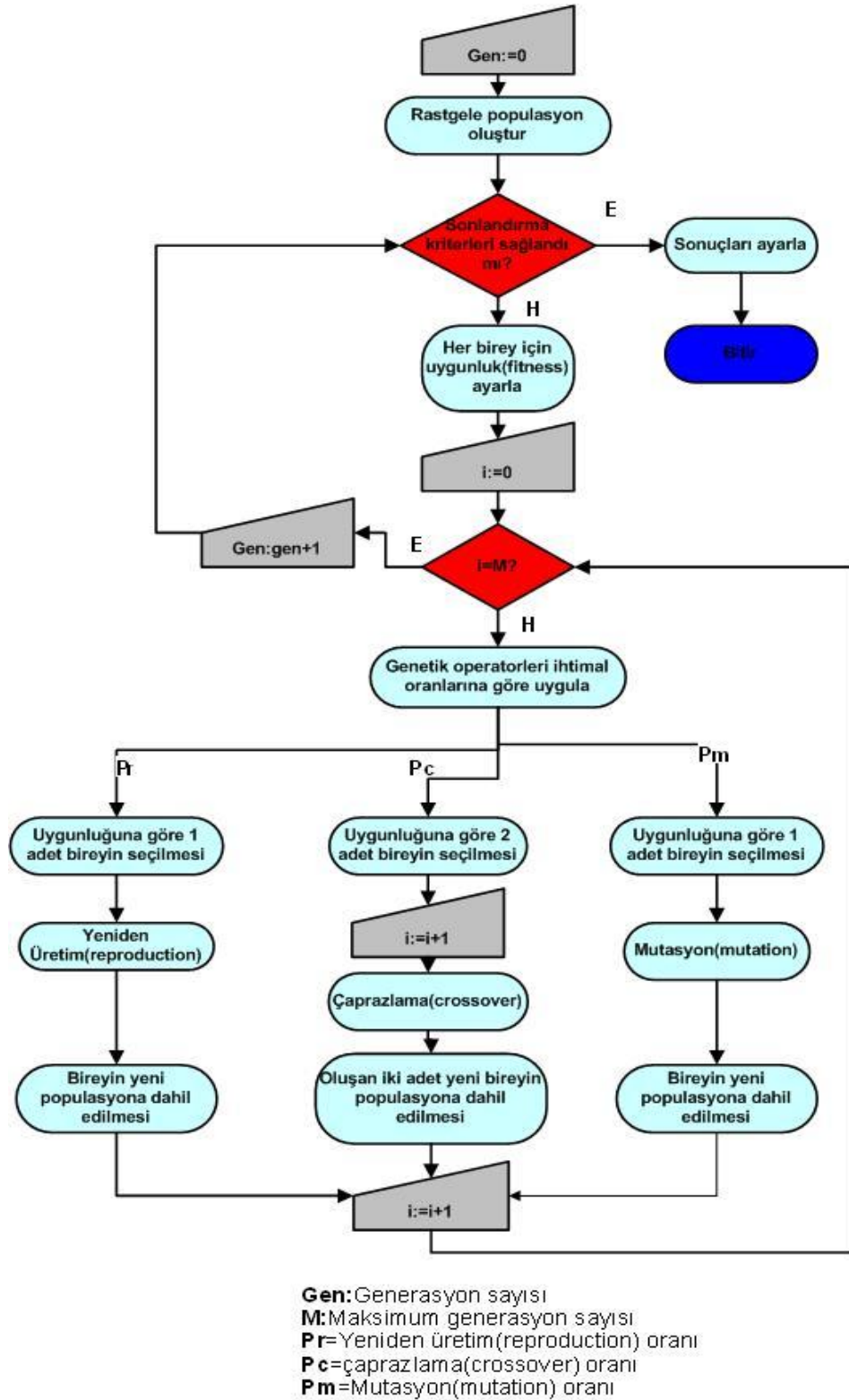
Günümüzün karmaşık ve zor koşulları problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle sert(hard) optimizasyon teknikleri yerine, yumuşak hesaplama (soft computing) ve evrimsel algoritma (evolutionary algorithm) kullanımı ön plana çıkmıştır. Evrimsel yaklaşımlardan olan GA da, bu arayışlar içinde önemli bir yer tutmaya başlamıştır. Uygulama başarıları artan ve sürekli geliştirilmeye çalışılan GA diğer yumuşak hesaplama yöntemleri ile birlikte kullanılarak hibrid (hybrid) çözümler geliştirilmesine çalışılmaktadır. GA, doğal seçim ilkelerine dayanan bir optimal yöntemdir. Temel ilkeleri John Holland tarafından ortaya atılmıştır.

GA, canlıların en iyi olan yaşar prensibini örnek alır ve iyi bireylerin kendi yaşamlarını muhafaza edip kötü bireylerin yok olmasına dayanır. Geleneksel optimizasyon yöntemlerine göre farklılıkları olan genetik algoritma, parametre kümesini değil kodlanmış biçimlerini kullanır. Olasılık kurallarına göre çalışan GA, yalnızca uygunluk fonksiyonuna gereksinim duyar. Çözüm uzayının tamamını değil belirli bir kısmını tarar. Böylece, etkin arama yaparak çok daha kısa bir sürede çözüme [32]. Diğer bir önemli üstünlüğü ise çözümlerden oluşan popülasyonu eş zamanlı incelemeleri ve böylelikle yerel en iyi çözümlere takılmamalarıdır. GA' nın özelliklerinden birisi de bir grup üzerinde çözümü araması ve bu sayede çok sayıda çözümün içinden en iyiyi seçmesidir.

Kısaca GA, rastlantısal arama teknikleri kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir[33].

III.2.1. Basit Genetik Algoritma(SGA)

Pek çok GA uygulamasında SGA(Simple Genetic Algorithm) modeli kullanılır. SGA D.E.Goldberg tarafından tanımlanmıştır ve diğer ardışık genetik algoritma türleri için referans oluşturan bir algoritmadır. SGA' da ikili kodlama biçimi, rulet tekerleği seçim yöntemi, bütünüyle yer değiştirme üreme yöntemi, tek noktadan çaprazlama ve bit mutasyon operatörleri kullanılmıştır [11]. SGA' ya ait akış diyagramı Şekil III.14'de gösterilmektedir.



Şekil III. 14 Basit genetik algoritmanın akış diyagramı

Genel olarak GA, çözüm bilgisinin hiç olmadığı veya çok az olduğu bir durumla aramaya başlar. Çözüm çevreden gelen etkileşime ve genetik operatörlere bağlıdır. GA, aramaya paralel bir şekilde, birbirinden bağımsız noktalardan başlar.

Bu nedenle, GA' nın alt optimal çözümlere takılma olasılığı azdır ve karmaşık arama problemleri “birden çok alt çözüm kümesi olan” için en iyi optimizasyon tekniği olarak bilinir. GA' yı diğer optimal arama algoritma türlerinden farklı kılan özellikleri şunlardır:

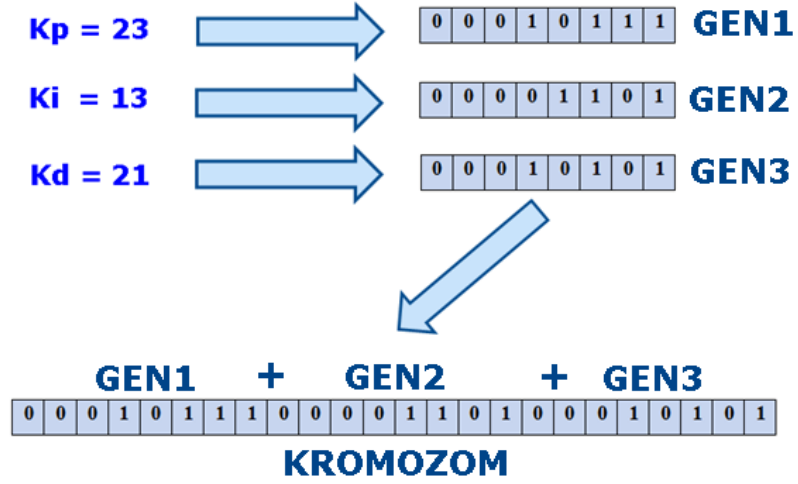
- GA, parametre kodlarıyla uğraşır, parametrelerin kendileriyle doğrudan ilgilenmez.
- GA, tek bir alama bağımlı kalarak çözüm aramaz. Yığının tamamında çözümü arar.
- GA, ne yaptığını değil nasıl yapıldığını bilir. Yani GA amaç işlevini kullanır, sapma değerleri veya diğer hata faktörlerini kullanmaz.
- GA' nın uygulanmasında kullanılan operatörler rastlantısal yöntemlere dayanır, belirli ve kesin yöntem kullanmaz.

III.2.2. Genetik Algoritma Operatörleri

III.2.2.1. Kodlama

GA' yı diğer arama metotlarından ayıran en önemli özellik, GA' nın parametreler yerine parametrelerin kodlanmış şekli ile işlem yapmasıdır. Bu nedenle, GA uygulamasında ilk adım, çözülecek problemin çözüm uzayını en uygun biçimde temsil edecek kodlama yönteminin seçilmesidir.

GA genel olarak, ikili (binary) kodlama biçimi kullanılır. Bu kodlama biçiminde, Şekil III.15'de gösterildiği gibi bütün parametreler ikili kodlanmış bir dizi şeklinde değerlendirilir. Bu tür kodlama bir zorunluluk değildir. Literatürde gray, tamsayı, gerçel sayı veya ağaç biçimini kullanan genetik algoritma türleri olduğu gibi, farklı sembolik alfabeler kullanan GA çalışmaları da mevcuttur[23].



Şekil III. 15 İkili kodlama

Gerçek sayılar gibi karmaşık değerlerin kullanıldığı problemlerde doğrudan değer kodlama kullanılabilir. İkili kodlamanın bu tip problemler için kullanılması problemlerin zorlaşmasına neden olacaktır. Değer kodlamada, her kromozom bazı değerlere eşittir. Değerler problemle ilgili herhangi bir şey olabilir. Gerçek sayılar, karakterler veya herhangi nesneler olabilir[34].

Kodlama biçimi, GA' nın performansını oldukça önemli oranda etkiler, fakat kodlama biçimi probleme bağlı olduğundan bütün problemler için geçerli en uygun kodlama biçimini söylemek imkânsızdır. Z.Michalewicz belli bir problem tipi için yapmış olduğu çalışmada gerçel sayı gösteriminin daha çabuk sonuca ulaştığını göstermiştir[23].

Gerçekleştirilen çalışmada PID kontrolörün katsayıları GA tarafından kullanılırken gerçel kodlama kullanılmıştır.

III.2.2.2. İlk Popülasyonun Oluşturulması

GA' yı diğer sezgisel arama metotlarından ayıran bir özellik; optimal çözümü noktadan noktaya değil, noktaların oluşturduğu yığın içinde aramasıdır. Bu nedenle GA' nın ilk işlemi çözümlerin kodlandığı bir çözüm grubu oluşturmaktır. Çözüm grubu popülasyon, çözümlerin kodları da kromozom olarak adlandırılır. İlk popülasyon genellikle rastlantısal olarak oluşturulur. Ancak bu işlem optimizasyon problemlerinde popülasyonun uygun olmayan çözümlere doğru yönelmesine sebep olabilir. Bu durumu ortadan kaldırmak için probleme özgün, daha önceden elde edilen bir bilgiye veya sezgiye (heuristic) bağlı olarak çeşitli sezgisel yöntemler geliştirilebilir.

III.2.2.3. Popölasyon Büyüklüğü

Popölasyonun büyüklüğü, problemin çözüm süresini etkilemektedir. Fazla sayıda kromozom içeren popölasyon problemin çözüm süresini uzatırken, az sayıdaki popölasyonda ise çözüm değerlerine ulaşılmamasına sebep olabilir veya sistemin belirli çözüm uzayına takılıp iyileşememesine neden olabilir.

III.2.2.4. Üreme

Üreme, mevcut nesilden gelecek nesle aktarılabacak bireylerin(dizilerin) seçilme işlemidir. Bu operatör, bireyle özdeş nitelikteki yeni bireyleri üretmekte ve belirlenen uygunluk değerine sahip iyi bireylerin bir sonraki nesle aktarılmasını sağlamaktadır.

Üreme operatöründe diziler, uygunluk değerine göre kopyalanır ve iyi kalıtsal özellikleri gelecek kuşağa daha iyi aktaracak bireyler seçilir. Üreme operatörü yapay bir seçimdir. Dizileri uygunluk değerlerine göre kopyalama, daha yüksek uygunluk değerine sahip dizilerin, bir sonraki kuşaktaki bir veya daha fazla yavruya daha yüksek bir olasılıkla katkıda bulunması anlamına gelmektedir.

III.2.2.5. Seçim Yöntemleri

Seçim yöntemleri, genel olarak en iyi olan yaşar prensibine göre oluşturulur. Amaç, yeni generasyonda daha yüksek uygunluğa sahip bireylerin sayısını arttırmaktır. Bir nesildeki bireylerden bir kısmının bir sonraki nesle aktarılırken, bir kısmı da yok olur. Bu durumda hangi bireyin bir sonraki nesle aktarılacağı seçim yöntemleri ile sağlanır. GA’ da seçim yöntemleri 3 grupta toplanır[35, 36].

Bunlar:

- Uygunluk Değeri Orantılı Seçim Yöntemleri
- Turnuva Seçim Yöntemi
- Sıralı Seçim Yöntemi

1.a Rulet Tekniğı Seçim Yöntemi

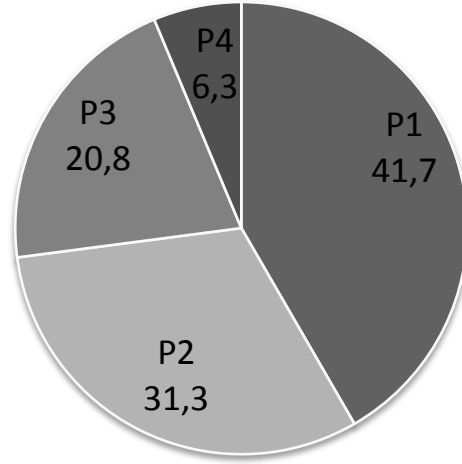
Bu seçim yöntemi uygunluk değeri orantılı seçim yöntemine girmektedir. Bu seçim yönteminde, bireyin beklenen seçim olasılığını, bireyin uygunluk değerinin generasyonun toplam uygunluk değerine bölümü belirler. Yöntem, genel olarak rulet oyununa benzetilir. Generasyondaki bireylerin toplam uygunluk değeri rulet tekerleğinin büyüklüğünü verir ve her bir birey rulet tekerleğinde uygunluk değeri kadar yer kaplar[37]. Tekrar üreme için rulet tekerleğinin döndürülmesi gerekir.

Bunun için sıfır ile toplam uygunluk değeri arasında rastgele bir sayı üretilerek bu sayının tekerleğin hangi parçasına denk geldiğine bakılarak birey seçilir. Böylelikle çemberin bir defa döndürülmesiyle sonraki nesle aktarılacak olan bireylerden bir tanesi seçilmiş olur. Benzer şekilde diğer bireylerinde belirlenmesi ile uygunluk değeri en iyi olan bireyler eşleştirme havuzuna(mating pool) alınır. Bu işlem sonra diğer nesle ait diziler elde edilir ve genetik operatörlerin uygulanmasıyla yeni nesil elde edilir. Aynı işlem her döngüde devam ederek nesil devamı sağlanır[38].

GA ile oluşturulan bir popülasyonda 4 birey olsun. Bu bireylerin uygunluk değerleri sırasıyla Φ_k ; 20, 15, 10 ve 5 ($k=1,2,3,4$) olsun. Her bir birey için seçilme olasılığı veya yaşama olasılığı, p_k Denklem III.14'den elde edilebilir.

$$p_k = \frac{\Phi_k}{\sum_{j=1}^n f_j}, k = 1,2,3,4 \quad (\text{III.14})$$

formülden elde edilen seçilme olasılığı sırasıyla $p_1=\%41,7$; $p_2 = \%31,3$; $p_3 = \%20,8$; $p_4= \%6,3$ olarak bulunur. Şekil III.16'da her bir bireyin yüzdesine göre oluşturulmuş çember görülmektedir.



Şekil III. 16 Rulet Tekerleği seçim yöntemi

Seçim işlemi için Rulet Tekerleği kullanılır. Rulet çevrilir, hangi bireyin üzerinde durur ise o birey seçilecektir. Popülasyon içersindeki bireyler Tablo III.1' de gösterildiği gibi 0 ile 1 değerleri arasında bir değere denk gelecek şekilde bir çevrilme işlemi yapılır. r_u , 0 ile 1 arasında rastgele üretilen bir sayıdır. Üretilen bu sayı ile Rulet tekerleği üzerinde bir birey seçilir. Bu sistemin dezavantajı ise uygunluk derecesi yüksek bir bireyin seçilmeyip yok olma ihtimali mevcuttur[23].

Tablo III. 1 Bireylerin uygunluk değerlerinin bilgisayar ortamına çevrilmesi

Birey sayısı	Birey seçilme değeri (p_k)	Bireyin alt ve üst sınırları
1	%40	$0.0 \leq r_u \leq 0.4$
2	%30	$0.4 < r_u \leq 0.7$
3	%20.8	$0.7 < r_u \leq 0.9$
4	%10	$0.9 < r_u \leq 1.0$

1.b Turnuva Seçim Yöntemi

Bu seçim yönteminde, bireyler rastgele olarak gruplanır ve gruptaki bireyler aralarında seçim işlemi için rekabete sokulur. Grup içinde, en yüksek uygunluk değerine sahip olan birey, bir sonraki generasyonda popülasyonu oluşturmak için ebeveyn olarak seçilmiş olur. Bu işlem, toplam popülasyon oluşturuluncaya kadar devam eder. Bu seçimde grup büyüklüğü performansı etkilemektedir. Turnuva seçim yöntemi deterministiktir. Küçük popülasyon kullanan uygulamalarda uygunluk değeri orantılı seçim yöntemlerinden daha iyi sonuç verir[39]. Bireyin seçilme olasılığı Denklem III.15’ de verilmiştir.

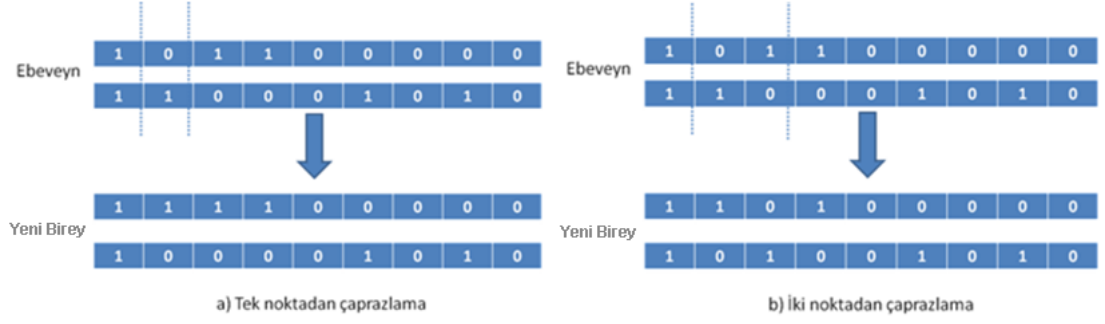
$$p_i = \frac{1}{N^{q8}} ((N - i + 1)^q - (N - i)^q) \quad (\text{III.15})$$

Burada; N = popülasyon büyüklüğü, p_i , bireyin seçilme olasılığı, q= turnuva büyüklüğü olarak verilir.

III.2.2.6. Çaprazlama

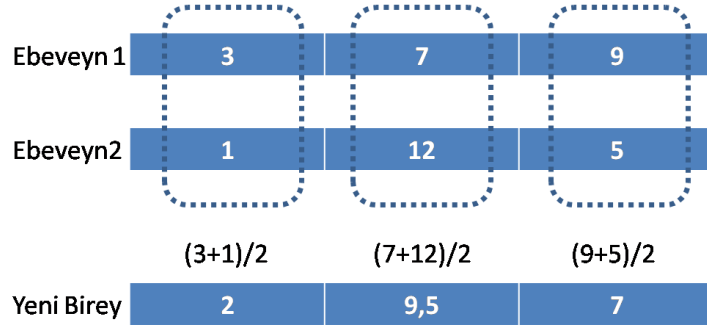
Çaprazlama operatörünü seçilen iki ebeveynin genlerinin rastgele seçilen bir noktadan bir bütün oluşturacak şekilde değiş tokuş edilmesi olarak açıklayabiliriz. Çaprazlama operatörü seçilen kodlama türüne göre çeşitlilik göstermektedir.

İkili kodlamada genel olarak çaprazlama noktası bir tanedir. Ancak çaprazlama noktası birden fazla seçilebilmektedir. Şekil-III.17’de çaprazlama operatörünün bir noktada ve iki noktada uygulanarak oluşturulan yeni nesiller görülmektedir. Çaprazlama olasılığı (P_c) ise bir popülasyon içindeki ebeveynlerin yeni bireyler üretmek için hangi olasılıkla çaprazlanacağını belirtir. Örnek olarak ($P_c=1$) alınırsa, bütün ebeveynlerin çaprazlanacağını gösterir.



Şekil III. 17 Çaprazlama operatörü

Gerçek kodlamanın kullanıldığı GA optimizasyonlarında genellikle aritmetik çaprazlama kullanılır. Aritmetik çaprazlamada ise yeni bireyler, Şekil III.18’de gösterildiği gibi ebeveynlerinin genlerinin aritmetik ortalaması alınarak üretilirler. Oluşturulan bireyler sınırlara ve kısıtlara uygun olarak oluşturulur.

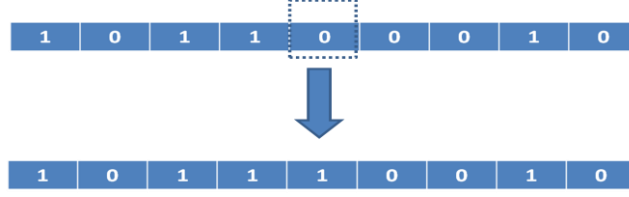


Şekil III. 18 Aritmetik çaprazlama

III.2.2.7. Mutasyon(Değişim)

GA’ da sistem, belirli bir döngü değerine geldikten sonra bireyler birbirlerine gitgide benzemektedir. Buda durum uzayının daralmasına sebep olmaktadır. Bireylere ne kadar çaprazlama operatörü uygulansa da birey çeşitliliği sağlanamamaktadır. Bu durumda, bireyi oluşturan bir dizi içindeki bir veya daha fazla gen rastlantısal olarak değiştirilir. Mutasyon operatörü, küçük olasılık değerinde kullanılan bir genetik algoritma operatörüdür. Çaprazlama operatöründe olduğu gibi mutasyon operatöründe de seçilen kodlama türüne göre değişmektedir.

İkili kodlamanın kullanıldığı GA optimizasyonunda, mutasyon işlemi sırasında bir gen alınarak değeri 1 ise 0 ve 0 ise bir olacak şekilde değişikliğe uğratılır. Şekil III.19’da mutasyon operatörünün popülasyon içindeki bir bireyi oluşturan, ikili kodlanmış gen dizisi içindeki rastgele bir geni değiştirmesi görülmektedir



Şekil III. 19 Mutasyon operatörü

Eğer GA optimizasyonu için parametreler gerçek kodlama ile kodlanmış ise gaussian mutasyon işlemi kullanılabilir. Gaussian mutasyon işleminde gen mutasyon için seçildiğinde, gaussian dağılımına göre bir değer seçilir ve seçilen değer genin üzerine eklenir. Gaussian eğrisinin genişliği bu genin değerinin pozitif ve negatif değerleridir. Böylece gaussian mutasyon işlemi, bir gen için uygulandığında mutasyon oranının gaussian eğrisi üzerindeki değerine göre genin değeri azaltılabilir veya artırılabilir.

III.2.3. Kısıtlı Optimizasyon Problemlerinin Çözümü

GA' nın kısıtlı optimizasyon problemlerinde karşılaşılan en önemli zorluk, klasik genetik operatörlerinin uygulanmasıyla kısıtları sağlamayan dizi yapılarının elde edilmesidir. Bu tür problemlerde uygun çözümlerin bulunduğu popülasyonlar elde edilse bile, çaprazlama ve değişim operatörlerin uygulanmasından sonra yeni çözümlerin elde edilmesi kolay değildir. Bu amaçla, kısıtlı optimizasyon problemlerinin genetik algoritma ile çözümü için farklı yaklaşımlar geliştirilmiştir[23]. Burada kullanılabilecek üç temel yaklaşım şunlardır:

1. Düzenleyici Algoritma yaklaşımı,
2. GENOCOP (Genetic Algorithms for Numerical Optimization for Constrained Problems),
3. Ceza fonksiyonu (Penalty Function) yaklaşımı.

III.2.3.1. Düzenleyici Algoritma

Bu yöntemde, uygun olmayan çözümleri sağlayan dizileri uygun diziler şekline dönüştürmek için düzenleyici algoritmalar kullanılır. Bu yöntemin en büyük dezavantajı, her problem için, probleme özgü düzenleyici algoritmaya ihtiyaç duyulmasıdır. Problemin özelliğine göre geliştirilen bu algortmada genetik operatörlerin uygulanmasından sonra diziden mevcut bilgilerin yok olması veya istenmeyen bilgilerin gelmesi durumunda ilk dizi bilgilerine sadık kalarak özel bir algoritma ile dizide istenmeyen bilgilerin düzeltilmesi gerçekleştirilmektedir.

III.2.3.2. GENOCOP

Bu yaklaşım Z.Michalewicz tarafından geliştirilmiş ve GENOCOP programına dayanır [36]. Bu yaklaşımda, kısıtlı optimizasyon problemlerini çözebilmek için farklı genetik operatörler kullanılmıştır. Burada kullanılan farklı genetik operatörlerden bazıları, Tekdüze mutasyon (Uniform mutation), Kısıtlı mutasyon (Boundary mutation), Tekdüze olmayan mutasyon (Non-uniform mutation), Sezgisel çaprazlama (Heuristic crossover), Aritmetik çaprazlama (Arithmetic crossover) dır.

III.2.3.3. Ceza fonksiyonu

Ceza fonksiyonu yaklaşımı; bu yaklaşımda kısıtlar dikkate alınmaksızın mümkün çözümler elde edilir. Kısıtları bozan dizilerin uygunluk değeri düşürülerek cezalandırılır. Cezalar, değerlendirme işlevi içinde yer alır. Ceza işlevi yaklaşımı kısıt sayısı az olan problemler için uygun olduğunu savunan Michalewicz ve Janikov, kısıt sayısı fazla olan problemler için yeni genetik operatörlerin geliştirilmesinin GA'nın etkinliğini artıracaklarını belirtmiştir.

Burada kullanılacak ceza fonksiyonu aşağıdaki gibi tanımlanabilir;

$$\Phi'(x) = \Phi(x) \quad ; \text{eğer } x \text{ hiçbir kısıdı ihlal etmemişse} \quad (\text{III.16})$$

$$\Phi'(x) = \Phi(x) + \varphi(x) \quad ; \text{eğer } x \text{ kısıdı ihlal etmişse} \quad (\text{III.17})$$

Burada; Φ , uygunluk fonksiyonu, φ ceza fonksiyonudur.

Kısıtlar ihlal edildiğinde atanan cezalar genellikle kısıdı ihlal derecesine bağlıdır. Buna göre, hiçbir kısıt ihlal edilmemişse ceza fonksiyonun değeri sıfır olacaktır.

III.2.4. GA ile Optimizasyon

GA araştırmalarının önemli bir bölümü fonksiyon optimizasyonu ile ilgilidir. GA, geleneksel optimizasyon tekniklerine göre zor, süreksiz ve gürültü içeren fonksiyonları çözmede daha etkindirler. Optimize edilecek uygunluk fonksiyonunun süreksiz olması halinde, süreksizlik noktalarında fonksiyonun türevi alınamayacağından, türev almaya dayalı optimizasyon yöntemleri kullanılamamaktadır. Oysa GA, problemlerin çözümü için türev veya diğer yardımcı bilgilere gereksinim duymadığından özellikle bu tip problemlerin çözümünde geleneksel yöntemlere göre önemli bir üstünlük sağlamaktadır[23].

Kontrol uygulamalarında GA' nın kullanımının önemli bir bölümünü ise kontrolör parametrelerinin oluşturduğu geniş arama uzayından, optimum kontrolör parametrelerinin bulunmasına yönelik yapılan çalışmalar oluşturur[23]. Gerçekleştirilen çalışmada, sistemi verilen referansı optimum şekilde kontrol eden PID kontrolörün katsayıları GA ile bulunmuştur.

III.3. KARINCA KOLONİSİ OPTİMİZASYONU(KKA)

KKA algoritmaları model tabanlı arama algoritmaları sınıfındandır. Bu sınıf ayrık optimizasyon problemlerini çözmede gittikçe artan popüleriteye sahip algoritma sınıfıdır. KKA teknikleri, optimizasyon problemlerini çözmek için karıncaların sosyal davranışını modeller. Karınca algoritmaları, gerçek karıncaların davranışının gözlenmesinden türetilen modelleri inceler[32].

Bazı problemleri çözmenin tek yolu cevapla ilgili bütün ihtimallerin tek tek test edilmesi ile gerçekleşir. Buna verilebilecek klasik bir örnek, şehir şehir dolaşan bir işadaminin izlemesi gereken en kısa rotanın bulunmasını ele alan gezgin satıcı problemidir. Bu tür problemler arkalarında iz bırakarak ilerleyen karıncaların uyguladığı yöntemle çözülebilir. 1989 ve 1990 yıllarında, ABD' nin Mexico eyaletindeki Santa Fe Enstitü'sünde Bonabeau ve arkadaşları "sanal karıncalar" oluşturarak benzer problemlerin bilgisayarlarla daha kolay çözülebileceğini gösterdiler. Buna göre sanal karıncalar arkalarında buldukları rotanın uzunluğunu da simgeleyen bir nevi koku izi bırakacak ve diğer sanal karıncalar da kestirme rotaları bu sayede bularak tercih edeceklerdir. Koku izinin belirli bir hızda buharlaşması da simüle edilerek tercih edilmeyen uzun rotalardaki koku izleri de yavaş yavaş yok olacak ve bu da sanal karıncaların kestirme yol dışındaki uzun yollara sapmasını önleyecektir[40].

Doğal karıncaların bu davranış kalıplarının, özellikle en kısa yol problemleri olmak üzere, pek çok optimizasyon probleminde kullanılabileceği ilk kez 1992 yılında Marco Dorigo ve arkadaşları tarafından ortaya atılmıştır[41].

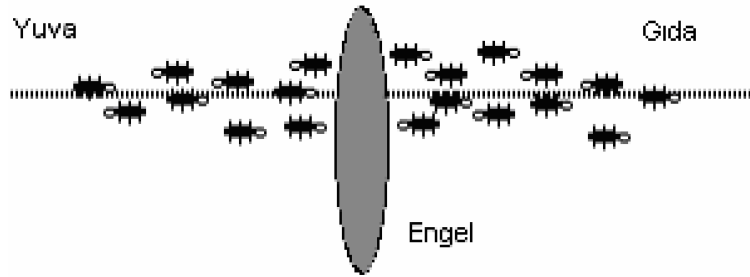
III.3.1. Gerçek Karıncalar

Karıncalar, yuvalarından bir gıda kaynağına giden en kısa yolu, herhangi görsel bir ipucu kullanmadan bulma yetisine sahiptirler. Ayrıca, çevrelerindeki değişikliklere uyum sağlama özellikleri vardır. Örnek vermek gerekirse, bir gıdaya giden yolda herhangi bir problem meydana gelmesi (bir engelin ortaya çıkması gibi) veya yolun kullanılamaz olması durumunda, yeniden en kısa yolu bulurlar. Şekil III.20'de karıncaların yuvalarından doğrusal bir yol boyunca gıdaya gidişlerini göstermektedir.



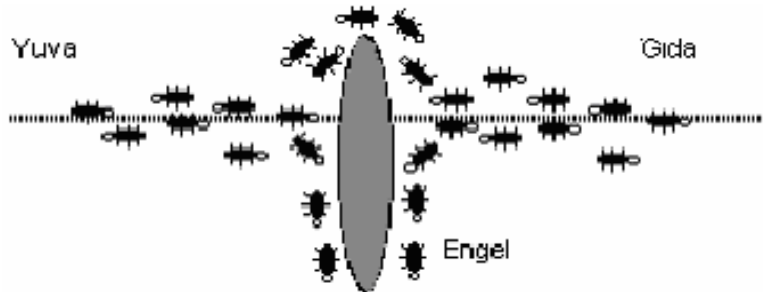
Şekil III. 20 Karıncaların izlediği yol

Karıncaların bu yolu bulmak için kullandıkları araç feromonlardır. Karıncalar ilerlerken belirli bir miktar feromon depo ederler ve olasılığa dayanan bir yöntemle feromonun daha çok olduğu yolu az olduğu yola tercih ederler. Depo ettikleri feromonları, gıdaya giderken seçtikleri yola bırakarak, kendilerinden sonraki karıncalara yol seçiminde yardımcı olurlar. Bu içgüdüsel davranış, onların gıdaya giden en kısa yolu, önceden var olan bir yolun kullanılamaz olması durumunda dahi nasıl bulduklarını açıklar. Gerçekte, gıdaya giden yolda Şekil III.21’de gösterildiği gibi herhangi bir engel meydana geldiğinde, bu engelin hemen önündeki karınca devam edemez ve yeni gidiş yönü için tercih yapmak zorunda kalır.



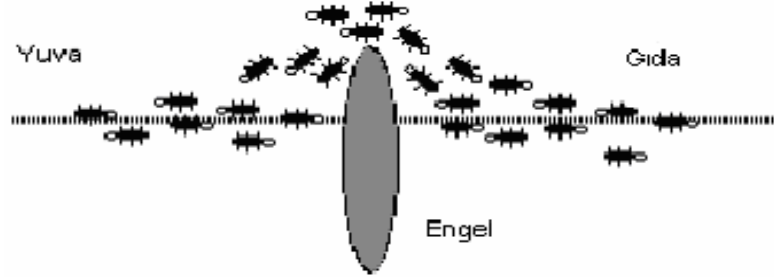
Şekil III. 21 Karıncaların bir engelle karşılaşması

Bu konumda, karıncanın yeni yön seçeneklerinin seçilme olasılıkları eşittir. Yani karınca sağ ve sol yönlerinden birisini seçebiliyorsa, bu yönlerin seçilme şansları eşittir. Karınca yaptığı seçime göre yoluna devam eder ve kendi yolunu Şekil III.22’de gösterildiği gibi çizer.



Şekil III. 22 Engelle karşılaşan karıncaların seçimi

Buradaki ilginç noktalardan bir tanesi de, kolonide engel karşısında yol için seçim yapan karıncaların seçtikleri yolun gıdaya giden en kısa yol olmaması durumunda, güzergâhlarını çok hızlı bir biçimde Şekil III.23’ de gösterildiği gibi yeniden yapılandırabilmeleridir.



Şekil III. 23 Karıncaların kısa yolu bulmaları

Yapılan seçimler de, bu yol üzerindeki feromen miktarını artıracak ve sonradan gelen karıncalar için tercih sebebi olacaktır. Sonradan gelen karıncaların, yeni en kısa yolu seçmelerindeki feromen pozitif etkisinin oluşabilmesi için, karınca ile yol üzerindeki engel arasındaki etkileşim hızlı gerçekleşmelidir. Her karıncanın, ortalama aynı hızda ve aynı miktarda feromen bıraktığı göz önüne alınırsa, karıncanın engeli fark edip en kısa yolu seçmesi normal süreçten uzun sürmelidir. Fakat sonradan gelen karıncaların feromene dayalı yol seçimi toplamda gıdaya giden süreci Şekil III.23’de gösterildiği gibi kısaltır[14].

III.3.2. Yapay Karıncalar

Gerçek karıncalar, kör olmalarına rağmen yuvalarından yiyeceğe giden en kısa yolu bulabilmektedir. Karıncaların bu özelliği, bir takım özellikler aynen kullanılarak ve bazı eklemelerle gerçek problemlerin çözümünde kullanılabilir hale getirilmiştir.

Yapay karıncalar oluşturulurken gerçek karıncalardan alınan özellikler:

- Karıncalar arasında feromen aracılığı ile kurulan iletişim,
- Feromeni fazla olan yolların öncelikle tercih edilmesi,
- Kısa yollarda feromen miktarının daha hızlı artması.

Gerçek karıncalara eklenen özellikler:

- Zamanın ayırık olarak hesaplandığı ortamda yaşarlar,
- Tamamen kör olmayıp, problem ile ilgili detaylara erişebilirler,
- Belli bir miktar hafıza ile problemin çözümü için oluşturdukları bilgileri tutabilirler.

Karınca tabanlı algoritmelerde temel fikir, basit iletişim mekanizmalarını kullanan yapay akıllı araçların, birçok karmaşık problem için çözümler üretebilmesidir[32].

III.3.3. Karınca Kolonisi Algoritması

Karınca Kolonisi Optimizasyonu Algoritması, yapı olarak paralel çalıştırılmaya çok uygundur. Koloni yapısında, birden fazla *asen kron veya paralel* çalışan karınca vardır. Paralel karınca koloni sisteminde birden fazla düğümün olduğu ve her bir karıncanın belirli bir anda bir düğümde çalıştığı varsayılırsa, karıncalar paralel sistemlerdeki akıllı araçlar olarak nitelendirilebilirler. Gezgini satıcı probleminde karıncalar, rastgele seçilen bir şehirden başlayarak bütün şehirleri ziyaret ederler ve geçtikleri yollara feromon bırakırlar. Bu feromonlar, sonraki karıncaların yollarını belirlemelerinde etkili olur, yani araçlar arasındaki iletişim, ortak sonucu doğurur. Karıncaların her birini bir düğüme gönderirken dikkat edilmesi gereken faktör, yarışma şartlarının meydana gelmesini engellemektir. Yarışma şartları ile kastedilen, aynı anda çalışan karıncaların, yani farklı düğümlerde çalışan karıncaların, feromon ve seçilen yol verilerini tutan global veri yapısını aynı anda değiştirememelidirler. Bu uygulamada temel amaç, KKA algoritması içerisinde olasılık hesabında kullanılan parametrik alfa (α), beta (β), ro (ρ) değerlerinin en uygunlarını bulmak olduğu için kolonideki karınca sayısının artırılması, bu değerlerin en uygunlarını bulma olasılığını arttıracaktır[14].

Problemin başlangıcında her karınca farklı veya aynı köşelere yerleştirilir. Bu karıncalar (t) anında hangi komşu düğüm noktasında olacaklarını (III.18) numaralı bağıntıya (olasılık bağıntısı) göre belirlenir.

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{i \in N_i} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta} & \left(\begin{array}{l} k \text{ izin} \\ \text{verilen} \\ \text{bir seçimse} \end{array} \right) \\ 0 & (\text{aksi halde}) \end{cases} \quad (\text{III.18})$$

$\tau_{ij}(t)$: t anında (i,j) köşelerindeki feromon iz miktarı.

η_{ij} : (i,j) köşeleri arasındaki görünürlük değeridir. Bu değer problem çözümünde ele alınan kritere göre değişmektedir.

α : Problemde feromon ize verilen bağıl önemi gösteren bir parametredir.

β : Problemde görünürlük değerine verilen önemi gösteren bir parametredir.

N_i : Henüz seçilmemiş olan düğüm noktaları kümesidir.

Karıncalar bu olasılık bağıntısına göre bir sonraki seçimlerini yaparlar. Problemdaki tüm düğüm noktaları gezildikten sonra bir tur veya iterasyon tamamlanmıştır. Bu noktada (III.19) numaralı bağıntıya göre feromen iz miktarı güncellenir [42].

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (III.19)$$

p : t ile $t+n$ süreleri arasında buharlaşan feromen iz oranı ($0 < p < 1$)

$\Delta\tau_{ij}$: Karıncanın bir turu boyunca (i,j) köşesini seçmelerinden dolayı bu köşedeki feromen iz miktarını gösterir. Bu miktar (III.20) numaralı bağlantıya göre hesaplanır.

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (III.20)$$

m : Toplam karınca sayısı

$\Delta\tau_{ij}^k$: k . karıncanın (i,j) köşesine bıraktığı feromen iz miktarı

(III.20) numaralı bağıntı ise her bir (k) karıncasının herhangi bir (i,j) köşesindeki feromen iz miktarına ne kadarlık katkı yapacağını gösterir [10].

$$\Delta\tau_{ij}^k = \left\{ \frac{Q}{L_k} \right\} \quad (III.21)$$

Q : Sabit bir değerdir.

L_k : k . karıncanın tur uzunluğu.

Eğer karınca tur boyunca (i,j) köşesini kullandı ise (III.21) ifadeye göre bıraktığı iz miktarı hesaplanır. Aksi halde iz miktarı sıfır olur[43].

III.3.3.1. Feromen Buharlaşması

Gerçek karınca kolonilerinde, feromen yoğunluğu buharlaşmadan ve bozulmadan dolayı azalmaktadır. KKA' da buharlaşmanın etkisi, uygun şekilde tanımlanmış bir buharlaşma kuralının uygulanmasıyla simule edilmektedir. Örneğin, feromenin yapay gecikmesi sabit bir oran olarak ayarlanabilir. Feromen buharlaşması, araştırmanın erken safhalarındaki yapay karıncaların oluşturduğu düşük - kaliteli çözümlerde bırakılmış olan feromenlerin etkisini azaltır. Feromen buharlaşması gerçek karıncalarda fark edilir bir etki yapmasa da, yapay karınca kolonilerinde çok yararlı olabilmektedir.

BÖLÜM IV

TEZ ÇALIŞMALARI

Bu bölümde, ilk olarak Gunt RT 532 Basınç Kontrol Prosesinin YSA ile modellenmesi gerçekleştirilecektir. İkinci olarak sırasıyla GA, KKA ve ZN ile PID kontrolörün optimizasyonu yapılacak ve elde edilen Genetik-PID, Karınca-PID ve ZN-PID kontrolörlerinin performansları incelenecektir. Yapılan çalışmada, sırasıyla birim basamak girişte, basamak girişler arasında geçişte, bozucu etkide, yörünge takibinde ve yörüngeler arasında geçişte sistemin davranışları incelenecektir.

IV.1. SİSTEMİN YSA ile MODELLENMESİ

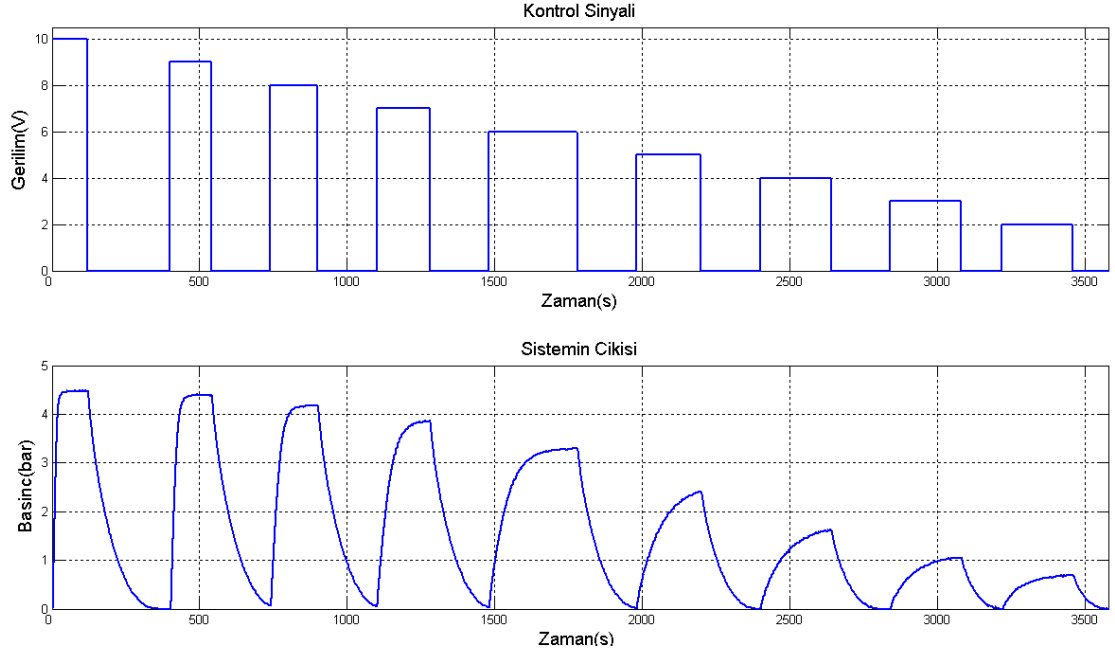
Bu çalışmada; YSA, hava girişi ve debi oransal valf den oluşan basınç tankını modellemede kullanılmıştır. Sisteme verilen kontrol işaretine karşılık basınç tankında oluşan hava basıncının davranışını modellemek amacıyla zaman serisi modellemeye dayanan seri-paralel YSA modeli kullanılmıştır.

IV.1.1. Kullanılan YSA Modeli

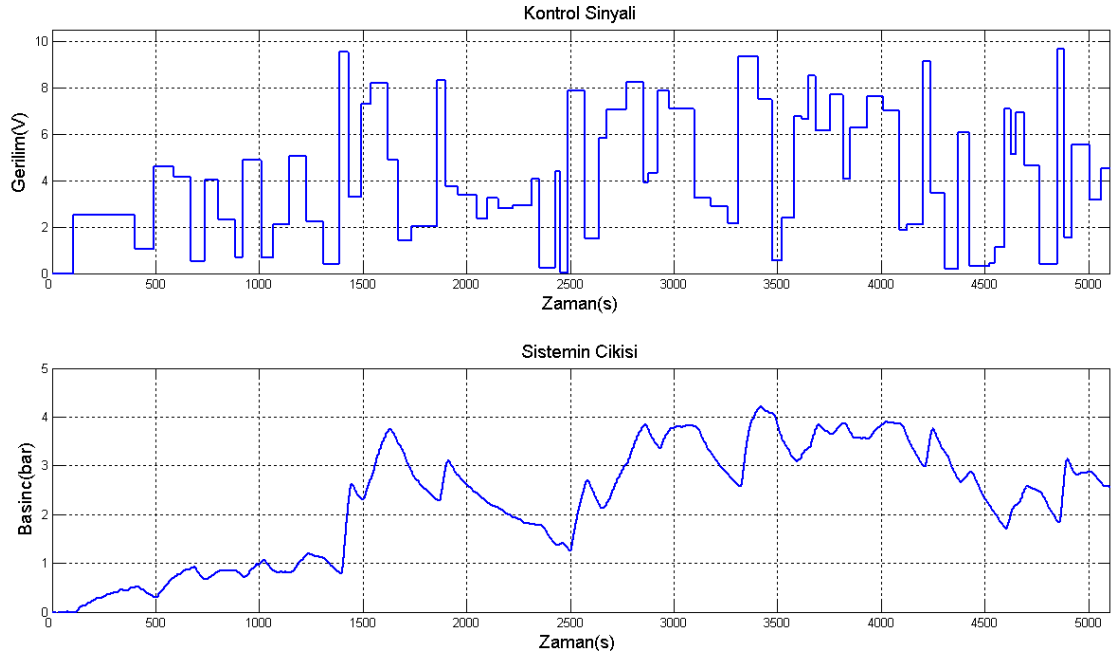
Sistemin modellenmesi, Şekil III.13'de gösterilen blok diyagramında gösterildiği gibi seri-paralel modelleme kullanılarak gerçekleştirilmiştir. Seri-paralel modelleme sisteminde; ana ilke olarak, sistem çıkışı ve onun geriye kaymış modları YSA modeline giriş olarak sağlanmaktadır. Yapılan çalışmada 4 adet zaman gecikmesi(D) kullanılmış ve YSA' nın önceki 4 giriş ve önceki 4 çıkış, ağırlık girişi olarak uygulanmıştır.

IV.1.2. Eğitim Verisinin Toplanması

YSA 'da giriş veri seti olarak basınç tankının açık çevrimde, Şekil IV.1'de gösterildiği gibi 0-10V arasında değişen basamak girişler ve Şekil IV.2'de gösterildiği gibi rastgele kontrol girişleri uygulanarak elde edilen basınç bilgileri kullanılmıştır. YSA eğitiminde, bu iki veri setinin birleştirilmesi ile elde edilen giriş çıkış bilgileri kullanılmıştır.



Şekil IV. 1 YSA eğitimi için uygulanan basamak girişler ve sistemin cevabı



Şekil IV. 2 YSA eğitimi için uygulanan rastgele girişler ve sistemin cevabı

IV.1.3. Verinin Normalleştirilmesi

Verinin normalleştirme işlemi, veri setini oluşturan sayıların nümerik hesaplama işlemlerinde doğruluğu artırmak için yapılan bir ölçeklemedir. Bu işlem, YSA'nın eğitimi için önemli bir aşamadır. Ayrıca normalleştirme aktivasyon fonksiyonunun şekline de yardımcı olur. Bu nedenle normalleştirme işlemi için Denklem IV.1' deki bağıntı kullanılmıştır.

$$\frac{y - y_{min}}{y_{max} - y_{min}} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (IV.1)$$

Burada; y çıkış değerini, y_{min} çıkışın minimum değerini, y_{max} çıkışın maksimum değerini, x çıkış değerini, x_{min} çıkışın minimum değerini, x_{max} çıkışın maksimum değerini göstermektedir.

IV.1.4. YSA Mimarisinin Seçilmesi

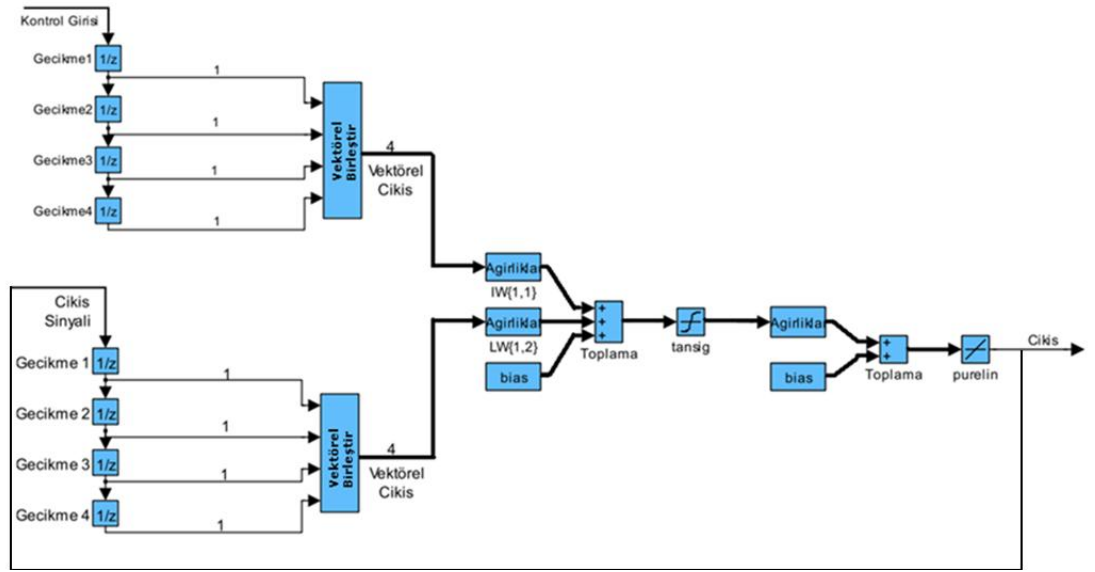
YSA modelini bulmaya yönelik yapılan deneylerin sonuçları Tablo IV.1'de gösterilmiştir. Tabloda, tek ve çift olmak üzere iki farklı sayıda gizli katman, gizli katmanda 5 ile 50 arasında değişen nöron sayısı ve eğitim sırasında farklı hata parametreleri seçilerek oluşturulan YSA modellerinin hata değerleri ve regresyon değerleri gösterilmiştir. Sisten için en uygun YSA modelinin bulunması için tek ve çift olmak üzere iki farklı gizli katman ve gizli katmanda 5, 10, 20, 30 ve 50 olmak üzere farklı nöron sayısı kullanılmıştır. Ayrıca elde edilen YSA modelleri, sırasıyla Karesel Hata Ortalaması (MSE-Mean Square Error), Düzenleştirilmiş Karesel Hata Ortalaması (MSEREG-Mean Square Error with Regularization) ve Karesel Hatların Toplamı (SSE-Sum Square Error) olmak üzere 3 farklı YSA performans fonksiyonları kullanılarak değerlendirilmiş ve bu sonuçlar Tablo IV.1' de gösterilmiştir.

Tablo IV. 1 YSA mimarisi için yapılan deney sonuçları

Katman Sayısı	Nöron Sayısı	MSE	Regresyon Değeri	MSEREG	Regresyon Değeri	SSE	Regresyon Değeri
Tek Katman	5	0.00000099	0.998551	0.101305	0.997941	0.0104193	0.997658
	10	0.00000082	0.997774	0.00807202	0.998901	0.0130032	0.996693
	20	0.00000043	0.997806	0.00748404	0.999241	0.00833915	0.997792
	30	0.00000037	0.997765	0.127915	0.999338	0.00830849	0.997797
	50	0.00000036	0.997755	0.0769577	0.999173	0.00875246	0.997955
Çift Katman	5	0.00000582	0.998939	0.154377	0.998857	1.62084	0.998653
	10	0.00000036	0.997796	0.256829	0.998615	0.00906618	0.99781
	20	0.00000041	0.997684	0.0838121	0.999024	0.00830944	0.997808
	30	0.00000035	0.99784	0.0431946	0.999227	0.0080064	0.997787

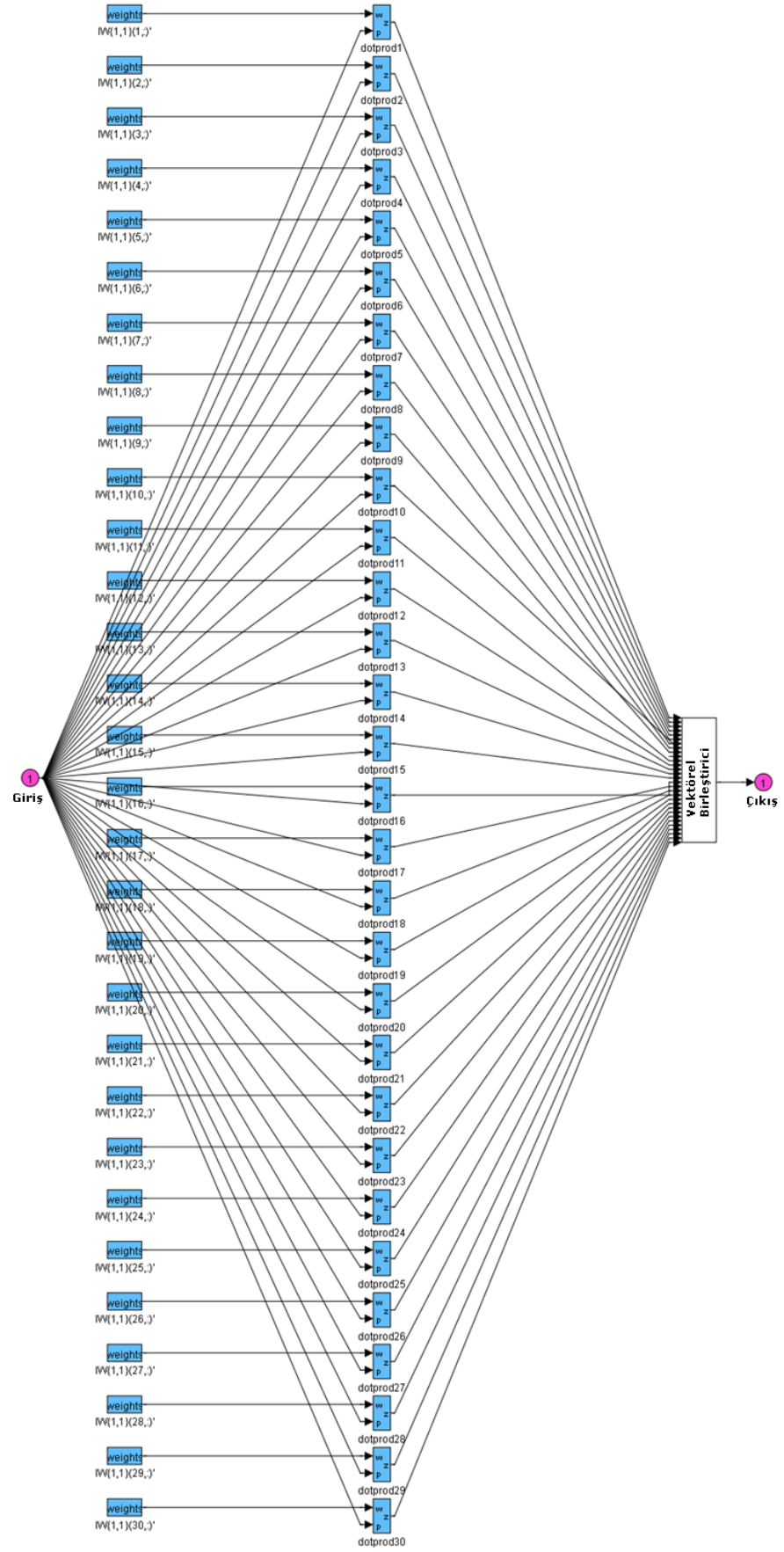
Tablo IV.1’den anlaşılabacağı gibi, bir adet gizli katmandan oluşan ve gizli katmanda 30 adet nöron bulunan, ileri yönlü besleme modeline dayanan bir YSA yapısı kullanılmıştır. YSA’ nın eğitimi için geriye doğru hata yayılma algoritması ve hatanın karelerinin ortalaması (MSE) değerini kullanan YSA performans fonksiyonu ile eğitim gerçekleştirilmiştir.

Oluşturulan YSA ‘nın giriş sayısı 2 ve çıkış sayısı 1 adettir. Giriş katmanında aktivasyon fonksiyonu olarak çıkışı -1 ve 1 aralığında değişen hiperbolik tanjant fonksiyonu ve çıkış katmanında ağı net girdisini doğrudan çıkış olarak veren doğrusal aktivasyon fonksiyonu seçilmiştir. Her katmanın çıkış değeri bir sonraki katmanın giriş değeridir. Bu şekilde giriş değerlerinin ağı girişinden çıkışına doğru ilerlemesi ile oluşturulan ileri beslemeli yapının blok diyagramı Şekil IV.3’ de gösterilmektedir. YSA’ nın girişinde bulunan 4 adet zaman gecikmesi ile $(1/z)$ uygulanan kontrol sinyalinin ve sistem çıkışının önceki 4 adet bilgisi ağa girişi olarak uygulanmaktadır.



Şekil IV. 3 Kullanılan YSA modelin blok diyagramı

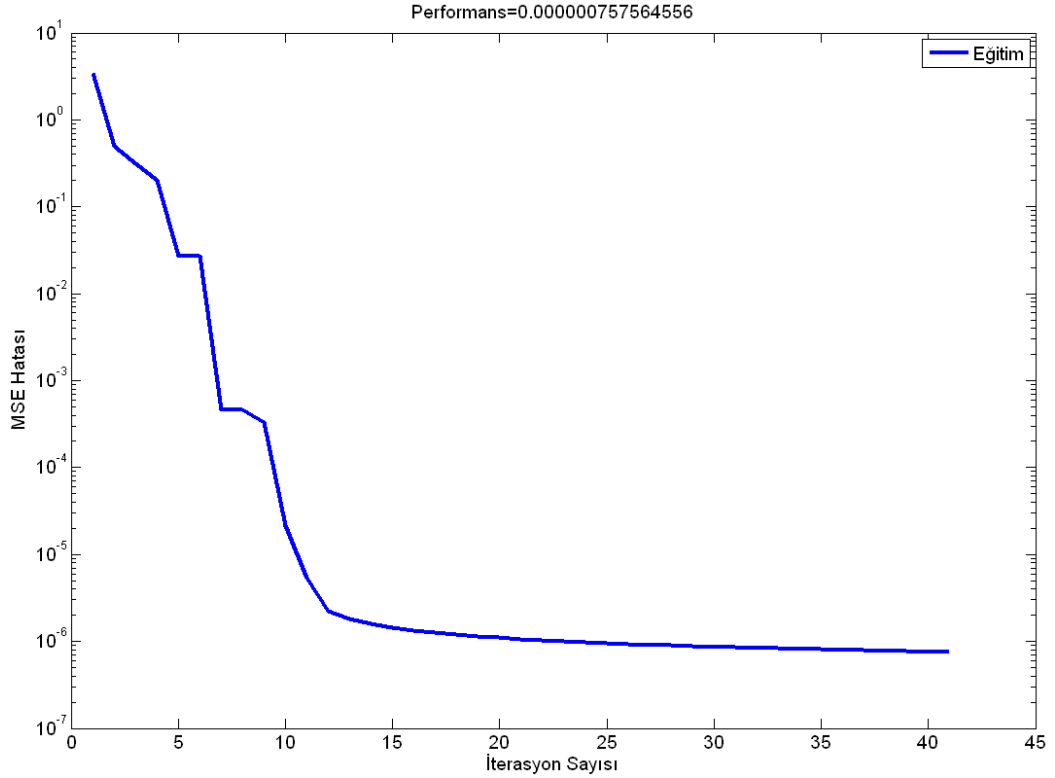
Burada IW ile gösterilen kontrol sinyalinden gelen girişlere uygulanan katsayılar ve LW çıkış sinyalinden gelen girişlere uygulanan katsayılardır. Gizli katmandaki IW olarak adlandırılan katsayıların nöron bağlantı şekli, Şekil IV.4’de gösterilmektedir. Girişe uygulanan kontrol ve çıkış sinyalleri sırasıyla IW ve LW katsayıları ile çarpılıp, giriş katmanı sonunda toplanarak çıkış katmanına uygulanmaktadır.



Şekil IV. 4 Gizli katmanındaki nöronların bağlantı şekli

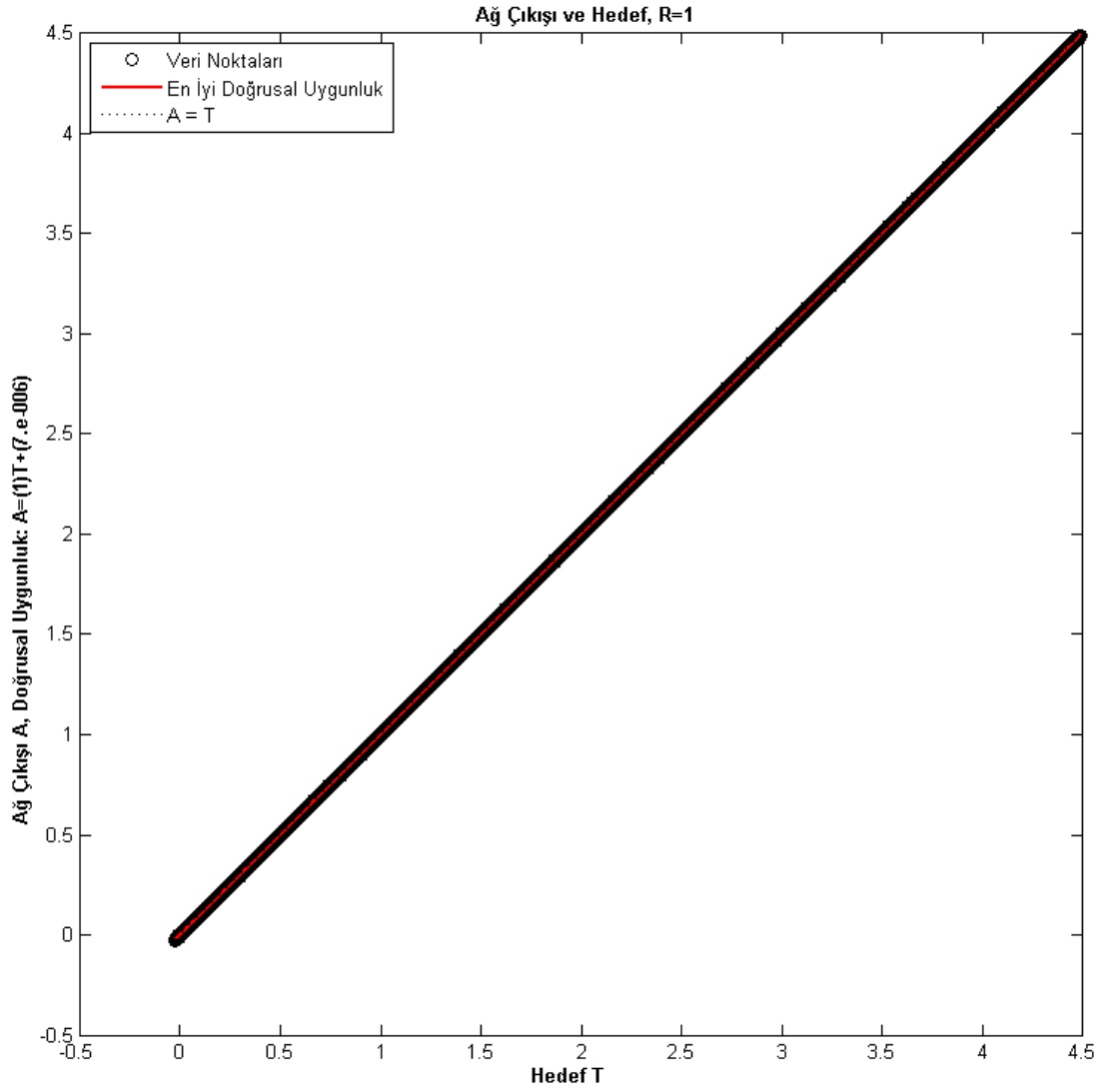
IV.1.5. YSA' nın Eğitimi

Oluşturulan YSA modeli eğitilirken, Levenberg-Marquardt eğitim algoritması kullanılmıştır. Eğitim aşamasında YSA modeli, 100 adım çalıştırılarak eğitilmiştir. Eğitim adımları boyunca MSE(Mean Squared Error) hatasının değişimi Şekil IV.5'de görülmektedir. Eğitim için parametreleri olarak iterasyon sayısı 50 ve MSE hatası 10^{-6} seçilmiştir. Ancak 41. iterasyon sonunda MSE hatası 10^{-6} 'nin altına düştüğü için YSA modelinin eğitimi sona ermiştir.



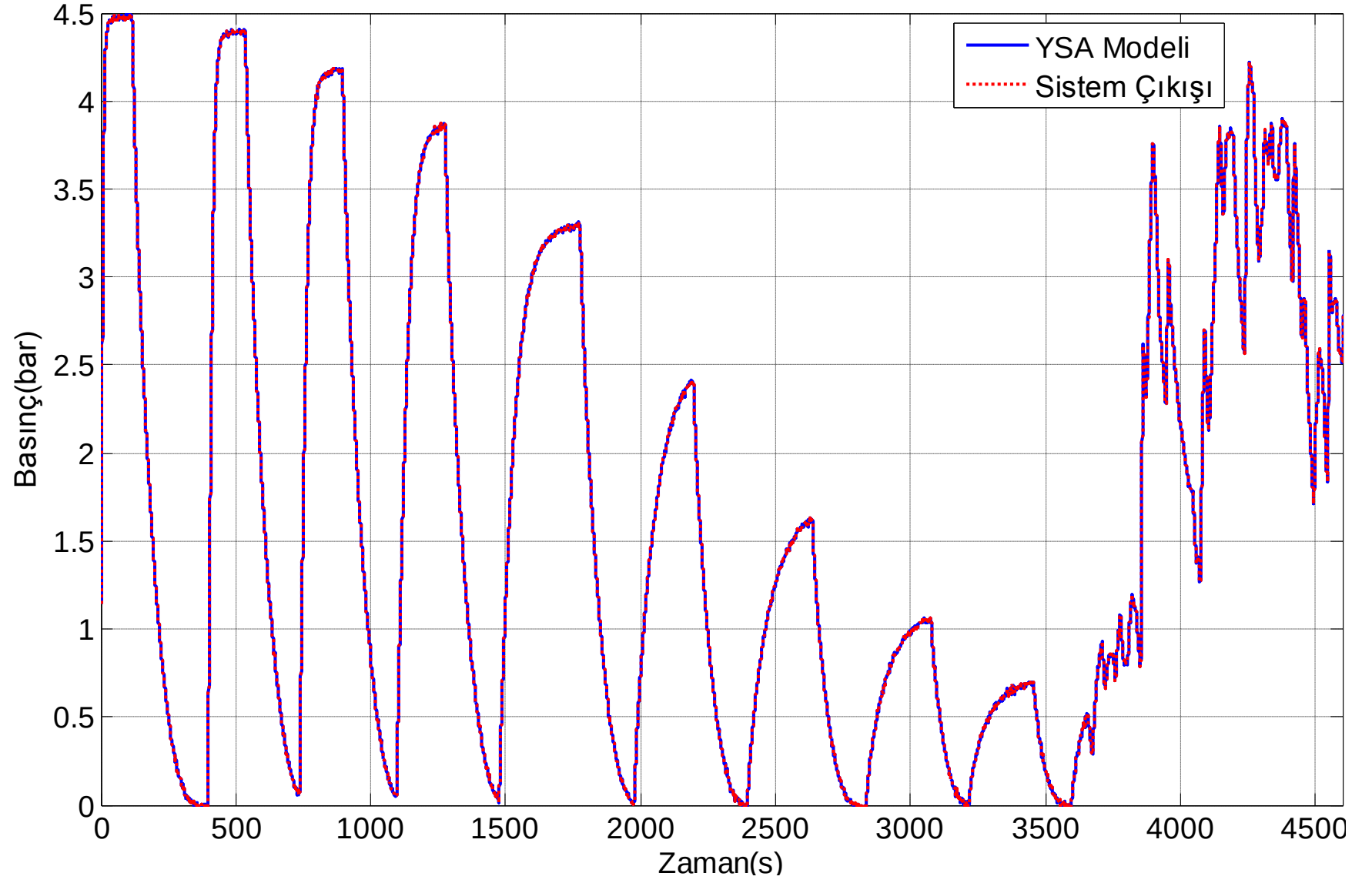
Şekil IV. 5 Eğitim boyunca MSE hatasının değişim eğrisi

Eğitim sonunda sistem ve YSA model çıkışının regresyon eğrisi Şekil IV.6'da gösterilmektedir. Şekil IV.6' dan anlaşılabacağı gibi eğitim sonunda YSA model çıkışının istenilen hedef çıkışa yakınsadığı görülmektedir.



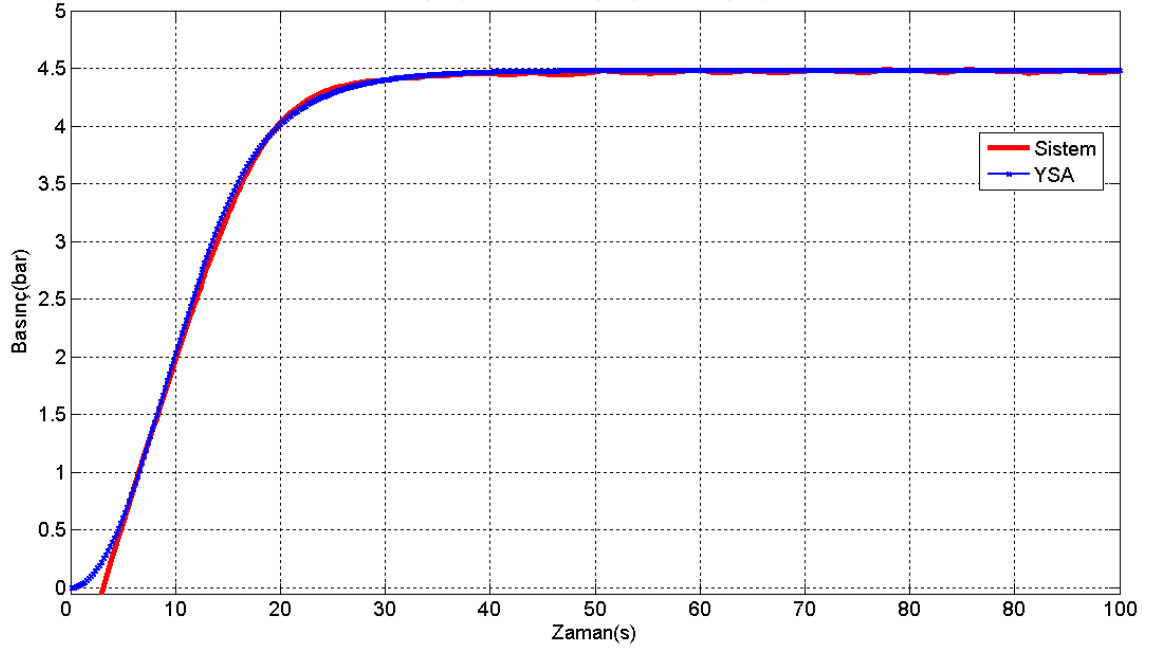
Şekil IV. 6 Hedef veriler ile YSA Model çıkışının regresyon eğrisi

Şekil IV.7’ de ise basınç sistemine uygulanan 0 ve 10V kontrol sinyali altında basınç değişimi ile eğitimin sonucu YSA modelin çıkış grafikleri verilmiştir. Grafiklerden de görüleceği üzere YSA model çıkışı, istenilen sistem çıkışına benzemektedir.



Şekil IV. 7 Sistem ile YSA Modeli çıkış grafiği

Şekil IV.8’ de, sisteme ve elde edilen YSA modeline açık çevrimde 4.5bar basamak giriş uygulandıktan sonra elde edilen grafikdir. Şekilden anlaşıldığı gibi YSA modeli, elde edilen veriler ile sistemi gerçeğe yakın modellemiştir. Yani basamak giriş uygulanınca aynı eğim ile yükselmiş ve sürekli konuma aynı değerlerde geçmiştir. Bu şekilde elde edilen YSA modeli, genetik PID kontrolör ve karınca kolonisi PID kontrolör tasarımı aşamasında offline çalışmada basınç tankı sisteminin yerine model olarak kullanılacaktır.



Şekil IV. 8 Açık çevrimde sistem ile YSA çıkışının karşılaştırılması

IV.2. GA ile PID KONTROLÖR TASARIMI

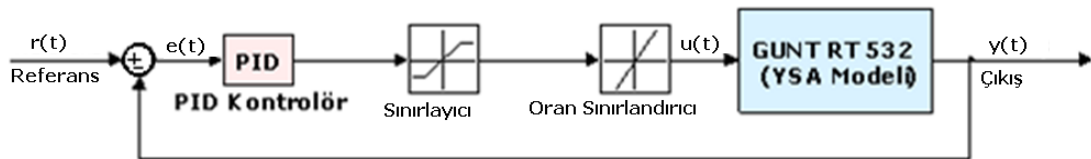
PID kontrolörün performansını etkileyen en önemli unsur, PID katsayıları olarak adlandırılan oransal, integral ve türev etki katsayılarıdır. Bu katsayıların uygun değerlerde atanması, sistemin istenilen şekilde kontrol edilmesi ile doğrudan alakalıdır.

PID Kontrolörün katsayılarının bulunması için matematiksel ve deneysel yöntemler vardır. Matematiksel yöntemlerin kullanılması için sistemin matematiksel modelinin bulunması gerekir. Deneysel yöntemlerde ise sistemi istenilen hata sınırları içerisinde kontrol edilmesi oldukça zordur ve birçok deney yapılması gerekir. Bu deneysel metotlardan ZN metodu, yapılan çalışmada yapay zekâ teknikleri ile bulunan katsayılar ile karşılaştırmada kullanılmıştır.

Bu bölümde, basınç kontrol ünitesinin PID kontrol için gerekli olan kontrolörün genel yapısı, blok yapısı, katsayı değerlerinin GA tarafından bulunması için tasarlanan PID kontrolör açıklanacaktır.

IV.2.1. Sistem İçin Tasarlanan Kontrolörün Genel Yapısı

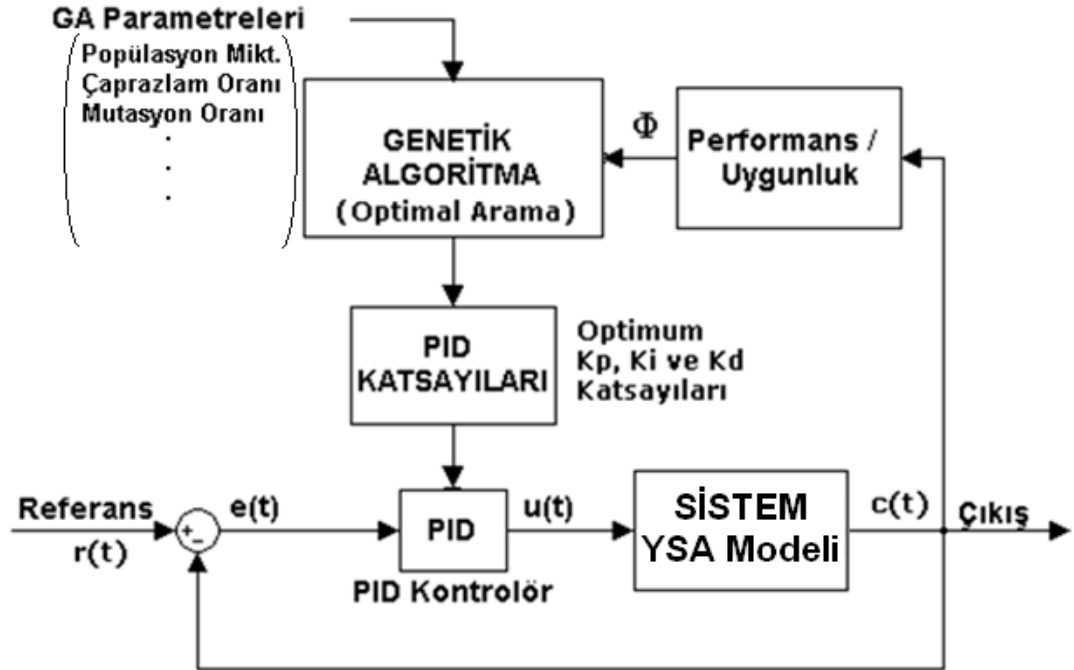
Basınç kontrol ünitesi için çevrim dışı çalışmada gerçekleştirilen sistemin genel yapısı Şekil IV.9’ da gösterilmektedir. Sistemin güvenli çalışması için sistemin kontrolünde çeşitli eklentiler yapılmıştır. PID kontrolörün çıkışı 10V ile sınırlandırılmış ve sistemin kontrolü sırasında pnomatik valfin çok hızlı değişim göstermemesi için oran sınırlayıcı kullanılarak valfin hareketi yavaşlatılmıştır. Gerçek zamanlı sistemin kontrolünde bu önlemler alındığı için çevrimdışı çalışmada da aynı eklentiler yapılmıştır.



Şekil IV. 9 Çevrim dışı çalışmada kullanılan sistemin kapalı genel yapısı

Şekil IV.9’ da gösterilen blok diyagramında bir giriş ve bir çıkıştan(siso) oluşan negatif geri beslemeli sistemde kontrolör referans ile sistemden gelen çıkışın farkını alarak oluşan hata sinyaliye göre kontrol sinyali üretmektedir.

Bu arama işlemi için tasarlanan Genetik-PID kontrolörün temel blok şeması Şekil IV.10'da gösterilmektedir.



Şekil IV. 10 GA ile PID kontrolörün optimizasyonu

Böyle bir kontrol sistemini gerçekleştirmek için, 3 önemli problemin çözülmesi gerekir. Bunlar:

1. PID parametrelerin ne şekilde genetik olarak kodlanması,
2. Genetik algoritmanın PID parametrelerini öğrenme metodu,
3. Sistem çıkış performansının genetik algoritma uygunluk değerine dönüştürülmesi.

İlerleyen bölümlerde bu problemlerin çözümü için kullanılan yöntemler açıklanacaktır[44].

IV.2.2. PID Parametrelerin Genetik Olarak Kodlanması

Çevrim dışı çalışma biçiminde, PID kontrolör katsayı değerlerinin GA tarafından bulunmadan önce, PID kontrolörün katsayılarının minimum değerlerinin belirlenmesi ve PID kontrolörün çıkışının sınırlandırılması gereklidir. Tasarlanan sistemde de, PID kontrolörün katsayılarının alt limit değeri sıfır olarak ayarlanmıştır.

GA, çözülecek problemin kendisi ile değil kodlanmış şekli üzerinden işlem yapar. Dolayısıyla çözülecek problemin kodlanma biçimi, GA performansı üzerinde oldukça önemli etki yapar. Literatürde en yaygın olarak kullanılan, Ağırlıklı İkili

Kodlama (Weighted Binary Code) ve Gerçek Kodlama olmak üzere iki tür kodlama çeşidi vardır. Bu çalışmada PID katsayılarının değerleri tek bir kromozom oluşturacak şekilde gerçel sayı kodlaması ile kodlanmıştır[23].

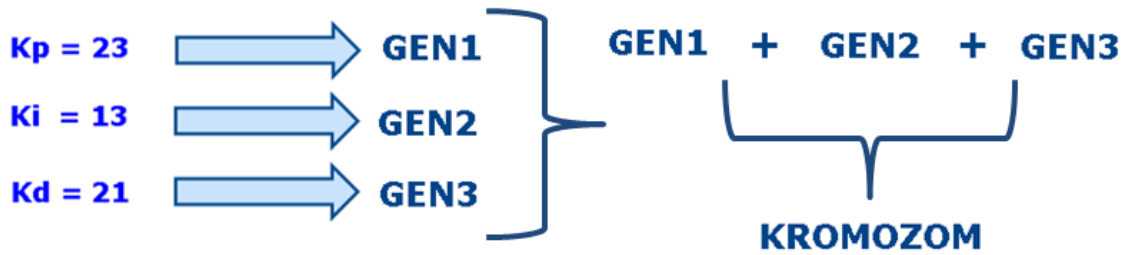
Bu kodlama biçiminde, PID kontrolör için oransal(K_p), integral(K_i) ve türev(K_d) olmak üzere toplam üç tane parametre yeterlidir. Burada her bir parametre bir gen olarak düşünülürse, 3 adet gen'e ihtiyacımızın olduğunu görürüz.

IV.2.3. PID Parametrelerin GA Tarafından Öğrenilmesi

GA parametreleri, kontrol parametreleri olarak adlandırılmaktadır. Kontrol parametreleri popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, kuşak aralığı, seçim stratejisi ve fonksiyon ölçeklemesi olarak sayılabilir[45].

GA parametreleri, genetik algoritma performansı üzerinde önemli etkiye sahiptir. Optimal kontrol parametrelerini bulmak için bir çok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılabilecek parametreler bulunamamıştır[46].

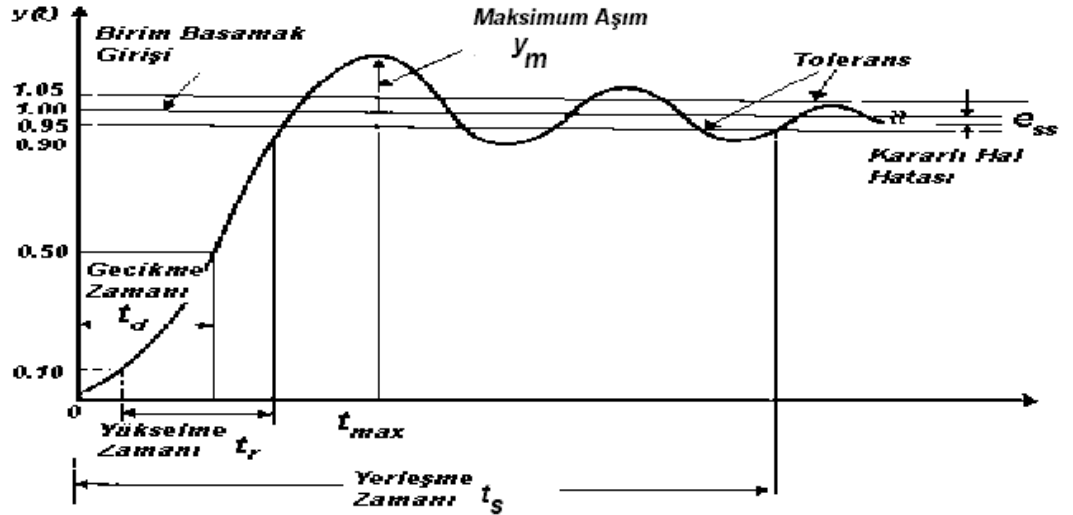
Gerçekleştirilen çalışmada kullanılan öğrenme yaklaşımı, literatürde genetik bulanık sistemlerde çok etkin çalıştığı gösterilen Pittsburgh yaklaşımıdır[23]. Bu yaklaşımı kullanmak için, PID kontrolörün katsayı değerleri tek bir kromozom içinde kodlanmıştır. Bu şekilde göre oluşturulan Kromozom yapısı Şekil IV.11'de gösterilmektedir.



Şekil IV. 11 PID Kontrolör için geliştirilen kromozom yapısı

IV.2.4. Sistem Çıkışının GA Uygunluk Değerine Dönüştürülmesi

Bir kontrol sisteminde sistem performansı, zaman uzayında incelenirken sistemin birim basamak girişine olan cevabı ortaya çıkarılır[23]. Bu incelemede kullanılan temel performans kriterleri Şekil IV.12' de gösterilmektedir. Bunlar, Maksimum aşım miktarı(y_m), gecikme zamanı (t_d), yükselme zamanı (t_r), yerleşme zamanı (t_s), kalıcı hal hatası (e_{ss}) parametreleridir.



Şekil IV. 12 Tipik bir birim basamak girişinde sistem cevabı

IV.2.4.1. Maksimum Aşım miktarı (y_m)

Maksimum aşım, sistemin kalıcı hal durumundan (y_{ss}) dan olan ilk taşma miktarını gösterir. Buna göre maksimum aşım miktarı şu şekilde ifade edilir;

$$\text{Maksimum aşım}(y_m) = y_{\max} - y_{ss} \quad (\text{IV.2})$$

Maksimum aşım miktarı sistemin bağıl kararlılığını ölçmek için kullanılır. Büyük aşım miktarları pekçok durumda tercih edilmez.

IV.2.4.2. Gecikme Zamanı (t_d)

Gecikme zamanı (t_d), sistem çıkışının, istenen çıkış seviyesinin %50 değerine ulaşması için geçen süre olarak tanımlanır.

IV.2.4.3. Yükselme Zamanı (t_r)

Yükselme zamanı (t_r), sistem çıkışın, istenen çıkış seviyesinin %10 değerinden % 90 değerine ulaşması için geçen süre olarak tanımlanır.

IV.2.4.4. Yerleşme Zamanı (ts)

Yerleşme zamanı (ts), sistem çıkışının, istenilen seviyenin %95 değerine ulaşp kaldığı süre olarak tanımlanır.

IV.2.4.5. Kalıcı Hal Hatası (ess)

Kalıcı hal hatası (ess), zaman sonsuza yaklaştığında, sistem çıkışı ile istenilen çıkış seviyesi arasındaki fark olarak tanımlanır.

Bu kriterler incelendiğinde bunlardan maksimum aşım miktarı(ym), yükselme zamanı(tr) ve kalıcı hal hatasının(ess) sistem performansını incelemede oldukça önemli oldukları görülür.

Yapılan çalışmada, sistemin çıkışının bir yörünge boyunca izlenmesi istenildiğinden, bu yörünge boyunca oluşacak hatayı minimize edecek bir uygunluk fonksiyonu oluşturulmuştur. Bu uygunluk fonksiyonu;

$$\Phi = \frac{\sum_{k=1}^n \sqrt{(r_k - y_k)^2}}{n} \quad (IV.3)$$

olarak gösterilmiştir. Burada Φ , uygunluk değeri k , çevrim dışı çalışmada örnekleme zamanı ve çalıştırılma süresine bağlı olarak elde edilen verilerin sırasını, n toplam veri sayısını, r referans verilerini ve y de sistem çıkışının verilerini gösterir. Bu denklemde, RMS (Root Mean Square) hata değeri kullanılarak kontrolörün performansı, GA için uygunluk değerine çevrilmiştir.

Denklem (IV.3)'deki formülden elde edilen uygunluk değerine göre sistemin çıkışı ile referans arasındaki RMS hatasının minimum olduğu PID katsayıları GA ile bulunmaktadır. . Böylece verilen referansı PID kontrolörlü sistem, minimum hata ile takip etmektedir.

IV.2.5. Kullanılan Genetik Operatörler

GA' da temel olarak kodlama, popülasyon büyüklüğü, çaprazlama, mutasyon ve seçim gibi genetik operatörler kullanılır. Gerçekleştirdiğimiz sistemde kullanmış olduğumuz genetik operatörler ve değerleri şöyledir;

Kodlama: Bölüm III.2'de GA' nın sistemin parametresiyle değil kodlanmış biçimiyle çalıştığı belirtilmişti. Yapılan çalışmada kodlama çeşidi olarak gerçel sayı kodlama seçilmiştir ve PID katsayılarının değerleri tek bir kromozom oluşturacak şekilde gerçel sayılar ile kodlanmıştır.

Popülasyon büyüklüğü: GA' da popülasyon büyüklüğünün performansı oldukça etkilediği bilinmektedir. Popülasyon büyüklüğünün çok yüksek değerlerinde işlemcinin kullanımı ve sonuç bulma zamanı oldukça artmakta, çok küçük popülasyon değerlerinde ise GA istenilen sonucu bulamamaktadır. Bu yüzden farklı deneyler yapılmış ve en uygun popülasyon büyüklüğü olarak $N=20$ seçilmiştir.

Çaprazlama: Aritmetik çaprazlama operatörü kullanılmıştır. Çaprazlama operatörünün olasılık değeri 0.8 olarak seçilmiştir.

Mutasyon: Gaussian mutasyon operatörü kullanılmıştır. Mutasyon operatörünün ölçeği ve daralma oranı 1 olarak seçilmiştir.

Üreme: Gerçekleştirilen çalışmada GA yapısında bütünüyle yer değiştirme üreme operatörü kullanılmıştır. Bu üreme yöntemine ilave olarak elistik model kullanılmıştır. Literatürde elistik model, popülasyon içinde en iyi uygunluk değerine sahip olan dört bireyin, bir sonraki generasyona değişikliğe uğramadan aktarılması olarak tanımlanır. Bu şekilde, en uygun bireyin kromozomlarının seçim etkisi ile sonraki generasyonlara aktarılmadan ölmesinin önüne geçilir. Elistik model ile GA'nın performansının arttığı gözlenmiştir.

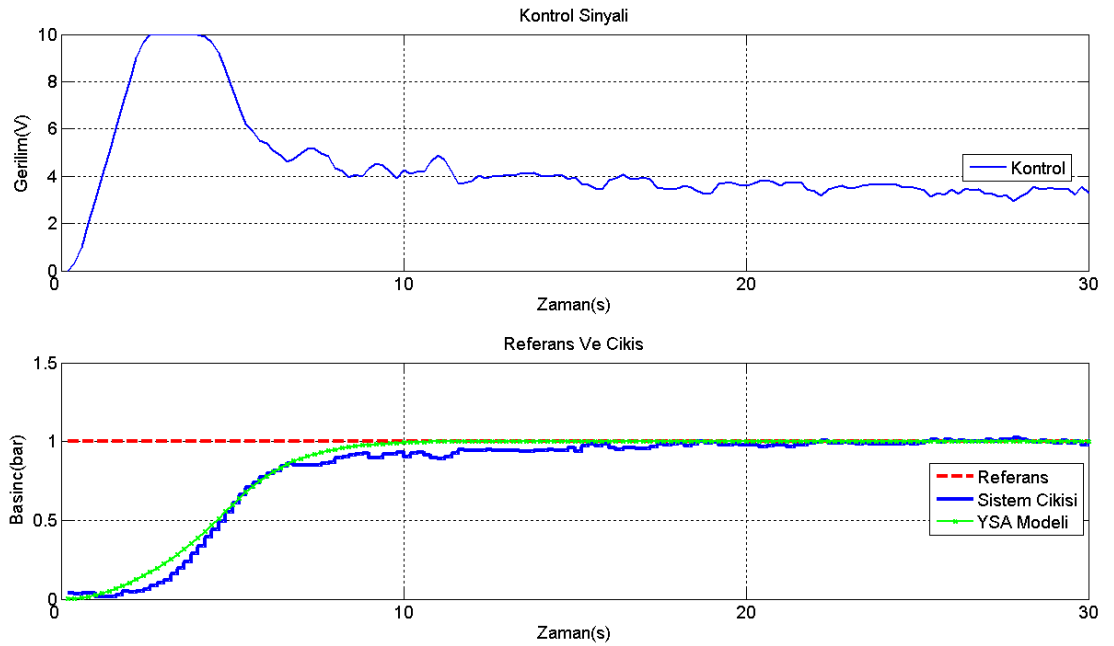
Seçim: Seçim operatörü için turnuva yöntemi kullanılmıştır. Yapılan deneylerde turnuva büyüklüğü 4 olarak seçilmiştir. Sistemin çalışmasında uyumsuz PID katsayılarının üremesiyle yani kötü bireylerin oluşmasıyla sistem osilasyon yapmış ve kalıcı hal hataları oluşmuştur. Bu durumlardan dolayı turnuva yöntemi ile oluşan kötü bireylerin bir sonraki generasyona geçişi engellenmiş ve turnuva yönteminin performansının daha iyi olduğu gözlenmiştir.

IV.3. GENETİK-PID KONTROLÖRÜN PERFORMANSI

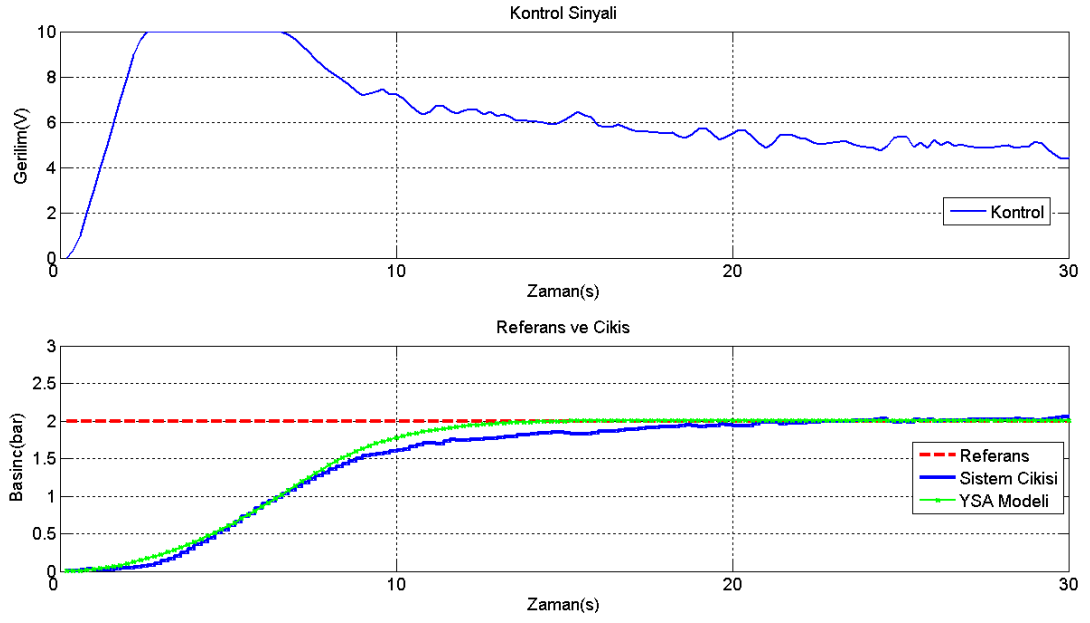
Bu kısımda YSA modeli ile yapılan optimizasyon sırasında elde edilen Genetik-PID kontrolörün gerçek zamanlı çalışmalarda performansı karşılaştırılmıştır. Yapılan çalışmalardan da anlaşılacağı gibi YSA modelinde bulunan PID katsayıları gerçek zamanlı olarak sistem kontrolünde kullanıldığında başarılı sonuçlar elde edilmiştir.

IV.3.1. Basamak Girişlere Sistemin Davranışı

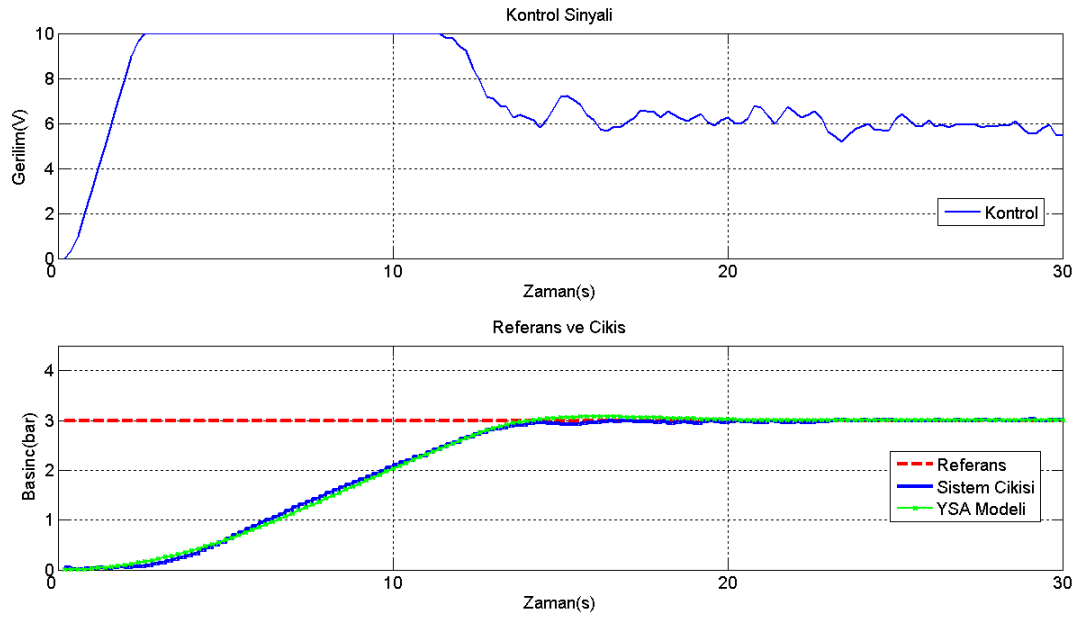
Sisteme; Şekil IV.13’de gösterildiği gibi 1bar, Şekil IV.14’de gösterildiği gibi 2bar, Şekil IV.15’de gösterildiği gibi 3bar, Şekil IV.16’de gösterildiği gibi 4bar değerleri referans olarak verilmiş, bu referans değerlerini takip eden YSA model ve gerçek sistem çıkışları gösterilmiştir. Bu çıkışları sağlayan kontrol sinyalleri şekillerin üst kısmında gösterilmiştir. Kontrol sinyallerinin davranışlarından da anlaşılacağı gibi sistemin çıkışı yaklaşık referans değerlere yaklaşıncı, PID kontrolör sistemi maksimum sürmeyi bırakmış ve kararlı durum için kontrol sinyali üretmeye başlamıştır.



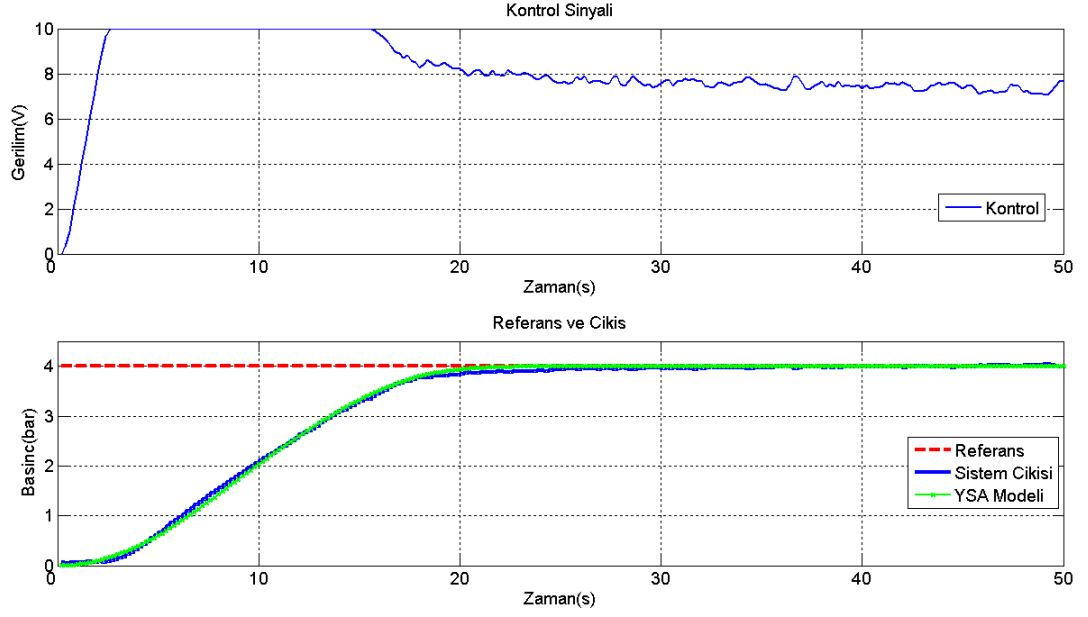
Şekil IV. 13 1bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları



Şekil IV. 14 2bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları



Şekil IV. 15 3bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları

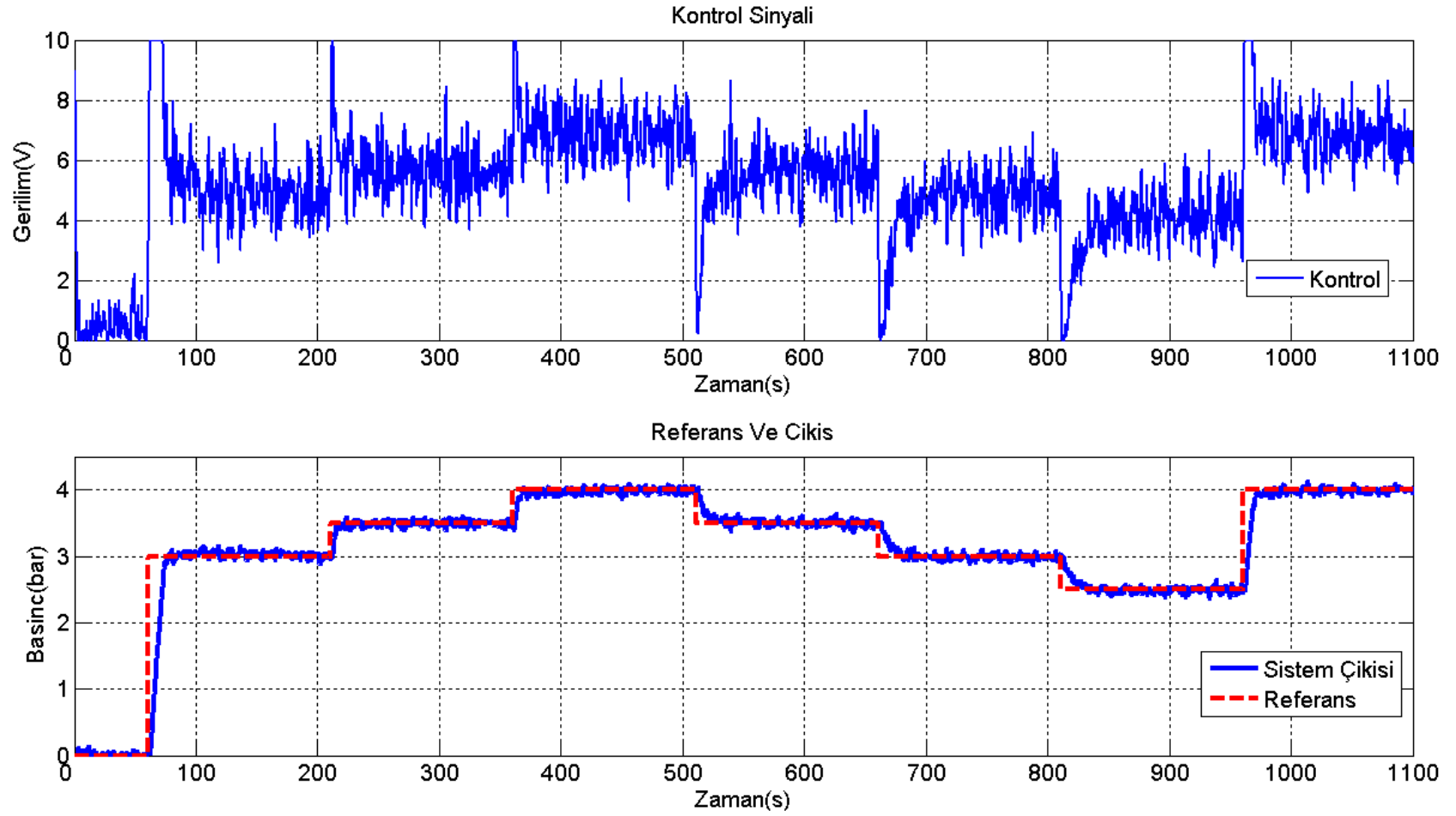


Şekil IV. 16 4bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları

IV.3.2. Basamak Girişler Arasında Geçiş Sistemin Davranışı

Bu çalışmada Genetik-PID kontrolörün, sistemin çalışma anında değişik referans değerine geçiş cevabını görebilmek için farklı referanslar vererek sistemin cevabı incelenmiştir. Çalışma sırasında sistem, ilk referans değerine oturduktan sonra referans değerini değiştirerek sistemin referanslar arasında geçişi incelenmiştir.

Şekil IV.17' de bu çalışmaya ait grafikler verilmiştir. Grafikten de anlaşılabacağı gibi sistem farklı basamak girişleri arasında geçişini başarılı bir şekilde gerçekleştirmiştir. Buradan da anlaşılabacağı gibi GA ile bulunan PID katsayıları ile sistem gerçek zamanlı olarak başarılı bir şekilde kontrol edilmiştir.



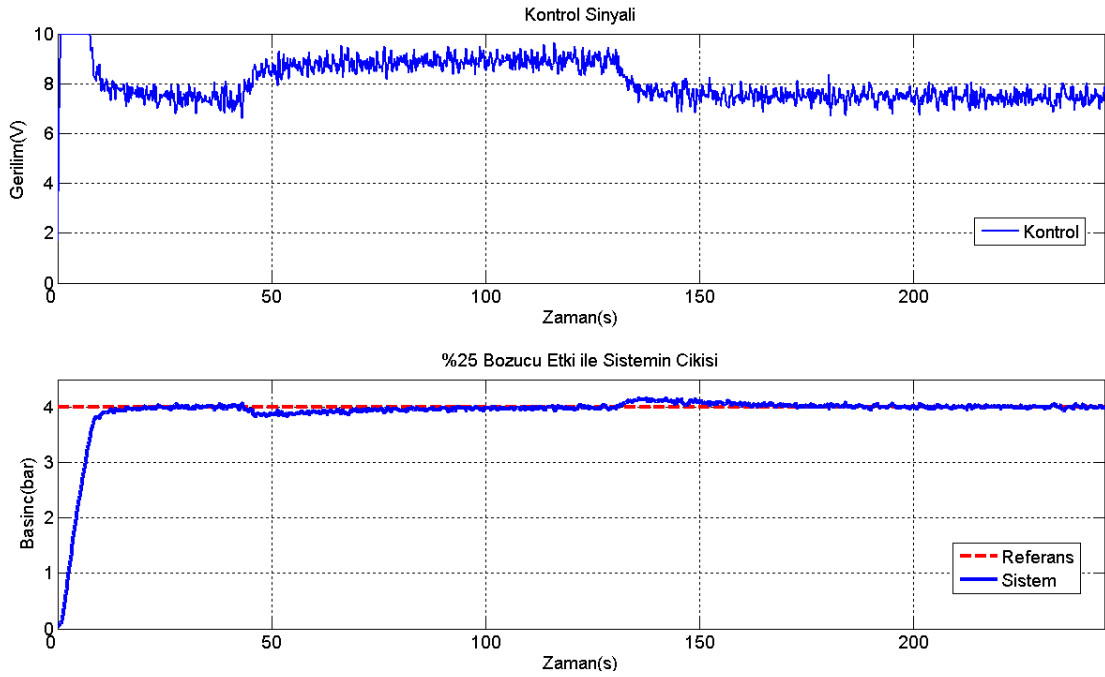
Şekil IV. 17 Basamak geçişler sırasında gerçek sistemin davranışı

IV.3.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı

Çalışmanın bu kısmında gerçekleştirilen Genetik-PID kontrolörün, basınç tankına dışarıdan ithal edilen bir bozucu etki karşısında performansı incelenecektir.

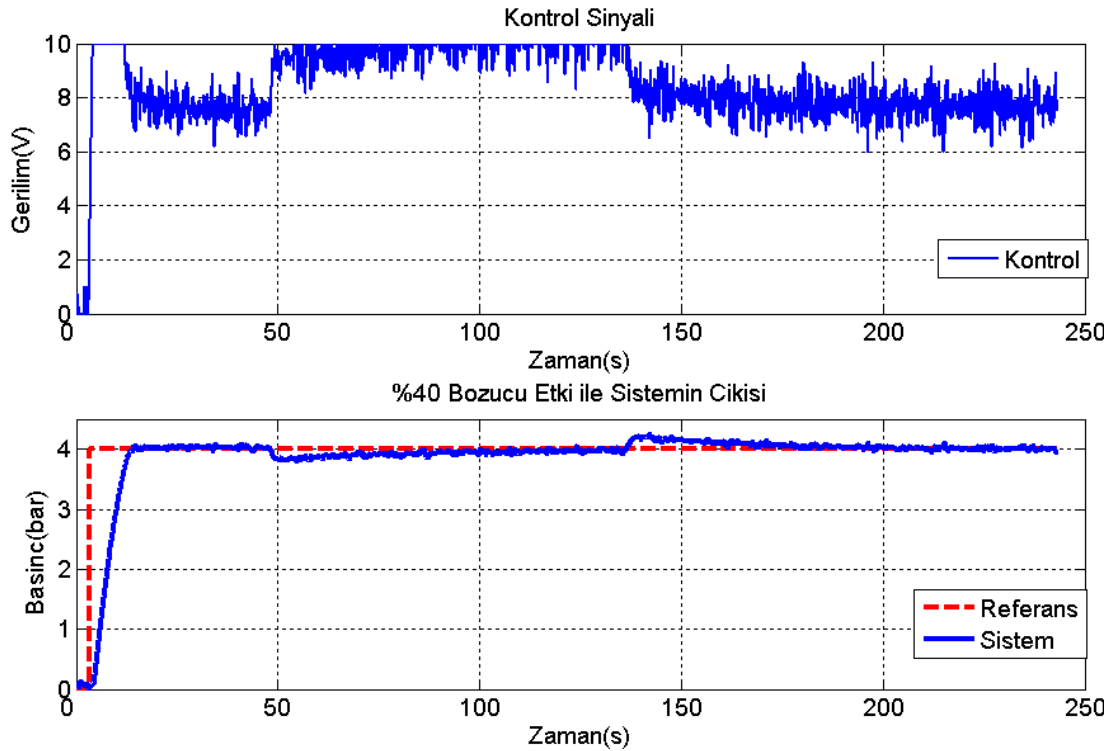
Sistemin çalışması sırasında basınç tankının çıkışında bulunan pnomatik oransal valfe 6V gerilim uygulanarak valfin %50 oranında açılması sağlanmıştır. Böylece sistem çalışırken fazla havanın boşaltılması gerçekleştirilmiştir. Bozucu ithal edilmesi ile çıkıştaki valfin açıklık değeri artırılarak bozucu ithal edilmiş olunur. Valfe 6V'tan fazla gerilim verilmesi ile tank içindeki havanın boşalma hızı da artacak ve sisteme negatif bozucu ithal edilmiş olunacaktır. Bu bozucuya karşılık sistemin davranışları incelenecektir.

Şekil IV.18' de sistemin çıkışındaki oransal valfe %25 oranında daha fazla gerilim verilerek Genetik-PID kontrolörün davranışı incelenmiştir. Burada sisteme, 4bar referans değerine oturduktan sonra 90s boyunca bozucu giriş uygulanmış ve sistemin tekrar normal çalışma düzenine geçilmiştir. Bozucu giriş uygulandığında sistemin çıkışı 4bar basınç değerinin altına inmiş ancak kontrol sinyalinin 7.5V seviyesinden 9V'a yükselmesi ile sistemin çıkışı tekrar 4bar seviyesine gelmiştir. Bozucu kalktıktan sonra sistemin çıkışı 4bar'ın üzerine çıkmış ve kontrol sinyalinin 7.5V seviyesine gelmesi ile sistemin çıkışı tekrar 4bar'lık referans değerine oturmuştur.



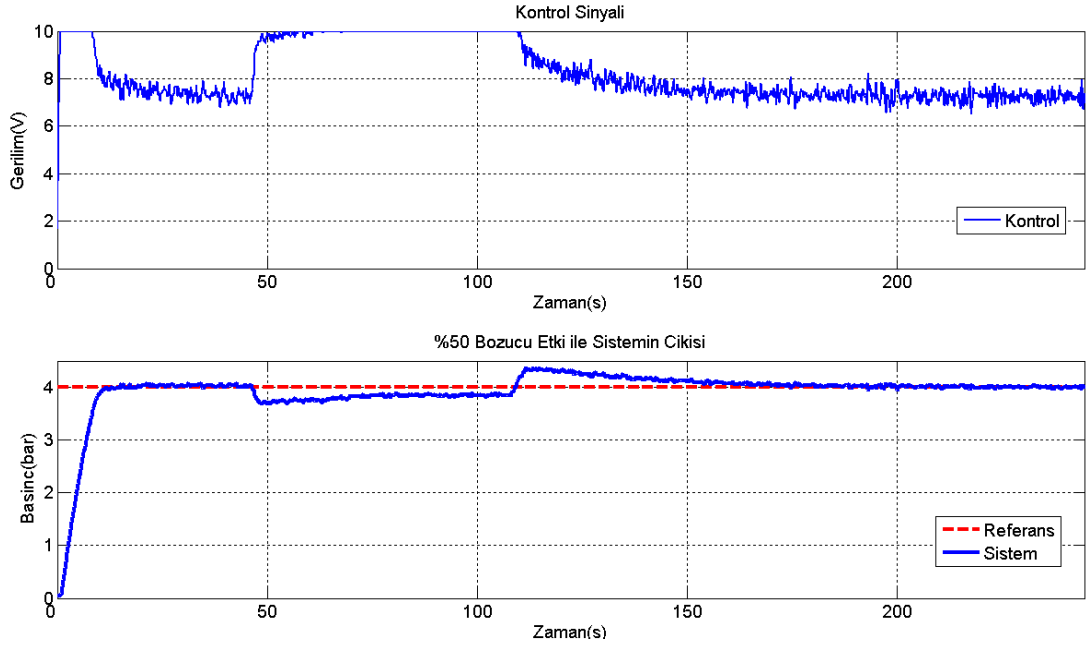
Şekil IV. 18 %25 bozucu uygulanan sistemin cevabı

Şekil IV.19’da sistemin çıkışındaki oransal valfe %40 oranında daha fazla gerilim verilerek Genetik-PID kontrolörün davranışı incelenmiştir. Burada sisteme, 4bar referans değerine oturduktan sonra 90s boyunca bozucu giriş uygulanmış ve sistemin tekrar normal çalışma düzenine geçilmiştir. Bozucu giriş uygulandığında sistemin çıkışı 4bar basınç değerinin altına inmiş ancak kontrol sinyalinin 7.5V seviyesinden 9.5V’a yükselmesi ile sistemin çıkışı tekrar 4bar seviyesine gelmiştir. Bozucu kalktıktan sonra sistemin çıkışı 4bar’ın üzerine çıkmış ve kontrol sinyalinin 7.5V seviyesine gelmesi ile sistemin çıkışı tekrar 4bar’lık referans değerine oturmuştur.



Şekil IV. 19 %40 bozucu uygulanan sistemin cevabı

Şekil IV.20' de sistemin çıkışındaki oransal valfe %50 oranında daha fazla gerilim verilerek Genetik-PID kontrolörün davranışı incelenmiştir. Burada sisteme, 4bar referans değerine oturduktan sonra 70s boyunca bozucu giriş uygulanmış ve sistemin tekrar normal çalışma düzenine geçilmiştir. Bozucu giriş uygulandığında sistemin çıkışı 4bar basınç değerinin altına inmiş ancak kontrol sinyalinin 10V yani %100 olması dahi sistemin referans değerine oturmasını sağlamamıştır. Bozucu kalktıktan sonra sistemin çıkışı 4bar'ın üzerine çıkmış ve kontrol sinyalinin 7.5V seviyesine gelmesi ile sistemin çıkışı 4bar'lık referans değerine tekrar oturmuştur.

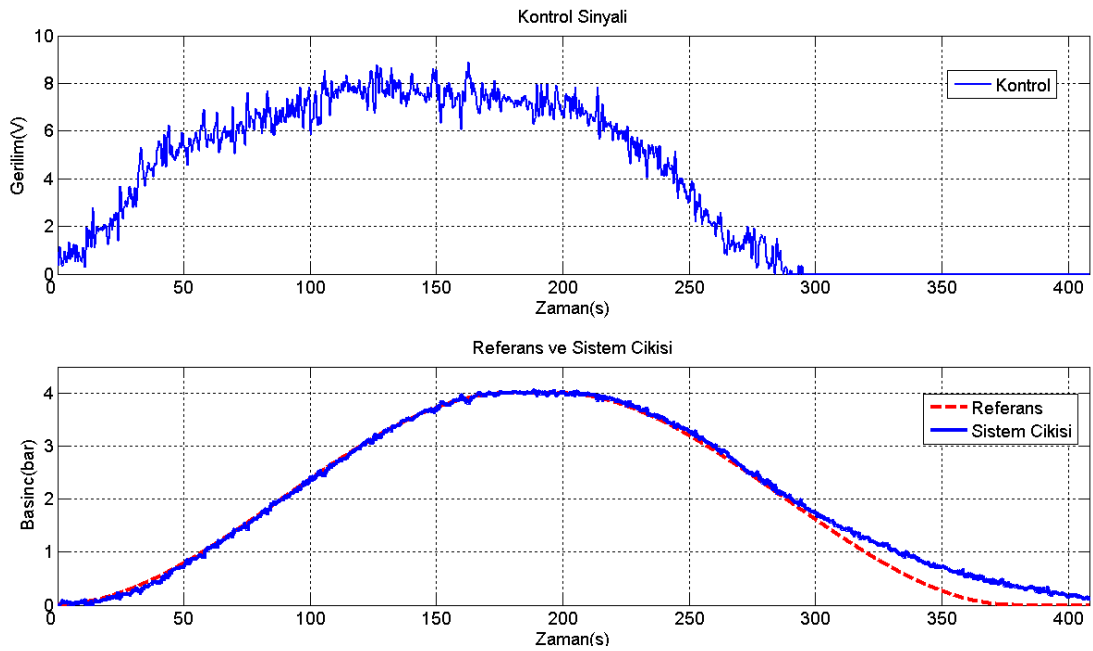


Şekil IV. 20 %50 bozucu uygulanan sistemin cevabı

IV.3.4. Genetik-PID Kontrolörün Yörünge Takip Cevapları

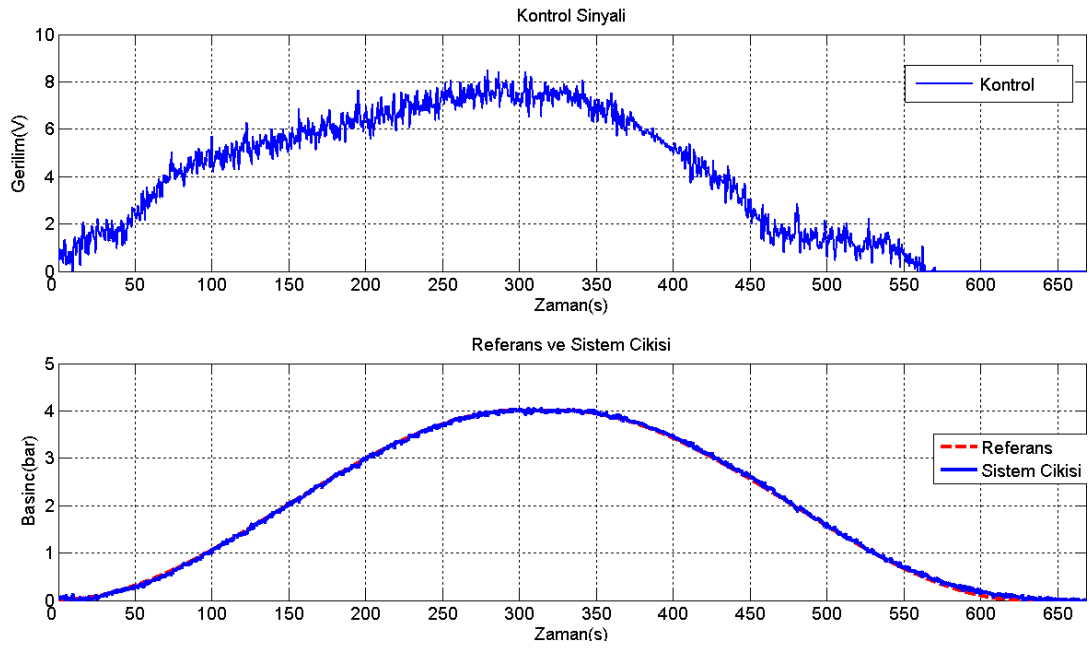
Çalışmanın bu kısmında Denklem II.4'de tanımlanan yörünge fonksiyonun, 0 bardan 4 bara kadar farklı eğim dereceleri ile oluşturulan yörüngelerin, sisteme referans yörünge olarak verilmesi ile sistemin davranışı incelenecektir.

İlk olarak sistemin, başlangıç değerinden 4 bara kadar takip etmesi istenmiştir. Bunun için eğime 6 değeri verilerek yörünge oluşturulmuş ve sisteme referans olarak verilmiştir. Sistem 4 bara geldiğinde ise ilk konumuna gelmesi için gerekli yörünge oluşturularak sistemin boşalması sırasındaki cevabı incelenmiştir. Sistemin cevabı Şekil IV.21'de gösterilmektedir. Sistemin dolum sırasında yörüngeyi başarılı bir şekilde takip ettiği görülmüş ancak 300s sonrasında kontrol sinyalinin 0 olmasına rağmen sistemin doğal boşalması yörüngeyi takip etmek için yeterli olmadığı için son kısımlarda yörünge ile sistem davranışı arasında fark oluşmuştur.



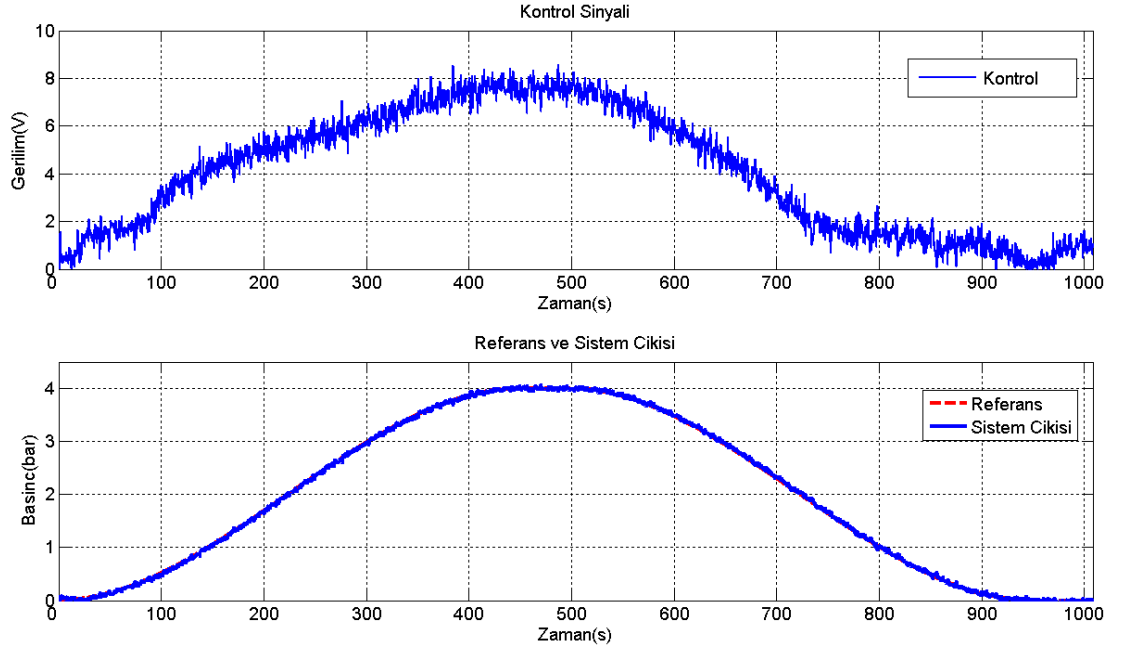
Şekil IV. 21 Eğimi 6 olan referans yörünge ve sistemin takibi

Dolum ve boşaltım referans yörüngeleri, eğim 10 yapılarak birleştirilmiş ve sisteme referans olarak verilmiştir. Bu referans yörüngeye sistem, Şekil IV.22’de gösterildiği gibi bir davranış göstermiştir. Sistemin, dolum ve boşaltım sırasında yörüngeyi başarılı bir şekilde takip ettiği grafikten anlaşılmaktadır. Ancak 560s civarında kontrolörün çıkış sinyali olarak 0V üretmesine rağmen sistemin doğal davranışı, yörüngeyi boşalma hızına yetişememiş ve küçük hata ile yörünge takip edilmiştir.



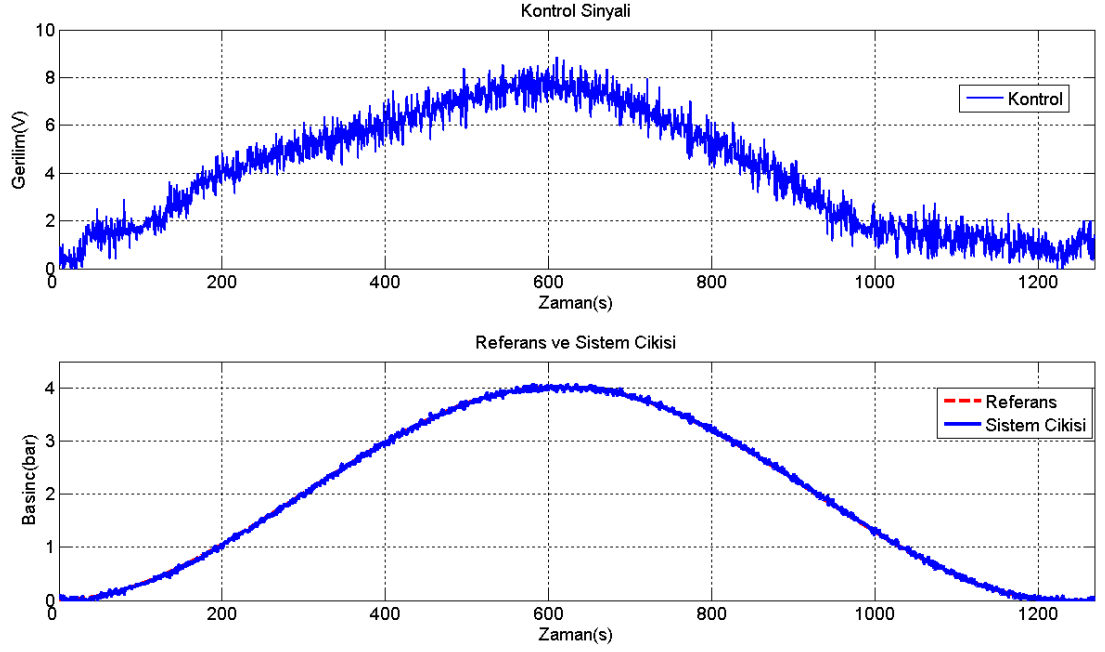
Şekil IV. 22 Eğimi 10 olan referans yörüngeyi sistemin tarafından takibi

Dolum ve boşaltım referans yörüngeleri, eğim 15 yapılarak birleştirilmiş ve sisteme referans olarak verilmiştir. Bu referans yörüngeye sistem, Şekil IV.23’de gösterildiği gibi bir davranış göstermiştir. Sistemin, dolum ve boşaltım sırasında yörüngeyi başarılı bir şekilde takip ettiği grafikten anlaşılmaktadır.



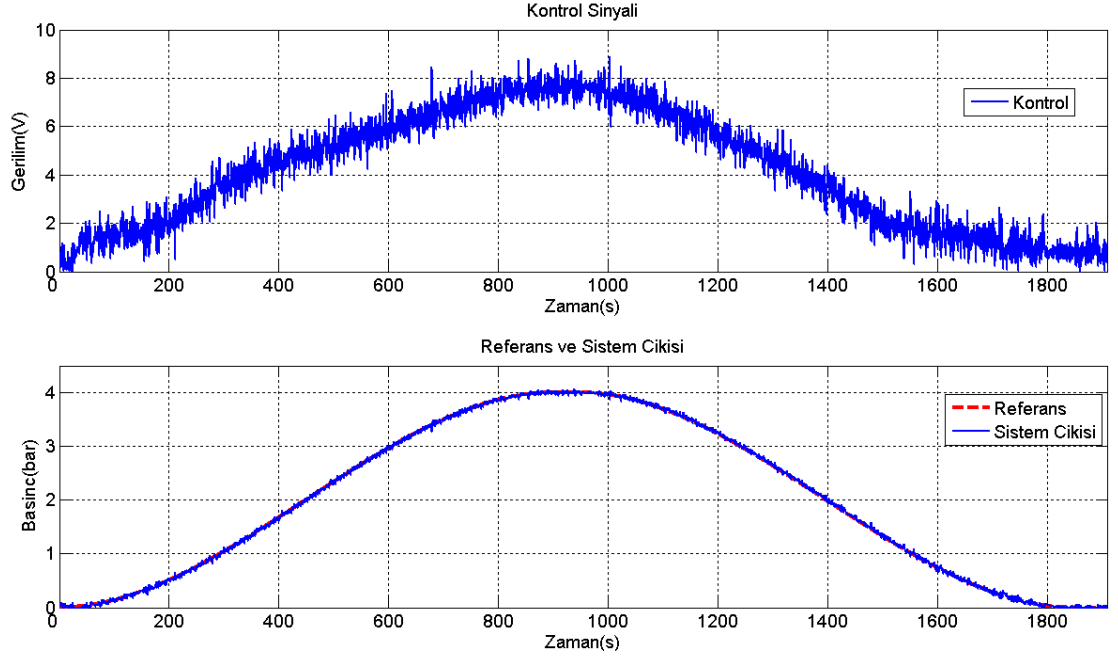
Şekil IV. 23 Eğimi 15 olan referans yörüngeyi sistemin tarafından takibi

Dolum ve boşaltım referans yörüngeleri, eğim 20 yapılarak birleştirilmiş ve sisteme referans olarak verilmiştir. Bu referans yörüngeye sistem, Şekil IV.24’de gösterildiği gibi bir davranış göstermiştir. Sistemin, dolum ve boşaltım sırasında yörüngeyi başarılı bir şekilde takip ettiği grafikten anlaşılmaktadır.



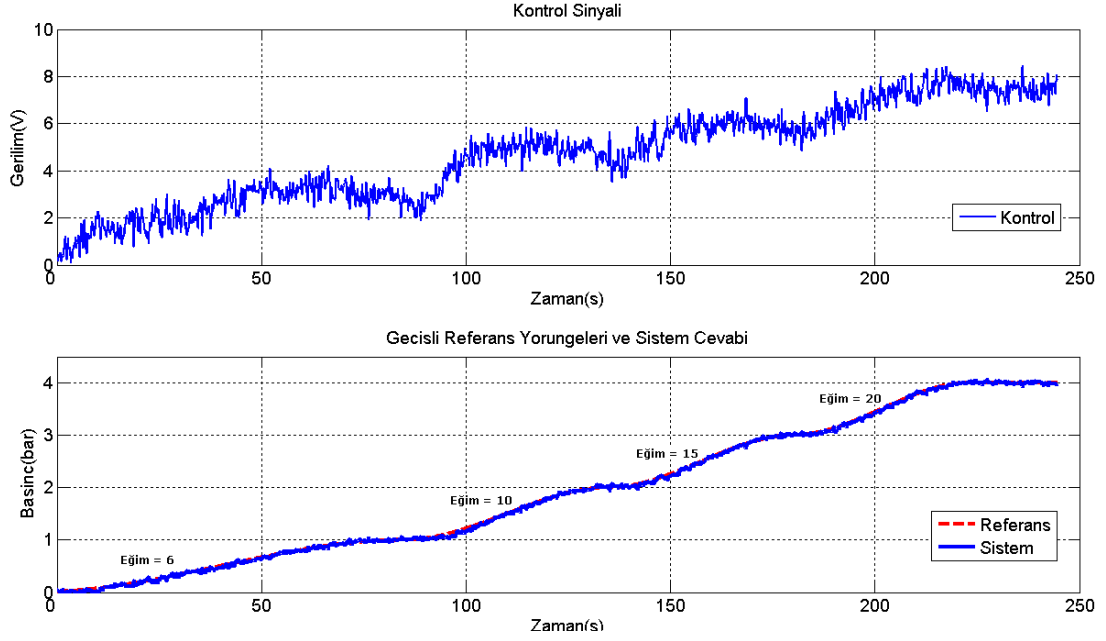
Şekil IV. 24 Eğimi 20 olan referans yörüngeyi sistemin tarafından takibi

Dolum ve boşaltım referans yörüngeleri, eğim 30 yapılarak birleştirilmiş ve sisteme referans olarak verilmiştir. Bu referans yörüngeye sistem, Şekil IV.25’de gösterildiği gibi bir davranış göstermiştir. Sistemin, dolum ve boşaltım sırasında yörüngeyi başarılı bir şekilde takip ettiği grafikten anlaşılmaktadır.



Şekil IV. 25 Eğimi 30 olan referans yörüngeyi sistem tarafından takibi

Genetik-PID kontrolörlü sistemin farklı eğimli yörünge değerleri arasında geçişi sırasında başarısını görmek için Şekil IV.26’ da gösterildiği gibi referans yörünge verilmiştir. Şekil IV.26’ dan anlaşılabacağı gibi sistem yörüngeler arasında başarılı bir şekilde geçiş sağlamıştır.

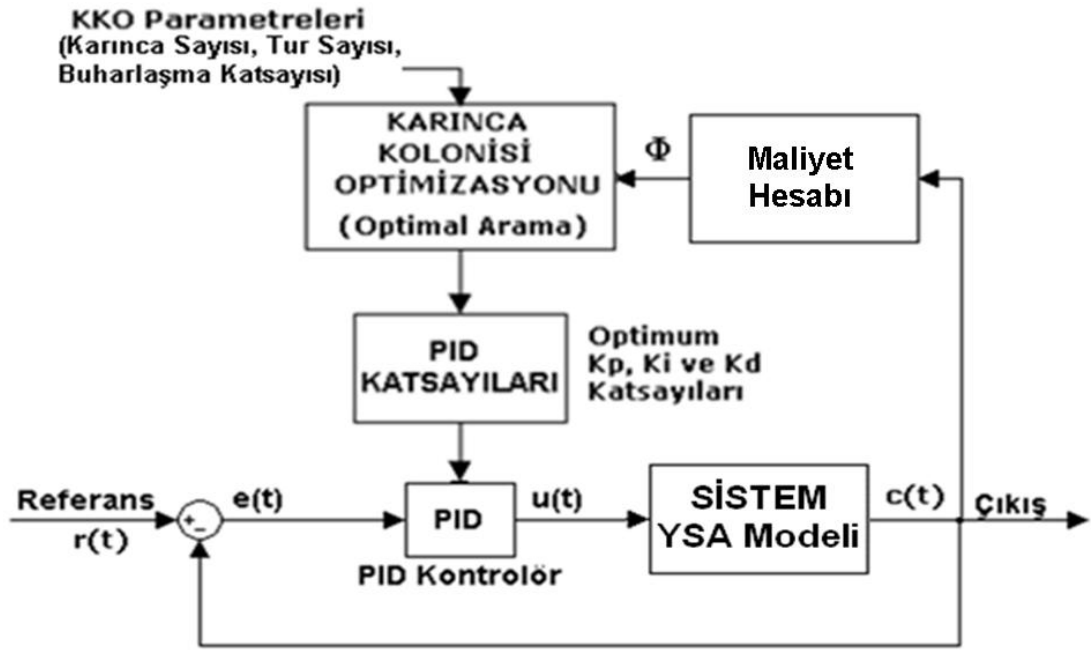


Şekil IV. 26 Yörüngeler arası geçişe sistemin cevabı

Elde edilen grafiklerden de anlaşılabacağı gibi, çevrim dışı çalışmada GA ile optimize edilen PID kontrolör ile sistemin gerçek zamanlı kontrolü farklı referanslar uygulansa dahi başarılı bir şekilde gerçekleştirilmektedir.

IV.4. KKA ile PID KONTROLÖR TASARIMI

Karıncaların en kısa yolu bulma mantığından yola çıkılarak elde edilen KKA algoritması ile herhangi bir kontrolöre ait parametre optimizasyonu sağlanabilmektedir. PID kontrollü bir sistemin K_p , K_i , ve K_d parametrelerini KKA algoritması ile optimize etmek mümkün olmaktadır. Şekil IV.27’de verilen referansı izlemek üzere gerekli PID katsayılarının bulunması için KKA algoritması ile tasarlanmış optimizasyonun blok diyagramı görülmektedir.

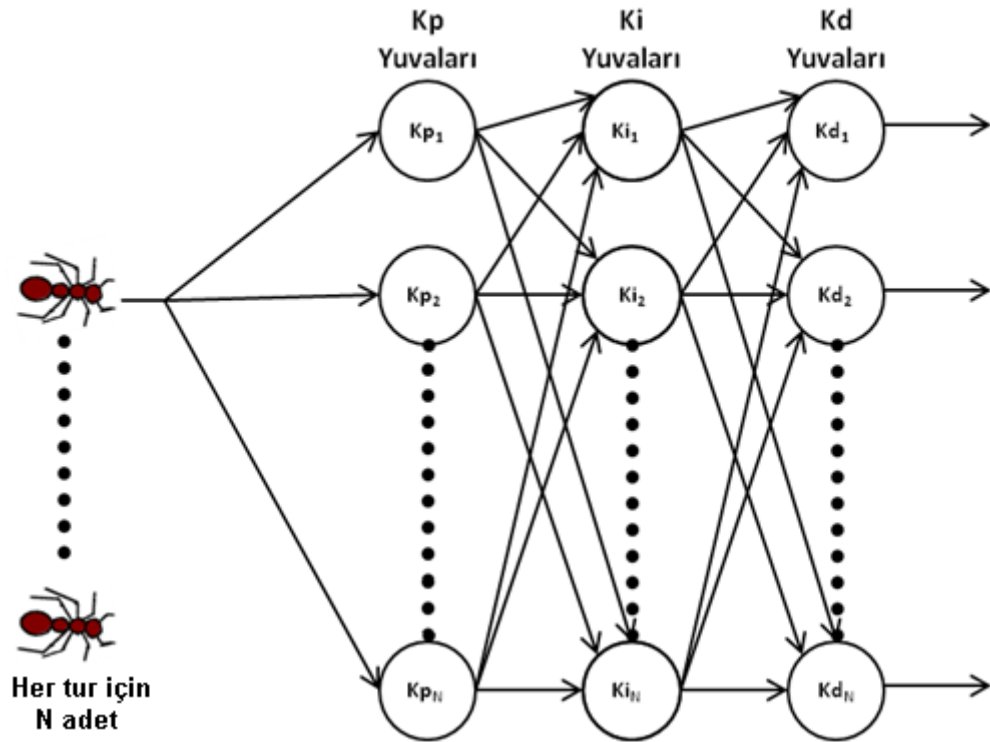


Şekil IV. 27 KKA ile PID Kontrolör optimizasyonu

KKA algoritmasında her yapay karıncanın seçmiş olduğu yollar farklı PID katsayılarına denk gelir. Optimizasyon işlemi başladığında KKA, kontrolör için PID katsayılarını atar ve bu katsayılar ile sistem çevrimdışı çalıştırılır. Sisteme verilen referans olarak yörünge verilmektedir. Böylece KKA’ nun bulacağı PID katsayı değerleriyle çalıştırılan simülasyondan elde edilen çıkış ile referans yörünge karşılaştırılacaktır. Birbirinden farklı bu çıkış eğrileri aslında her bir karıncanın izlediği yol olarak düşünülebilir. Dolayısıyla referans yörüngeye en yakın çıkışı verecek olan PID değeri optimal değerlerdir. Çıkış işareti optimizasyon sürecinde incelenirse ilk başlardaki rastgele ve dağınık olan eğriler kendini daha dar bir bantta ve referansın etrafında hareket eden eğrilere bırakır. Böylece sistem çıkışı ile referans yörünge arasında az hata oluşturan yani yapay karıncalar tarafından fazla tercih edilerek feromeni yüksek olan yollara ait değerler, optimizasyon sonucu elde edilen K_p , K_i ve K_d katsayılarıdır.

Optimize edilecek parametre sayısı arttıkça yani yuva sayısı arttıkça karıncaların işi daha da zorlaşmaktadır. Ancak bir süre sonra kötü yolların feromen miktarının azalması, kısa yolların feromen miktarının artması ile karıncalar sadece iyi yollar arasından en iyisini bulmaya çalışacaklardır. Bu sürecin sonunda karıncalar referansa çok yakın yol bulacaklardır. KKA algoritmasının tasarımında dikkat edilmesi gereken bazı özellikler vardır. Öncelikle optimize edilecek değişkenlerin arama uzayının çok büyük olmasını engellemek için maksimum ve minimum değerleri seçilir. Bu sayede KKA, sınırlandırılan çalışma uzayı içerisinde arama yapabilecektir.

Şekil IV.28’ de gösterildiği gibi PID kontrolör için her bir katsayı değeri bir yuva gibi düşünülürse bu yuvalara giren bir karınca bir sonraki yuvayı seçerken varsa feromeni fazla olan yolu, yoksa yani ilk karınca ise rastgele bir yuva seçerek diğer yuvaya girmektedir.



Şekil IV. 28 PID katsayıları için yapay karıncanın izleyebileceği yollar

Her yuvadan çıkışında önceki yuvalardan aldığı değerleri de tutmaktadır. 3. yuvadan çıktığı zaman elde ettiği değerler PID parametrelerine ait değerlerdir. Bu değerler karıncanın izlediği yol ile elde edilecek yörüngeyi verecektir. Dolayısıyla elde edilecek yörünge ile referans yörünge karşılaştırılarak karıncanın izlediği yollara

“ödül” veya “ceza” feromenleri verilecektir. Gerçekleştirilen çalışmada, aynı tur içerisinde “lokal feromen güncelleme”, her bir tur sonunda ise “global feromen güncelleme” olmak üzere feromen güncelleme iki şekilde yapılmaktadır. Lokal feromen güncelleme işleminde her bir karınca izlediği yolu bitirdikten sonra bırakılan feromeni her bir tur içerisinde Denklem IV.5’de verilen eşitlik ile güncellemektedir[15].

$$\tau_{ij}(t) = \tau_{ij}(t-1)^\lambda + \frac{0.2a}{c(i)} \quad (IV.5)$$

Burada τ_{ij} , feromen matrisinin satır ve sütun indisindeki feromen miktarını, a genel feromen güncelleme katsayısı, λ buharlaşma katsayısı ve $c(i)$ ise karınca tarafından o tur içerisinde elde edilen referans yörünge ile elde edilen çıkış eğrisi arasındaki RMS hatasıdır. Dolayısıyla karıncanın izlediği yola ait RMS hatası düşük oldukça yola verilen feromen değeri artacaktır. Global feromen güncelleme için ise pozitif ve negatif olmak üzere iki farklı feromen güncelleme yöntemi kullanılmaktadır. Her tur sonunda o tura ait en iyi ve en kötü tura ait feromen değerleri Denklem IV.6 ve Denklem IV.7’de verilen eşitlikler ile tekrar değiştirilmektedir.

$$\tau_{iyi_{ij}}(t) = \tau_{iyi_{ij}}(t-1) + \frac{a}{c_{iyi}} \quad (IV.6)$$

$$\tau_{kötü_{ij}}(t) = \tau_{kötü_{ij}}(t-1) - \frac{0.6a}{c_{kötü}} \quad (IV.7)$$

Bu denklemlerde verilen c_{iyi} , referans ile sistem çıkışı arasında oluşan en küçük RMS hatası iken $c_{kötü}$ ise en büyük RMS hatasını göstermektedir. Aynı tur içerisinde a/c_{iyi} ve $0.6a/c_{kötü}$ ise elde edilen feromen değişimlerini göstermektedir. Dolayısıyla tur içerisindeki en iyi karıncaya ait yola a/c_{iyi} kadar feromen daha eklenirken tur içerisindeki en kötü karıncadan $0.6a/c_{kötü}$ kadar feromen azaltılmaktadır. Karıncaların yolları izlemeleri sırasında zaman içerisinde kullanılmayan yollardaki feromen miktarları azaltılmaktadır. Bu sayede karıncanın yeni yollar denemesine olanak tanınmakta ve algoritmanın doğru çalışması sağlanmaktadır. Her bir turdan sonra yapılacak feromen buharlaştırma işlemi için kullanılacak eşitlik Denklem IV.8 ve Denklem IV.9’da verilmiştir.

$$\tau_{ij}(t) = \tau_{ij}(t-1)^\lambda + \Delta \quad (IV.8)$$

$$\Delta = \tau_{iyi} + \tau_{kötü} \quad (IV.9)$$

Burada;

λ : Buharlařma katsayısı

τ_{iyi} : En iyi karıncaya ait indisleri tutan matris

$\tau_{kötü}$: En kötü karıncaya ait indisleri tutan matris

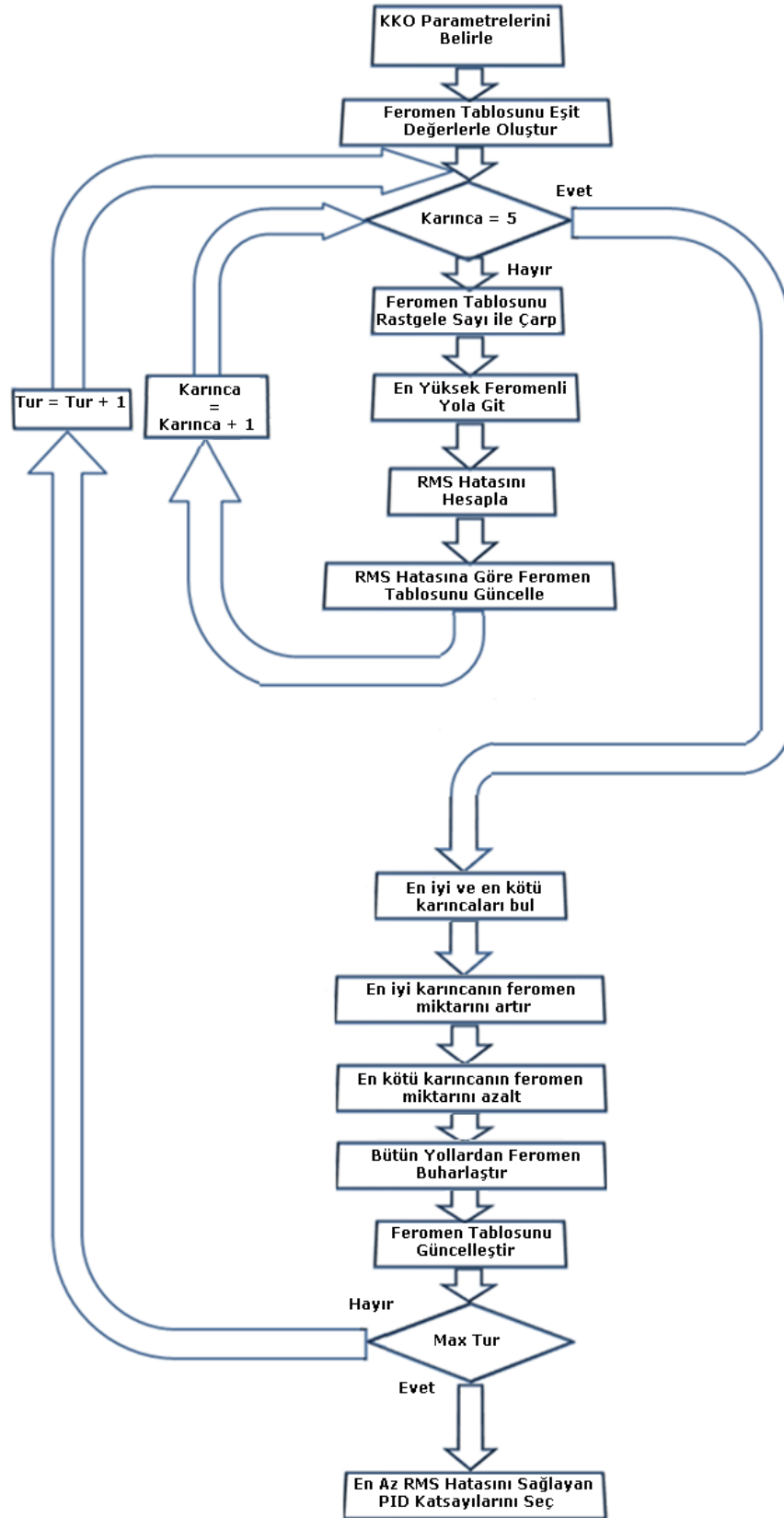
Δ : Her tur sonundaki feromen miktarını tutan matris

Burada KKA deęişkenlerinin (λ, a) deęerleri deneme yanılma ile verilmelidir. En kısa zamanda referans yörünge üzerinde sonuç veren deęerler o sistem için seçilen en uygun parametre deęerleridir. Benzer şekilde çıkış eğrisi referans yörüngeye yüksek doğrulukta oturmuştur[14].

IV.4.1. Kullanılan KKA Parametreleri

KKA ile PID kontrolörünün katsayıları elde edilirken seçilen KKA algoritmasının parametreleri deneme yanılma metodu ile bulunmuştur. Yapay karıncaların geçecekleri yuva, PID katsayıları olarak ayarlanmıştır. PID katsayılarının alt deęerleri 0, Üst deęerler verilen referansa göre deęişiklik göstermektedir. Ancak farklı üst deęerler verilerek yapılan çalışmalar sonucunda bulunan PID parametrelerinin sınır deęerleri $K_p=20$, $K_i=5$ ve $K_d=10$ olarak civarında bulunmuştur. Her bir PID parametresi alt limit ile üst limit arasında 100'er parçaya bölünerek, optimizasyon çalıştırıldığında yapay karıncaların geçebilecekleri yuvalar yani her turda bir karıncanın seçebileceęi K_p , K_i ve K_d deęerleri olarak ayarlanmıştır.

KKA ile PID kontrolörün katsayıları bulunurken izlenen adımlar Şekil IV.29' da gösterilmiştir. İlk olarak optimizasyon için KKA parametreleri belirlenmiştir. Yapılan çalışmada her turda 5 adet karınca olmak üzere toplam 100 tur gerçekleştirilmiştir. Karıncaların izledikleri yollardaki feromen deęerini artırmak için feromen sabiti $a= 0.06$ olarak seçilmiştir. Her tur sonunda karıncaların kullandıkları yollara bıraktıkları feromen miktarı, buharlařma sabiti $\lambda= 0.95$ ile buharlaştırılmıştır.



Şekil IV. 29 KKA ile PID kontrolör tasarımı için kullanılan algoritma

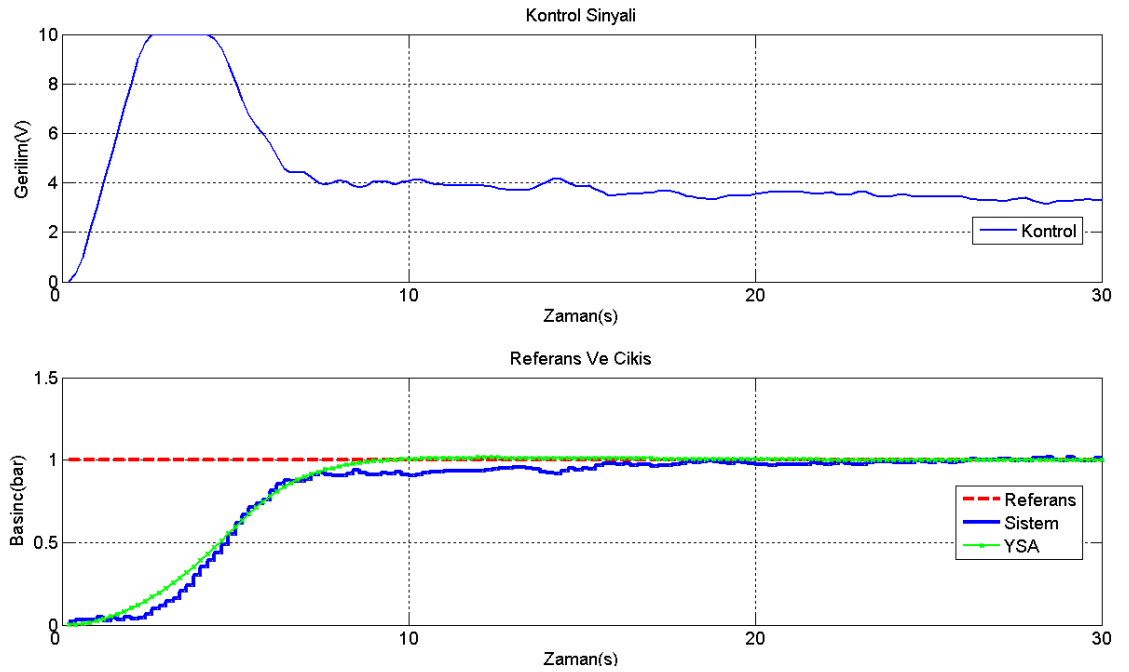
KKA parametreleri seçildikten sonra yapay karıncaların gidebilecekleri yolların hepsine eşit sayıda feromen değeri verilmiş ve bu değer feromen matrisinde tutulmuştur. İlk yapay karınca yuvadan çıktığında gidebileceği bütün yollarda eşit miktarda feromen olduğu için rastgele bir yol seçerek toplam üç yuvaya uğrayarak yolunu tamamlamıştır. Yolun sonunda seçilen PID katsayıları ile simülasyon çalıştırılmış ve referans ile simülasyon çıkışı arasındaki RMS hatası hesaplanarak bu yapay karıncanın geçmiş olduğu yolların feromen matrisindeki değeri artırılmıştır. Arkasından gelen yapay karıncanın da tur içinde aynı yoldan gitmemesi için feromen tablosu rastgele sayı ile çarpılarak güncellenmiştir. Böylece her turda karıncalar farklı PID katsayıları değerlerini alarak verilen referans yörüngeyi takip etmeye çalışmışlardır. 5 yapay karınca da turunu tamamlandığında bu turdaki en az rms hatasına sahip yapay karıncanın yolundaki feromen miktarı ekstra artırılmış ve en fazla RMS hatasına sahip yapay karıncanın yollundaki feromen miktarı iyi olan karıncanın artırılan miktarından daha fazla oranda azaltılmıştır. Ayrıca her tur sonunda bütün yapay karıncaların geçtikleri yollardaki feromen miktarı, buharlaşma sabiti(λ) ile buharlaştırılmıştır. Eğer seçilen PID katsayısı ile küçük RMS hatası elde edilmiş ise bir sonraki karıncalarda bu katsayılar etrafında dolanarak toplam turlarını bitirmeye çalışacaktır.

Maksimum tur tamamlandığında feromen tablosundaki en yüksek değere sahip yollar, yani seçilmiş olan PID katsayıları optimizasyon sonucu olarak kaydedilir.

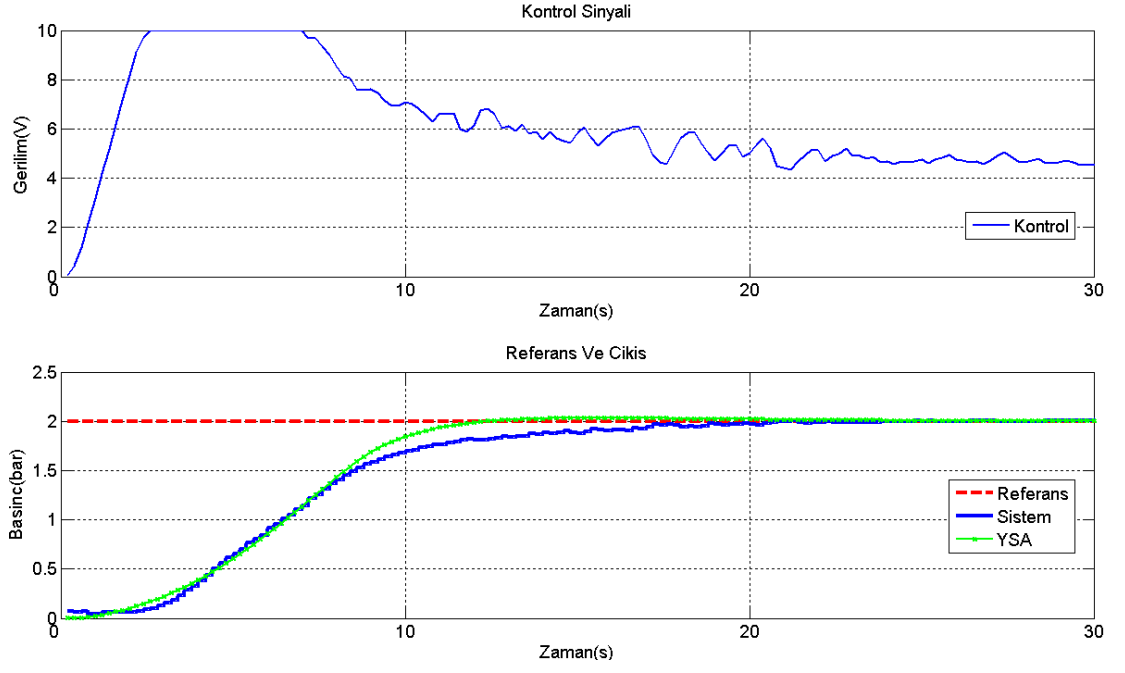
IV.5. KARINCA-PID KONTROLÖRÜN PERFORMANSI

IV.5.1. Basamak Girişlere Sistemin Davranışı

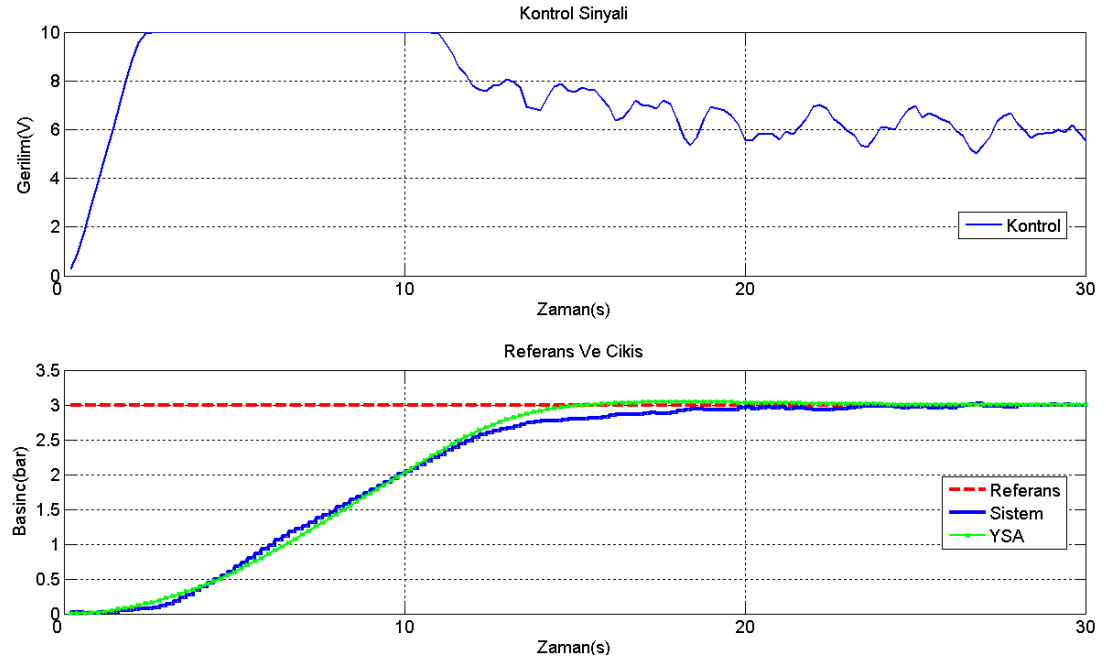
Sisteme; Şekil IV.30'da gösterildiği gibi 1bar, Şekil IV.31'de gösterildiği gibi 2bar, Şekil IV.32'de gösterildiği gibi 3bar, Şekil IV.33'de gösterildiği gibi 4bar değerleri referans olarak verilmiş, bu referans değerlerini takip eden YSA modelin ve sistemin çıkışları gösterilmiştir. Bu çıkışları sağlayan kontrol sinyalleri şekillerin üst kısmında gösterilmiştir. Kontrol sinyallerinin davranışlarından da anlaşılacağı gibi sistemin çıkışı yaklaşık referans değerlere yaklaşıncı, PID kontrolör sistemi maksimum sürmeyi bırakmış ve kararlı durum için kontrol sinyali üretmeye başlamıştır.



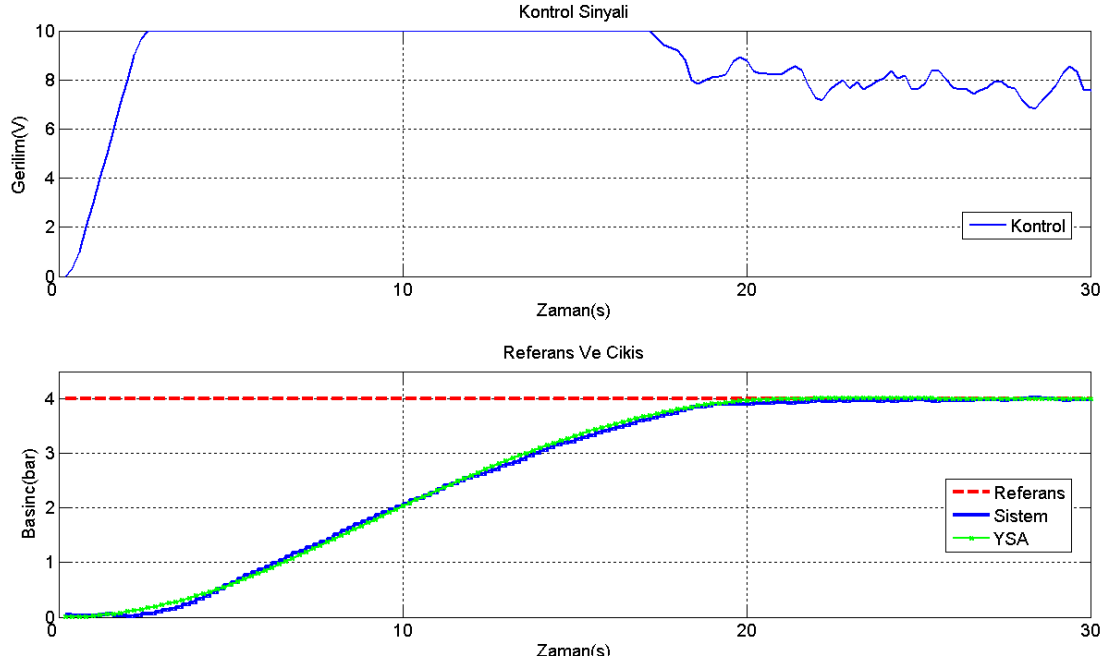
Şekil IV. 30 1bar referans girişe ait kontrol, YSA model ve gerçek sistem çıkışları



Şekil IV. 31 2bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları



Şekil IV. 32 3bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları

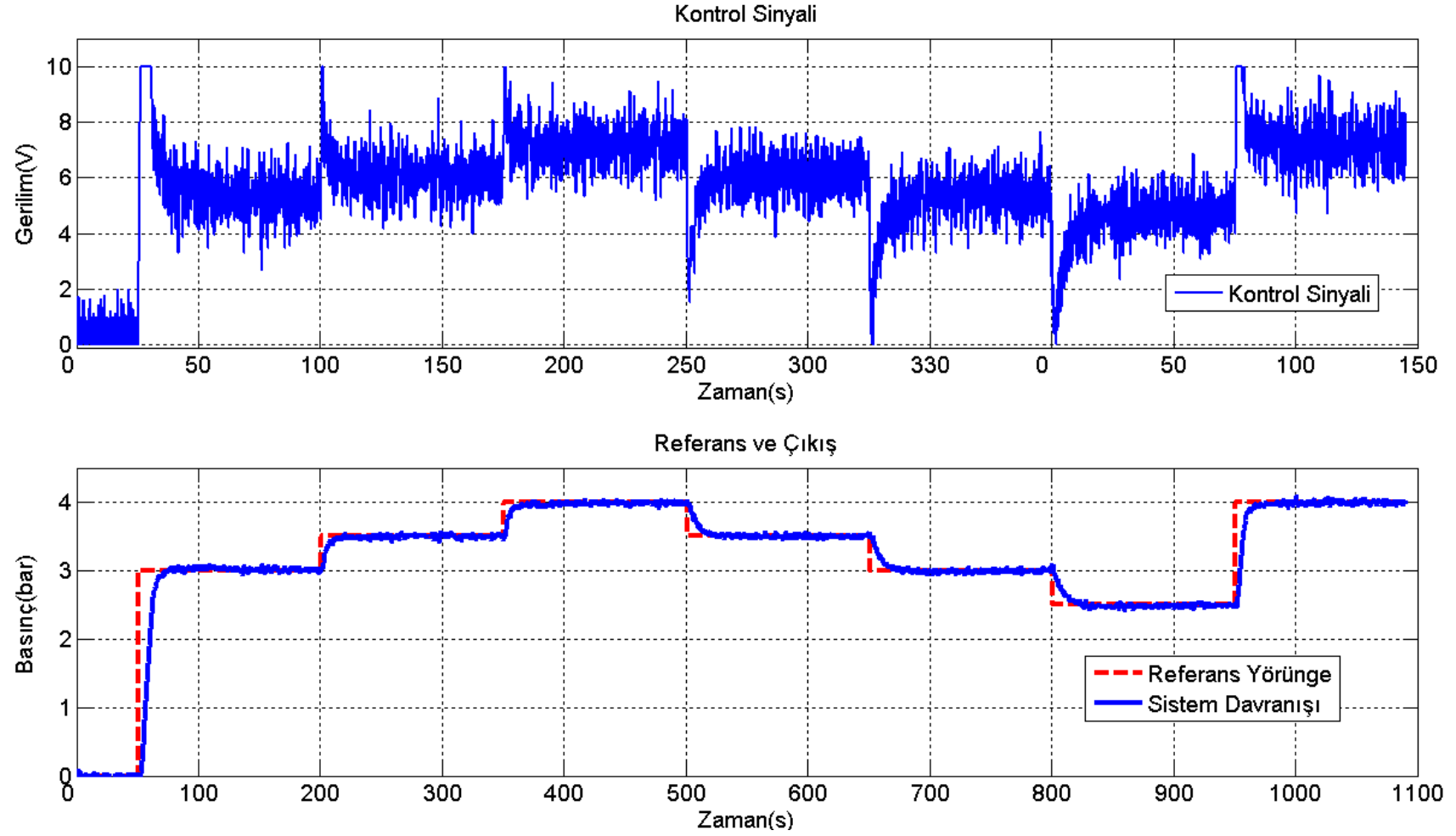


Şekil IV. 33 4bar referans girişe ait kontrol, YSA Model ve gerçek sistem çıkışları

IV.5.2. Basamak Girişler Arasında Geçiş Sistemin Davranışı

Bu çalışmada Karınca-PID kontrolörün, sistemin çalışma anında değişik referans değerine geçiş cevabını görebilmek için farklı referanslar vererek sistemin cevabı incelenmiştir. Çalışma sırasında sistem, ilk referans değerine oturduktan sonra referans değerini değiştirerek sistemin referanslar arasında geçişi incelenmiştir.

Şekil IV.34'de bu çalışmaya ait grafikler verilmiştir. Grafikten de anlaşılabacağı gibi sistem farklı basamak girişleri arasında geçişini başarılı bir şekilde gerçekleştirmiştir. Buradan da anlaşılabacağı gibi KKA ile bulunan PID katsayıları ile sistem gerçek zamanlı olarak başarılı bir şekilde kontrol edilmiştir.



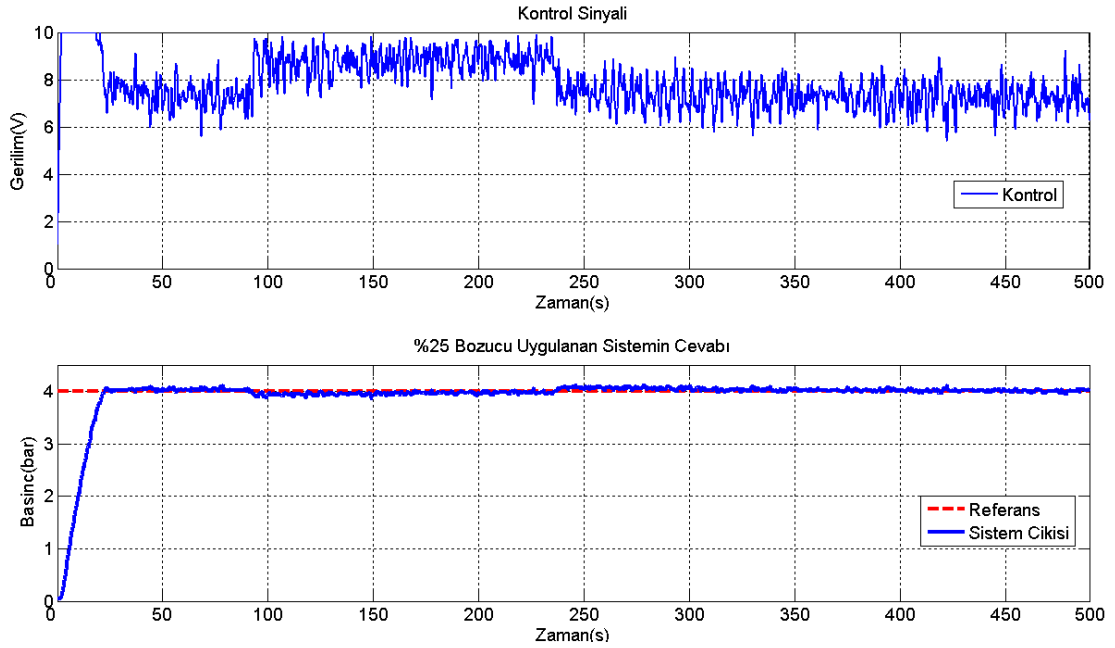
Şekil IV. 34 Basamak geçişler sırasında sistemin davranışı

IV.5.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı

Çalışmanın bu kısmında gerçekleştirilen Karınca-PID kontrolörün, basınç tankına dışarıdan ithal edilen bir bozucu etki karşısında performansı incelenecektir.

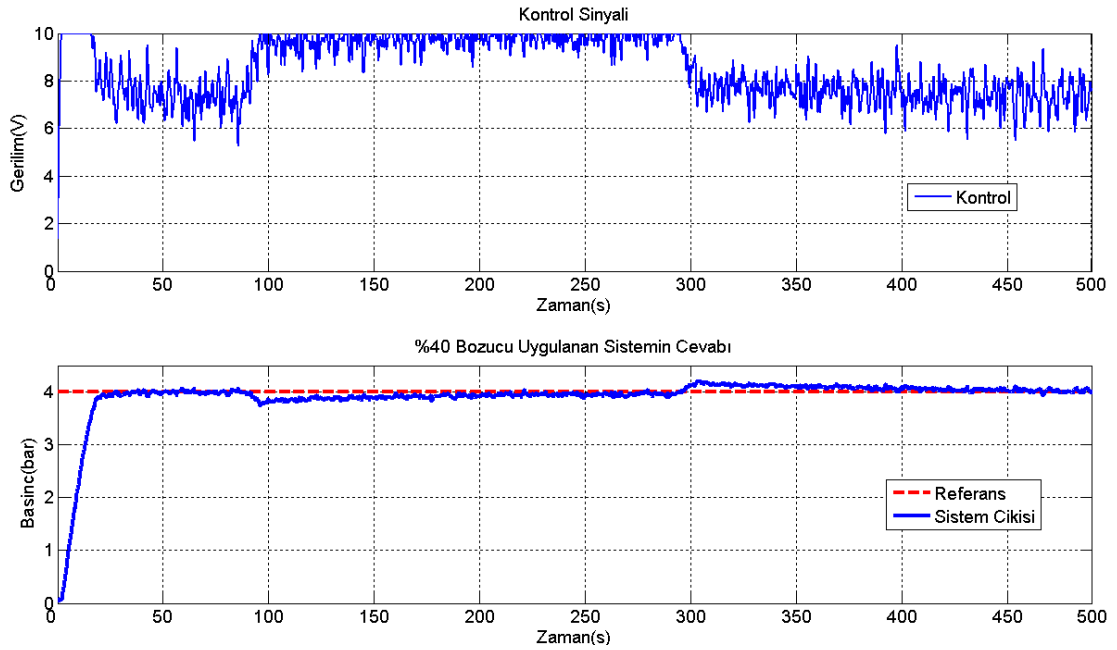
Bozucu ithal edilmesi ile çıkıştaki valfin açıklık değeri artırılarak bozucu ithal edilmiş olunur. Valfe 6V'tan fazla gerilim verilmesi ile tank içindeki havanın boşalma hızı da artacak ve sisteme bozucu ithal edilmiş olunacaktır. Bu bozucuya karşılık sistemin davranışları incelenecektir.

Şekil IV.35'de sistemin çıkışındaki oransal valfe %25 oranında daha fazla gerilim verilerek Karınca -PID kontrolörün davranışı incelenmiştir. Burada sisteme, 4bar referans değerine oturduktan sonra 150s boyunca bozucu giriş uygulanmış ve sistemin tekrar normal çalışma düzenine geçilmiştir. Bozucu giriş uygulandığında sistemin çıkışı 4bar basınç değerinin altına inmiş ancak kontrol sinyalinin 7.5V seviyesinden 9V'a yükselmesi ile sistemin çıkışı tekrar 4bar seviyesine gelmiştir. Bozucu kalktıktan sonra sistemin çıkışı 4bar'ın üzerine çıkmış ve kontrol sinyalinin 7.5V seviyesine gelmesi ile sistemin çıkışı tekrar 4bar'lık referans değerine oturmuştur.



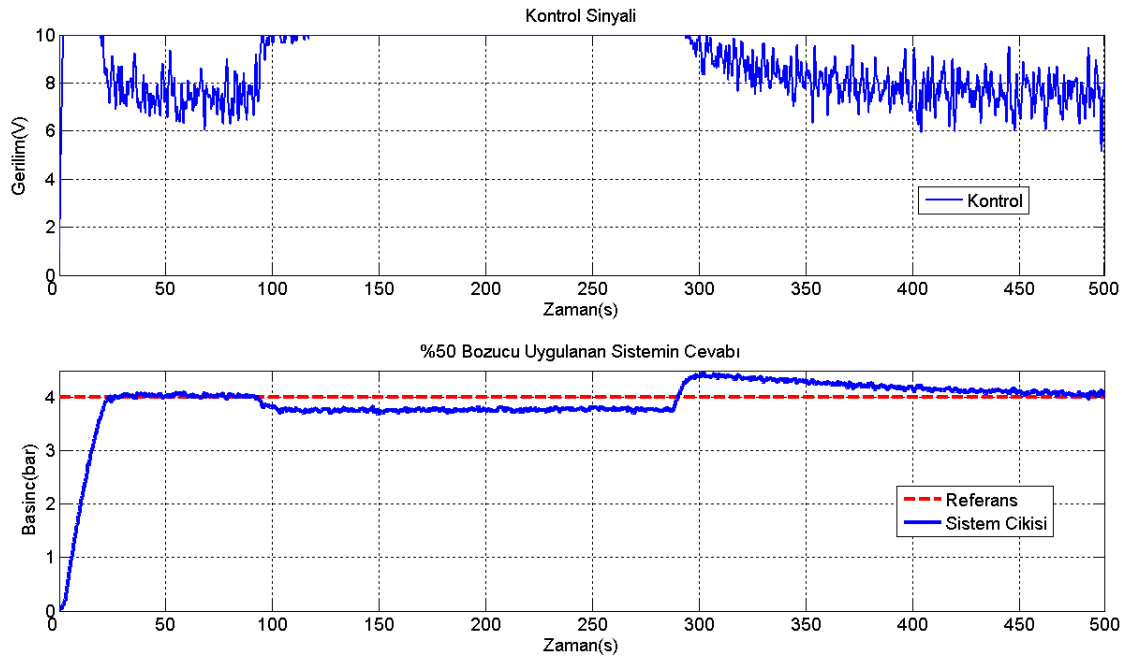
Şekil IV. 35 %25 bozucu uygulanan sistemin cevabı

Şekil IV.36'da sistemin çıkışındaki oransal valfe %40 oranında daha fazla gerilim verilerek Karınca -PID kontrolörün davranışı incelenmiştir. Burada sisteme, 4bar referans değerine oturduktan sonra 200s boyunca bozucu giriş uygulanmış ve sistemin tekrar normal çalışma düzenine geçilmiştir. Bozucu giriş uygulandığında sistemin çıkışı 4bar basınç değerinin altına inmiş ancak kontrol sinyalinin 7.5V seviyesinden 9.5V'a yükselmesi ile sistemin çıkışı tekrar 4bar seviyesine gelmiştir. Bozucu kalktıktan sonra sistemin çıkışı 4bar'ın üzerine çıkmış ve kontrol sinyalinin 7.5V seviyesine gelmesi ile sistemin çıkışı tekrar 4bar'lık referans değerine oturmuştur.



Şekil IV. 36 %40 bozucu uygulanan sistemin cevabı

Şekil IV.37’de sistemin çıkışındaki oransal valfe %50 oranında daha fazla gerilim verilerek Karınca-PID kontrolörün davranışı incelenmiştir. Burada sisteme, 4bar referans değerine oturduktan sonra 70s boyunca bozucu giriş uygulanmış ve sistemin tekrar normal çalışma düzenine geçilmiştir. Bozucu giriş uygulandığında sistemin çıkışı 4bar basınç değerinin altına inmiş ancak kontrol sinyalinin 10V yani %100 olması dahi sistemin referans değerine oturmasını sağlamamıştır. Bozucu kalktıktan sonra sistemin çıkışı 4bar’ın üzerine çıkmış ve kontrol sinyalinin 7.5V seviyesine gelmesi ile sistemin çıkışı 4bar’lık referans değerine tekrar oturmuştur.

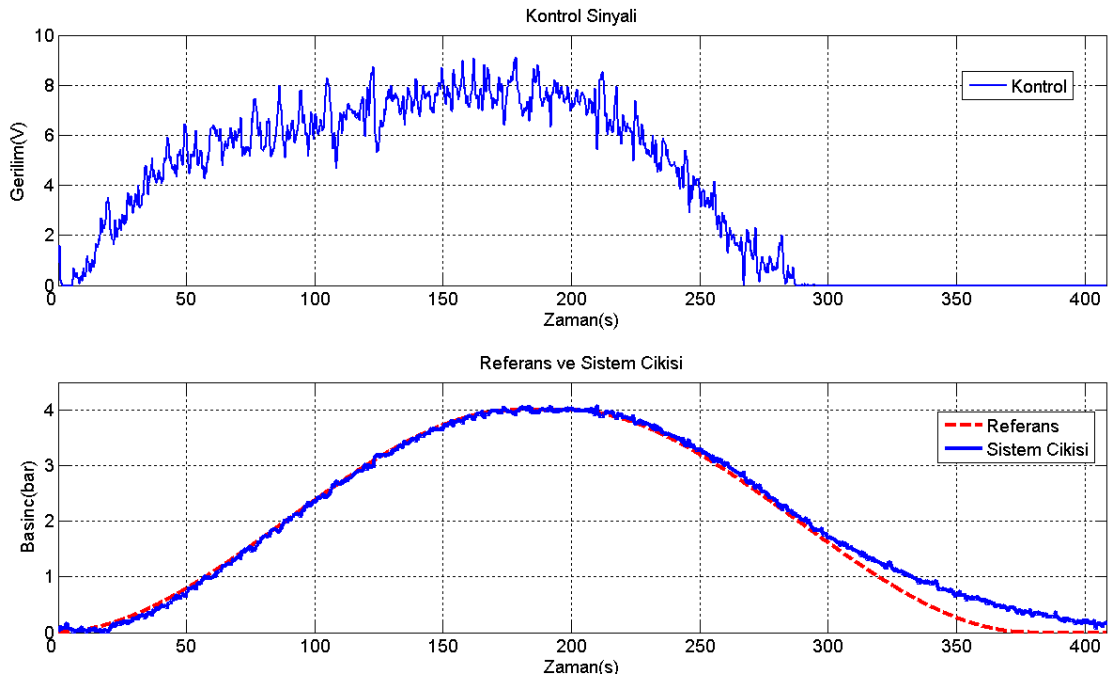


Şekil IV. 37 %50 bozucu uygulanan sistemin cevabı

IV.5.4. Karınca-PID Kontrolörün Yörünge Takip Cevapları

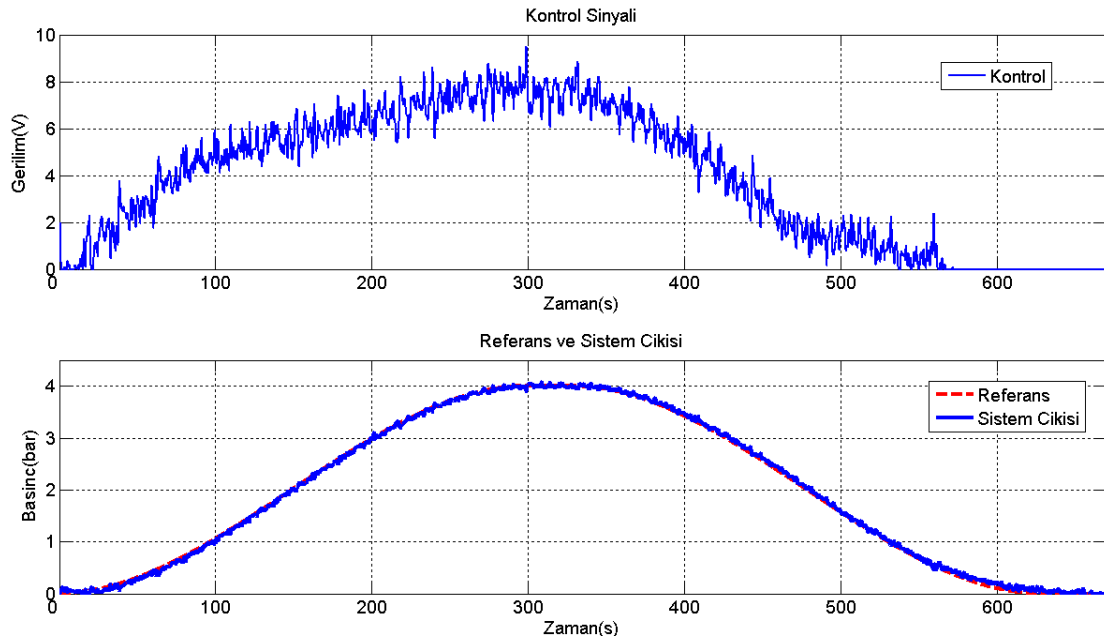
Çalışmanın bu kısmında Denklem II.4'de tanımlanan yörünge fonksiyonun, 0 bardan 4 bara kadar farklı eğim dereceleri ile oluşturulan yörüngelerin, sisteme referans yörünge olarak verilmesi ile sistemin davranışı incelenecektir.

İlk olarak sistemin, başlangıç değerinden 4 bara kadar takip etmesi istenmiştir. Bunun için eğime 6 değeri verilerek yörünge oluşturulmuş ve sisteme referans olarak verilmiştir. Sistem 4 bara geldiğinde ise aynı eğimle ilk konumuna gelmesi için gerekli yörünge oluşturularak sistemin boşalması sırasındaki cevabı incelenmiştir. Sistemin cevabı Şekil IV.38'de gösterilmektedir. Sistemin dolum sırasında yörüngeyi başarılı bir şekilde takip ettiği görülmüş ancak 300s sonrasında kontrol sinyalinin 0 olmasına rağmen sistemin doğal boşalması yörüngeyi takip etmek için yeterli olmadığı için son kısımlarda yörünge ile sistem davranışı arasında fark oluşmuştur.



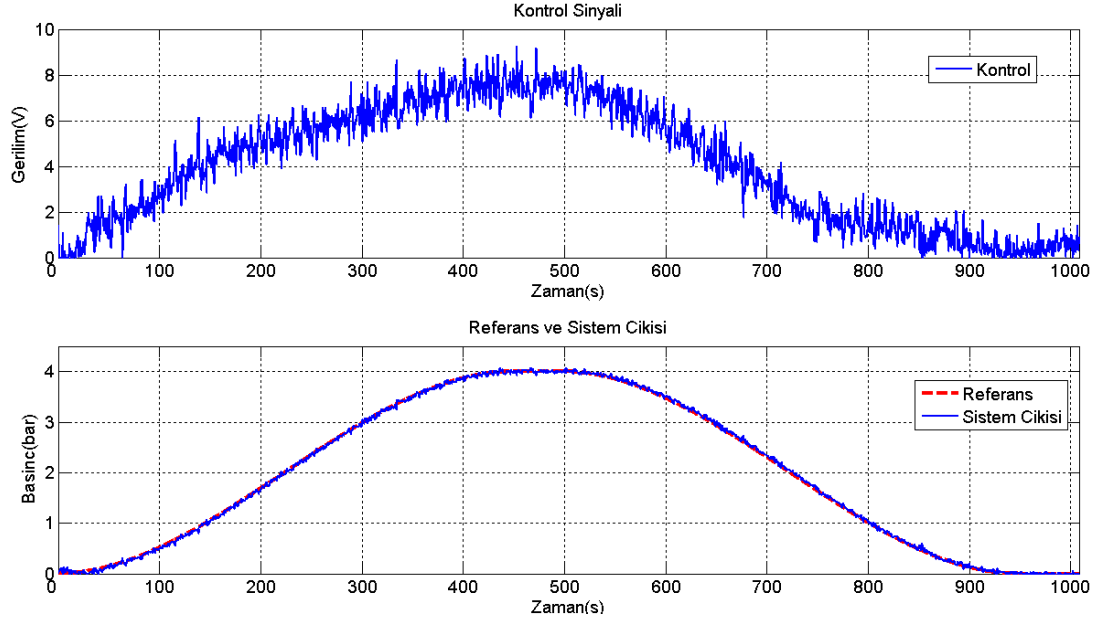
Şekil IV. 38 Eğimi 6 olan referans yörünge ve sistemin takibi

Dolum ve boşaltım referans yörüngeleri, eğim 10 yapılarak birleştirilmiş ve sisteme referans olarak verilmiştir. Bu referans yörüngeye sistem, Şekil IV.39’da gösterildiği gibi bir davranış göstermiştir. Sistemin, dolum ve boşaltım sırasında yörüngeyi başarılı bir şekilde takip ettiği grafikten anlaşılmaktadır. Ancak 560s civarında kontrolörün çıkış sinyali olarak 0V üretmesine rağmen sistemin doğal davranışı, yörüngeyi boşalma hızına yetişememiş ve küçük hata ile yörünge takip edilmiştir.



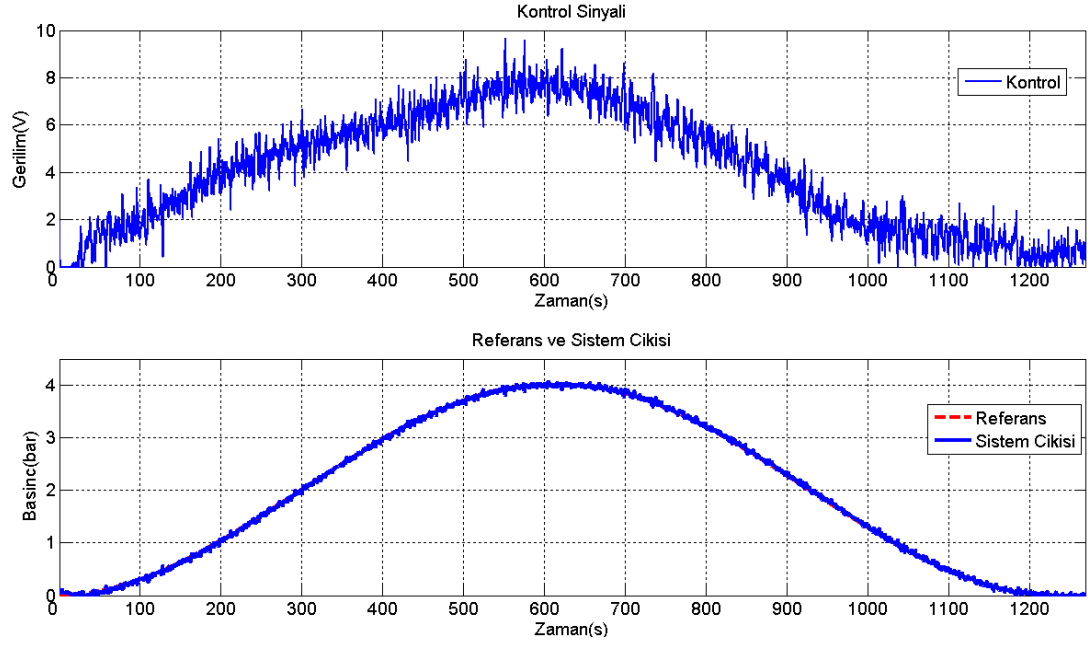
Şekil IV. 39 Eğimi 10 olan referans yörüngeyi sistemin tarafından takibi

Dolum ve boşaltım referans yörüngeleri, eğim 15 yapılarak birleştirilmiş ve sisteme referans olarak verilmiştir. Bu referans yörüngeye sistem, Şekil IV.40'da gösterildiği gibi bir davranış göstermiştir. Sistemin, dolum ve boşaltım sırasında yörüngeyi başarılı bir şekilde takip ettiği grafikten anlaşılmaktadır.



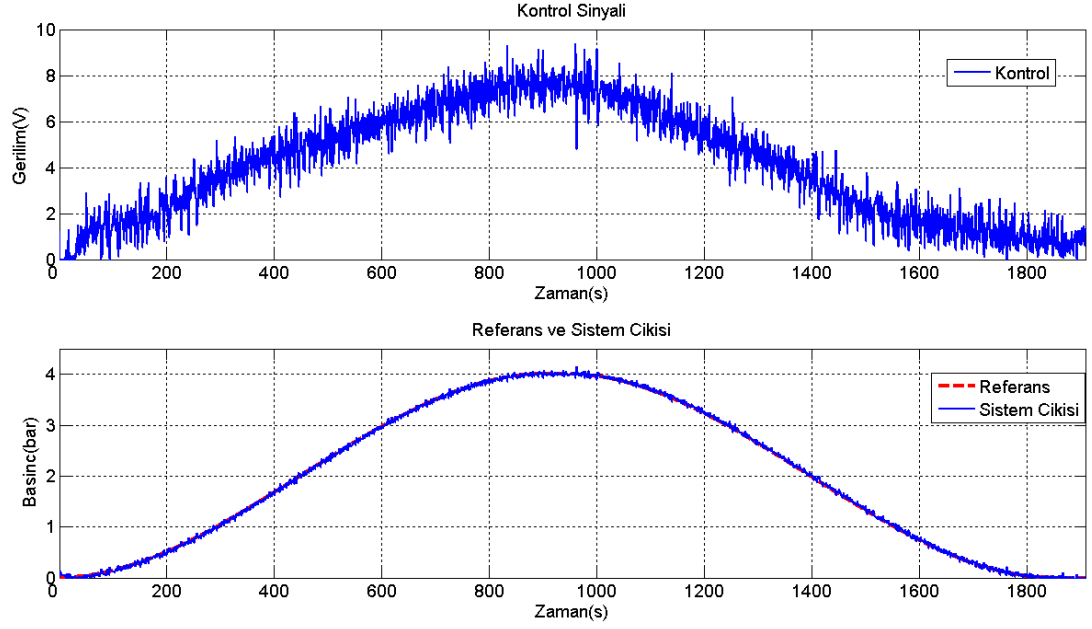
Şekil IV. 40 Eğimi 15 olan referans yörüngenin sistem tarafından takibi

Dolum ve boşaltım referans yörüngeleri, eğim 20 yapılarak birleştirilmiş ve sisteme referans olarak verilmiştir. Bu referans yörüngeye sistem, Şekil IV.41’de gösterildiği gibi bir davranış göstermiştir. Sistemin, dolum ve boşaltım sırasında yörüngeyi başarılı bir şekilde takip ettiği grafikten anlaşılmaktadır.



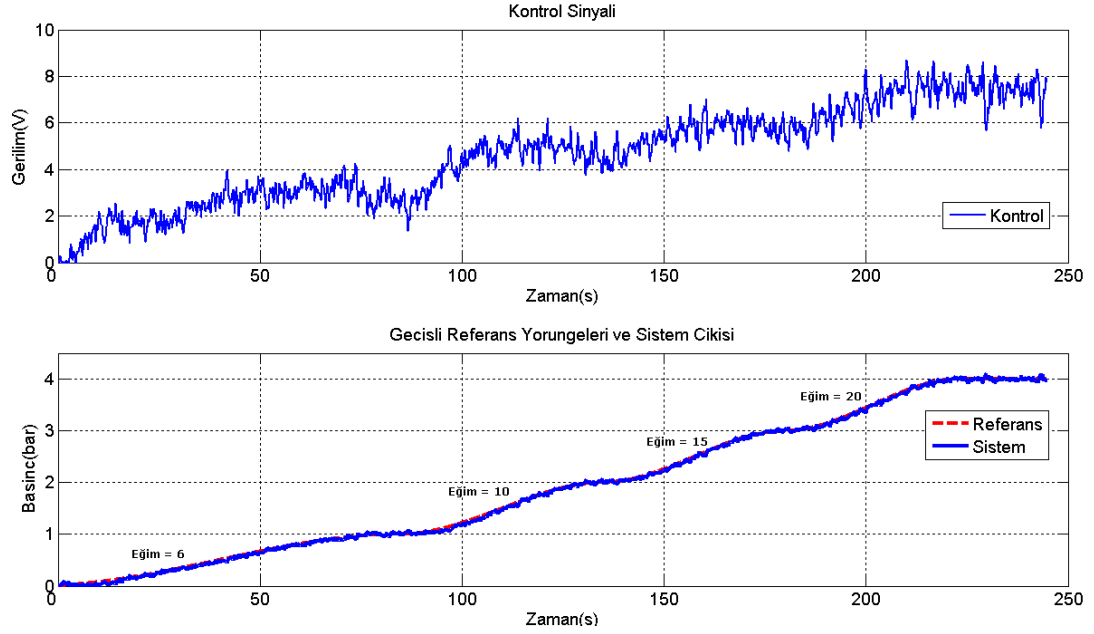
Şekil IV. 41 Eğimi 20 olan referans yörüngeyi sistemin tarafından takibi

Dolum ve boşaltım referans yörüngeleri, eğim 30 yapılarak birleştirilmiş ve sisteme referans olarak verilmiştir. Bu referans yörüneye sistem, Şekil IV.42’de gösterildiği gibi bir davranış göstermiştir. Sistemin, dolum ve boşaltım sırasında yörüneyi başarılı bir şekilde takip ettiği grafikten anlaşılmaktadır.



Şekil IV. 42 Eğimi 30 olan referans yörüngenin sistem tarafından takibi

Karınca-PID kontrolörlü sistemin farklı eğimli yörünge değerleri arasında geçişi sırasında başarısını görmek için Şekil IV.43’de gösterildiği gibi referans yörünge verilmiştir. Şekil IV.43’den anlaşılabacağı gibi sistem yörüngeler arasında başarılı bir şekilde geçiş sağlamıştır.

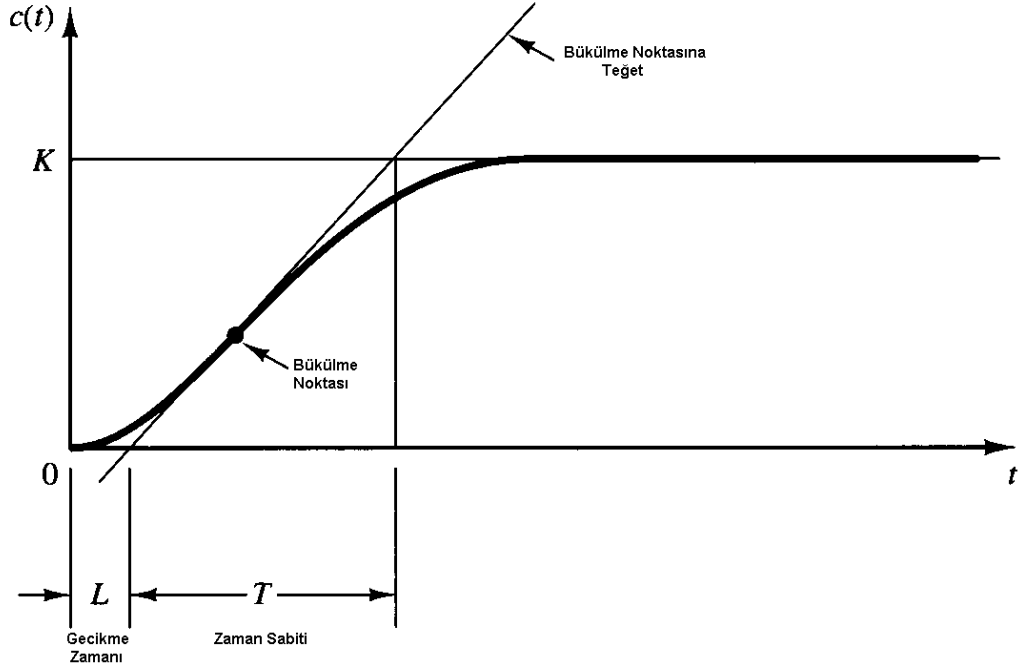


Şekil IV. 43 Yörüngeler arası geçişe sistemin cevabı

Elde edilen grafiklerden de anlaşılabacağı gibi, çevrim dışı çalışmada KKA ile optimize edilen PID kontrolör ile sistemin gerçek zamanlı kontrolü farklı referanslar uygulansa dahi başarılı bir şekilde gerçekleştirilmektedir.

IV.6. ZIEGLER- NICHOLS(ZN) YÖNTEMİYLE PID KATSAYILARININ BULUNMASI

Bu yöntemde sistemin basamak girişe karşılık %25'lik aşım yapacağı göz önüne alınarak hesaplamalar yapılmaktadır. Sisteme belirli bir değerde sabit basamak giriş uygulandığında, sistemin cevabı Şekil IV.44'de gösterilen grafiğe benziyorsa bu metot kullanılır.



Şekil IV. 44 Sabit genlikli basamak girişe sistemin cevabı

Sistemin cevabından L ve T değerleri tespit edilerek Tablo IV.2'ye göre PID katsayıları hesaplanır.

$$TF = \frac{K e^{-Ls}}{Ts + 1} \quad (IV.10)$$

Buradaki L ifadesi gecikme zamanını, T ifadesi ise zaman sabitini sembolize etmektedir[16].

Tablo IV. 2 Ziegler-Nichols 1.yöntem katsayı tablosu

Kontrolör Tipi	Kp	Ti	Td
P	T/L	∞	0
PI	0.9T/L	L/0.3	0
PID	1.2T/L	2L	0.5L

$$Kd = Kp \frac{1}{T_i} \quad (IV.11)$$

$$Ki = Kp * T_d s \quad (IV.12)$$

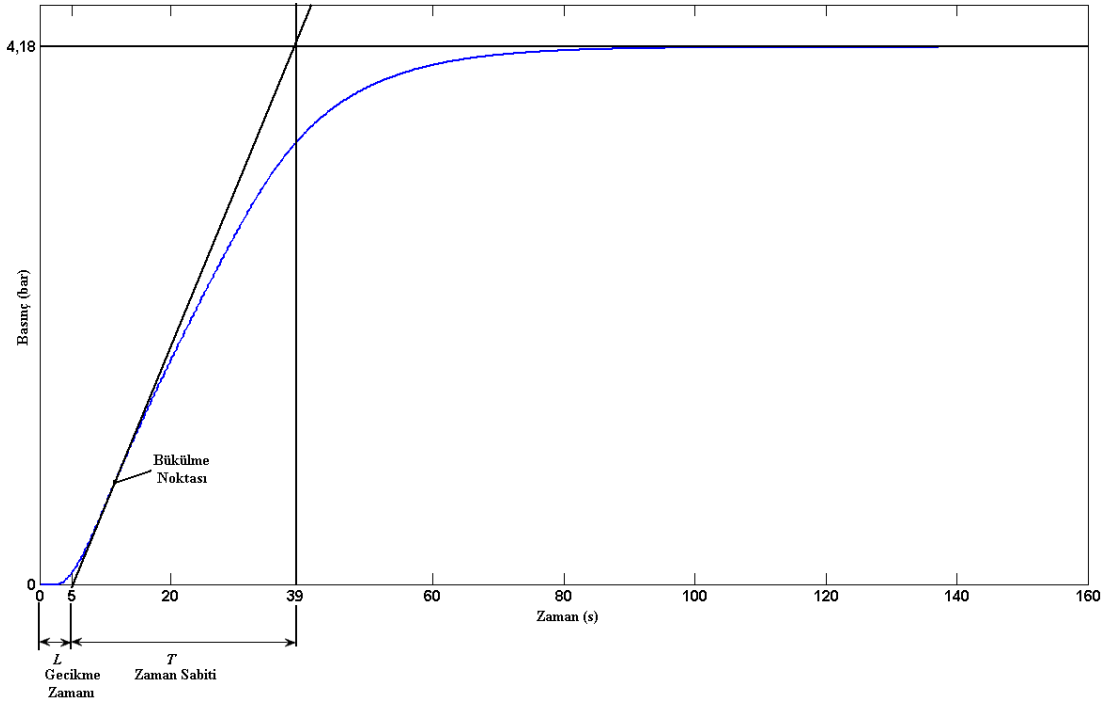
Sisteme sabit genlikli basamak giriş uygulandığında sistemin cevabı Şekil IV.45’de gösterilmektedir. Burada sistemin basınç dolum eğrisinde, dönüm noktasına teğet çizildiğinde gecikme zamanı(L) 5s ve zaman sabiti(T) 39s olarak tespit edilmiştir. Elde edilen L ve T değerleri Tablo IV.2’de yerine yazıldığında;

$$K_p = 8.16$$

$$K_i = 0.1$$

$$K_d = 2.5$$

olarak bulunmuştur.



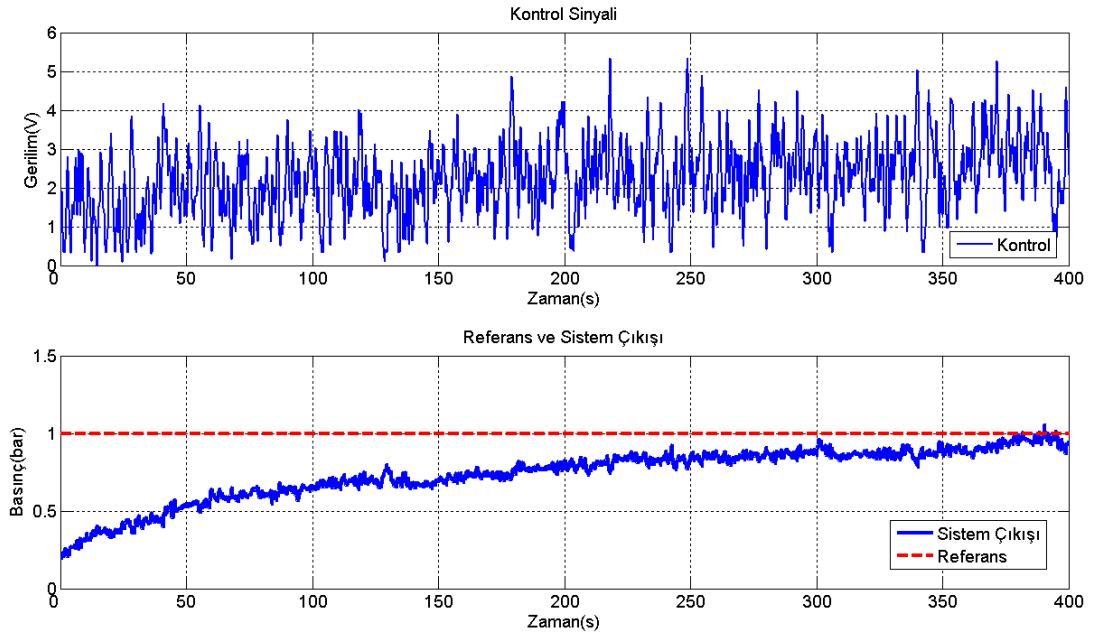
Şekil IV. 45 Basamak giriş sistemin cevabı

IV.6. ZN-PID KONTROLÖRÜN PERFORMANSI

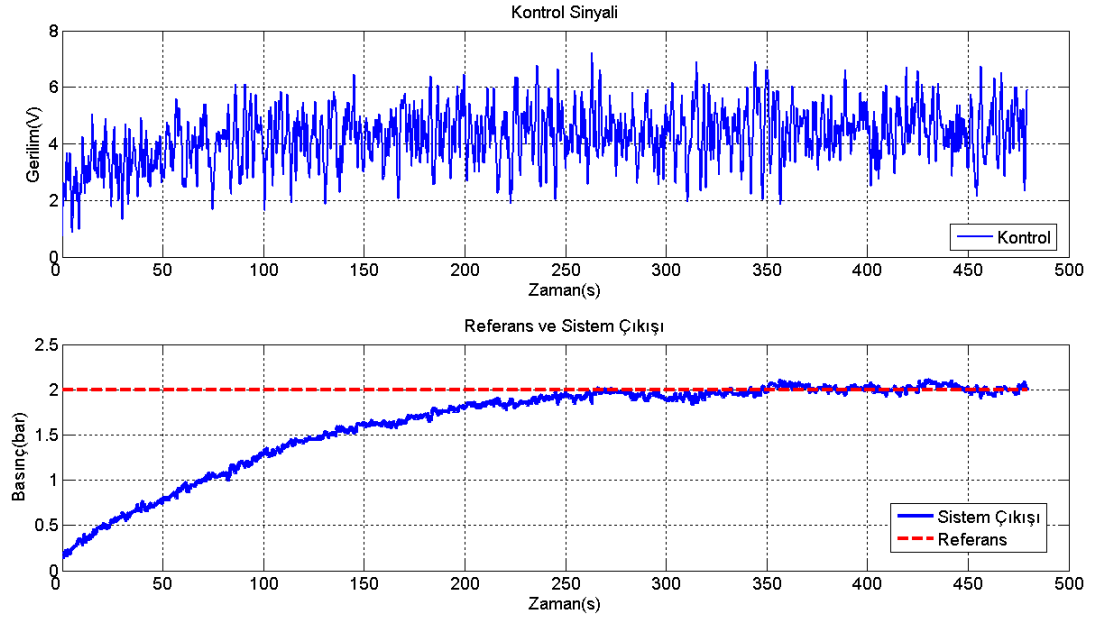
Bu kısımda YSA modeli üzerinden ZN metodu ile yapılan optimizasyon sırasında elde edilen ZN-PID kontrolörün gerçek zamanlı çalışmalarda performansı incelenmiştir. Yapılan çalışmalardan anlaşılabacağı gibi ZN metoduyla bulunan PID katsayıları ile gerçek zamanlı olarak sistem kontrolünde kullanıldığında, ZN yönteminin PID kontrolörü yeterince optimize edemediği anlaşılmaktadır.

IV.6.1. Basamak Girişlere Sistemin Davranışı

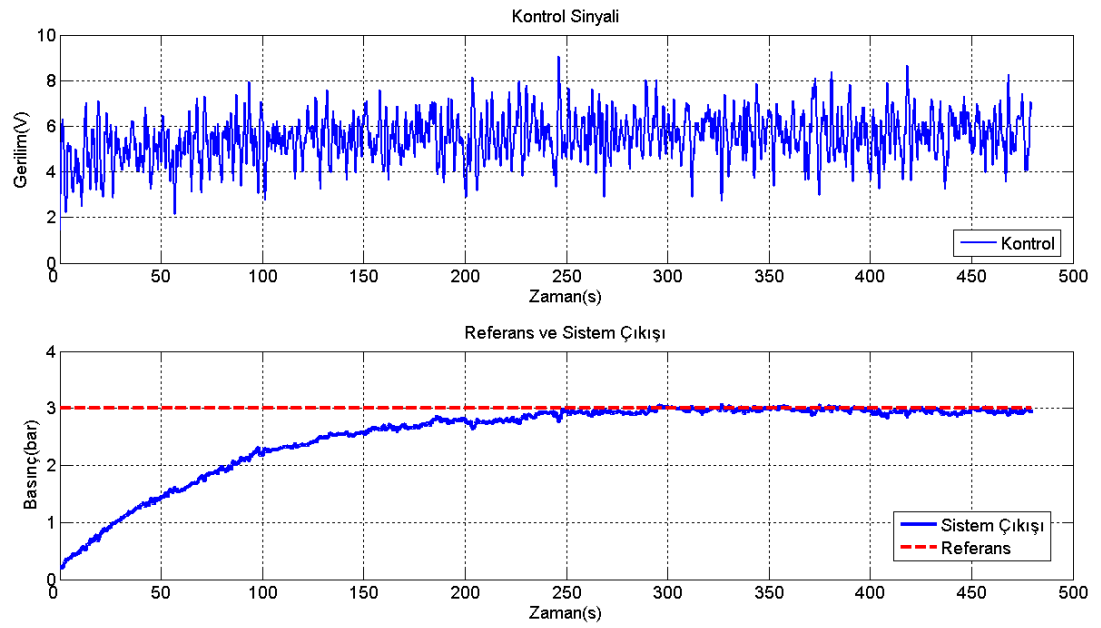
Sisteme; Şekil IV.46'da gösterildiği gibi 1bar, Şekil IV.47'de gösterildiği gibi 2bar, Şekil IV.48'de gösterildiği gibi 3bar, Şekil IV.49'da gösterildiği gibi 4bar değerleri referans olarak verilmiş, bu referans değerlerini takip eden gerçek sistemin çıkışları gösterilmiştir. Bu çıkışları sağlayan kontrol sinyalleri şekillerin üst kısmında gösterilmiştir. Kontrol sinyallerinin davranışlarından da anlaşılabacağı gibi sistemin çıkışı ile referans değer arasındaki hatanın büyük olmasına rağmen PID kontrolör sistemi sürememekte ve sistem referansa çok geç oturmaktadır.



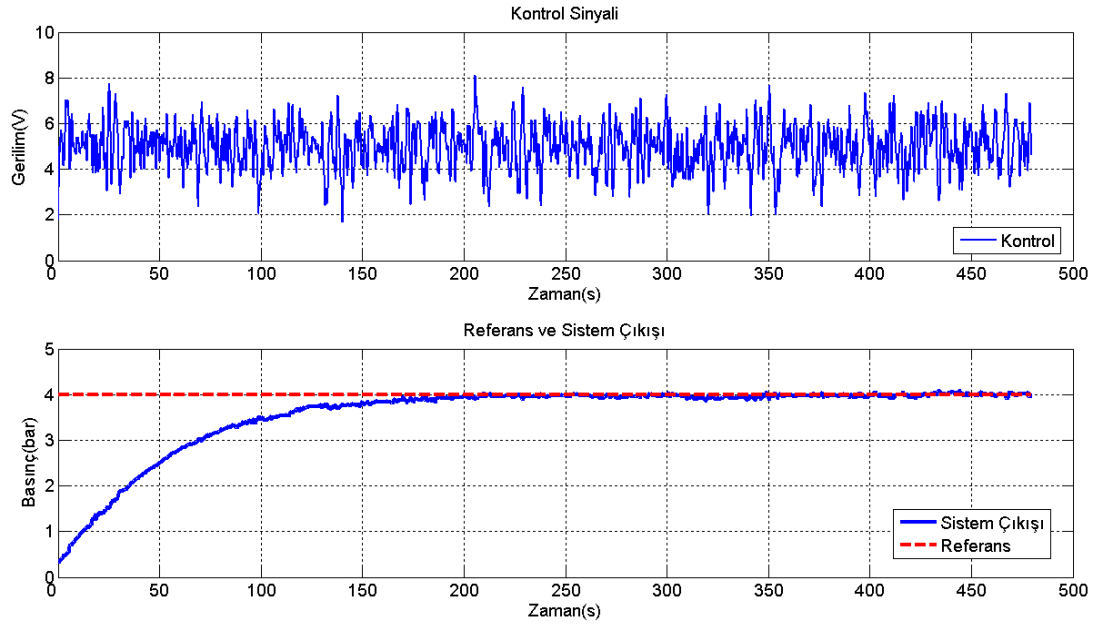
Şekil IV. 46 1bar referans girişi ait kontrol ve gerçek sistem çıkışları



Şekil IV. 47 2bar referans girişe ait kontrol ve gerçek sistem çıkışları



Şekil IV. 48 3bar referans girişe ait kontrol ve gerçek sistem çıkışları

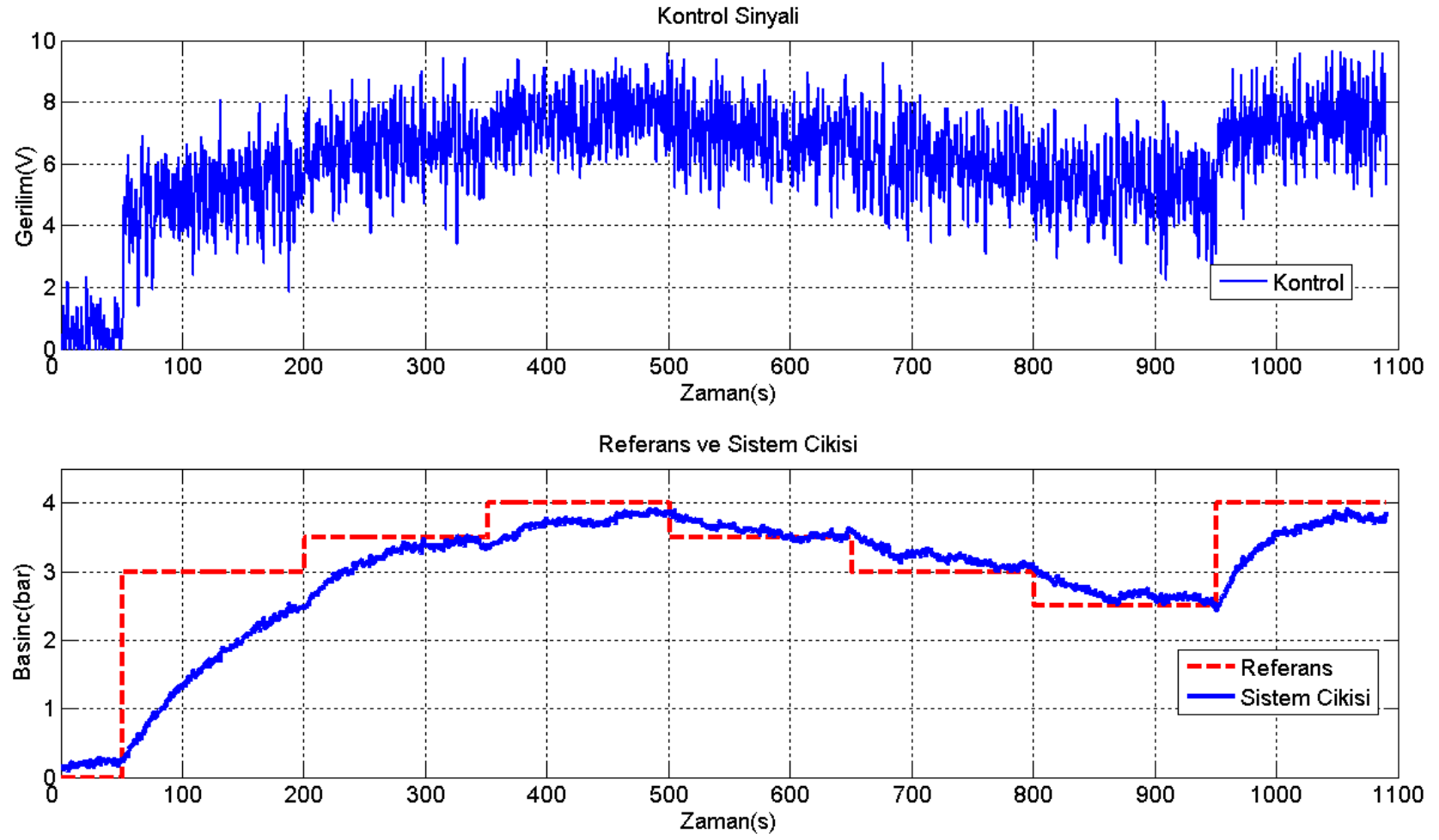


Şekil IV. 49 4bar referans girişi ait kontrol ve gerçek sistem çıkışları

IV.6.2. Basamak Girişler Arasında Geçiş Sistemin Davranışı

Bu çalışmada ZN-PID kontrolörün, sistemin çalışma anında değişik referans değerine geçiş cevabını görebilmek için değişen referanslar ile sistemin cevabı incelenmiştir. Çalışma sırasında ZN_PID kontrolörlü sistem, ilk referans değerine oturduktan sonra referans değerini değiştirerek sistemin referanslar arasında geçişi incelenmiştir.

Şekil IV.50' de bu çalışmaya ait grafikler verilmiştir. Grafikten de anlaşılabacağı gibi sistem farklı basamak girişleri arasında geçişini istenilen şekilde gerçekleştirememiştir.



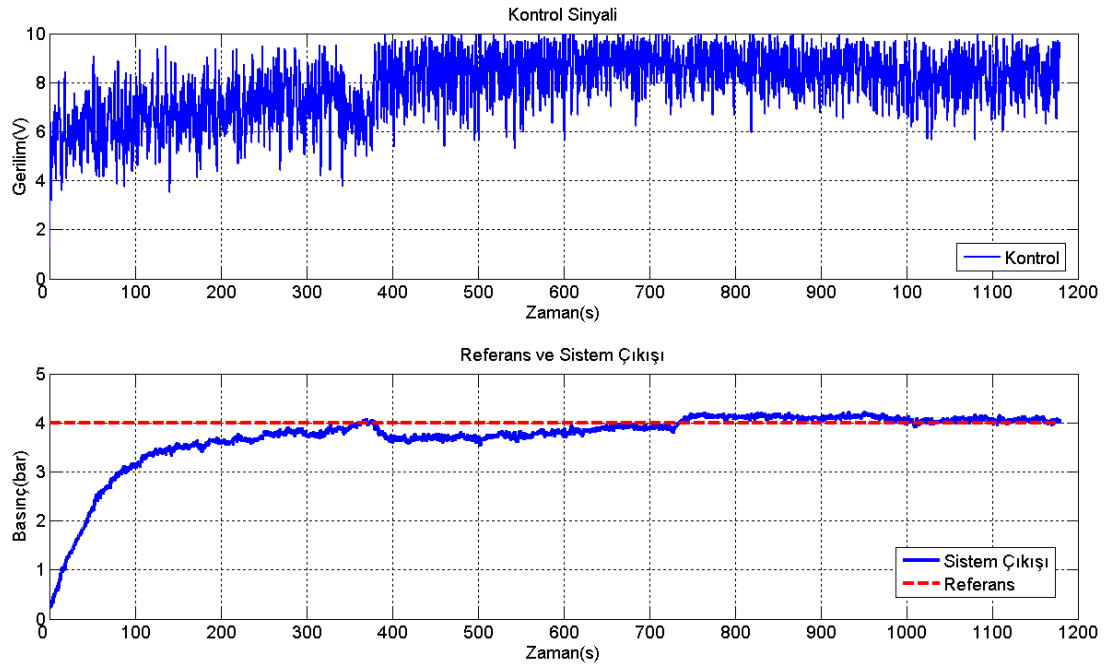
Şekil IV. 50 Basamak geçişler sırasında gerçek sistemin davranışı

IV.6.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı

Çalışmanın bu kısmında gerçekleştirilen ZN-PID kontrolörün, basınç tankına dışarıdan ithal edilen bir bozucu etki karşısında performansı incelenecektir.

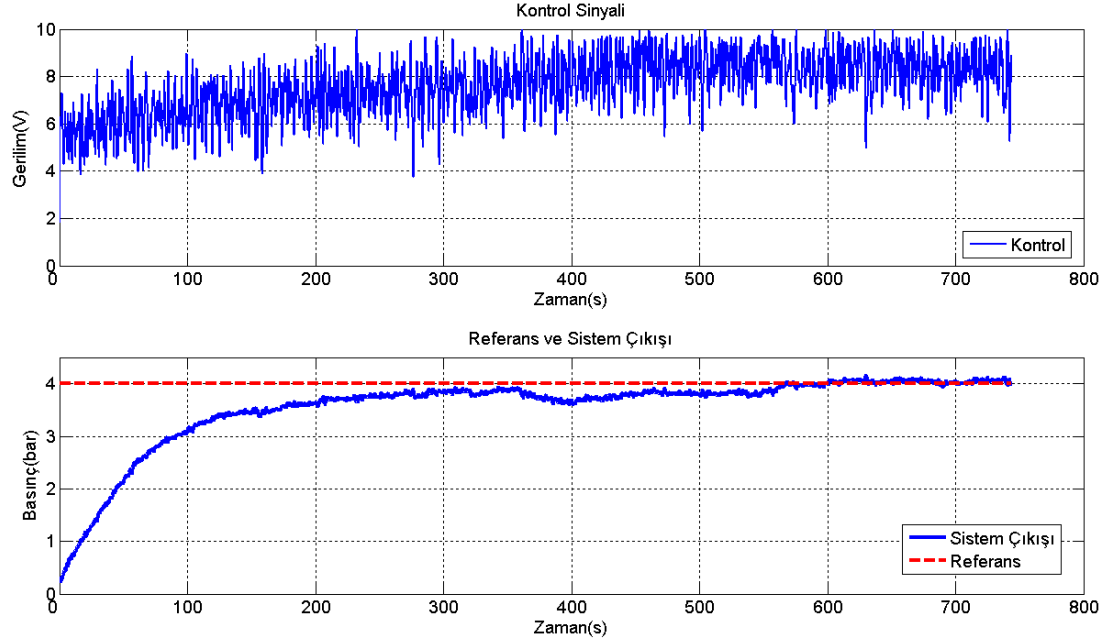
Sistemin çalışması sırasında basınç tankının çıkışında bulunan pnomatik oransal valfe 6V gerilim uygulanarak valfin %50 oranında açılması sağlanmaktadır. Böylece sistem çalışırken fazla havanın boşaltılması gerçekleştirilmektedir. Bozucu ithal edilmesi ile çıkıştaki valfin açıklık değeri artırılarak bozucu ithal edilmiş olunur. Valfe 6V'tan fazla gerilim verilmesi ile tank içindeki havanın boşalma hızı da artacak ve sisteme bozucu ithal edilmiş olunacaktır. Bu bozucuya karşılık sistemin davranışları incelenecektir.

İlk olarak ZN-PID kontrolörlü sisteme verilen referansa oturduktan sonra %15 bozucu giriş Şekil IV.51'de gösterildiği gibi 350s sonra uygulanmıştır. Bozucu giriş uygulandıktan 340s sonra sistem referans değerine oturabilmiştir. Bozucu giriş kaldırıldıktan sonra sistem tekrar referans değerine 260s sonra oturmuştur.



Şekil IV. 51 %15 bozucu uygulanan sistemin cevabı

ZN-PID kontrolörlü sisteme %25 bozucu giriş uygulandığında sistem Şekil IV.52'de gösterildiği gibi bir davranış göstermiştir. Bozucu giriş 120s uygulanmış ancak sistem bozucu giriş uygulandıktan sonra verilen referans değerini kalıcı hal hatası ile takip edebilmiştir. Bozucu giriş kaldırıldıktan sonra sistem 180s sonra tekrar referans değerine oturmuştur.



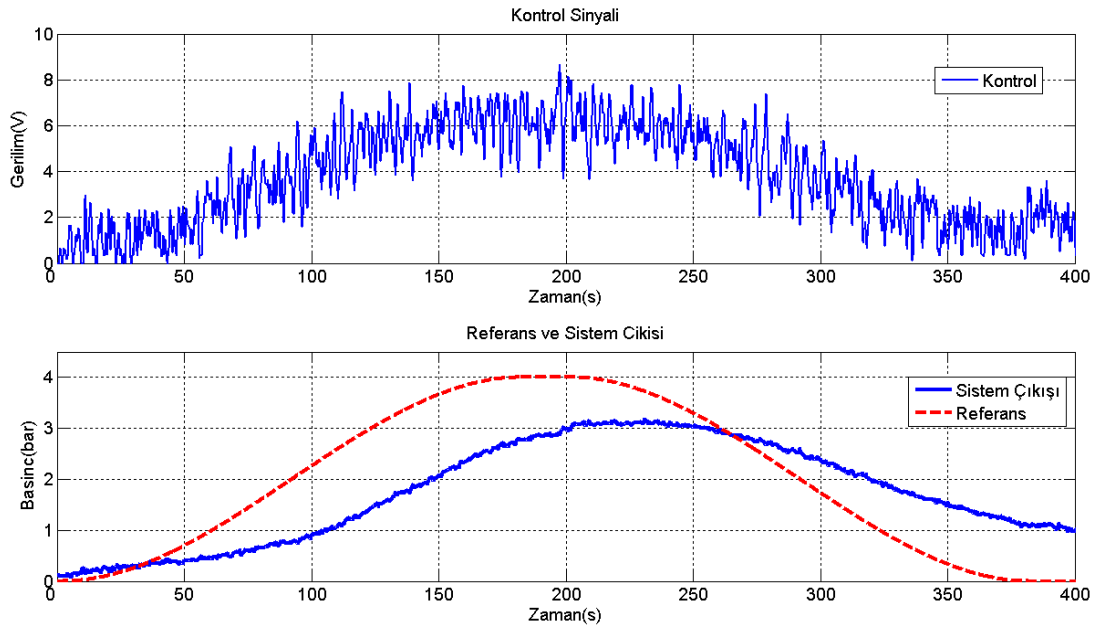
Şekil IV. 52 %25 bozucu uygulanan sistemin cevabı

IV.6.3. ZN-PID Kontrolörün Yörünge Takip Cevapları

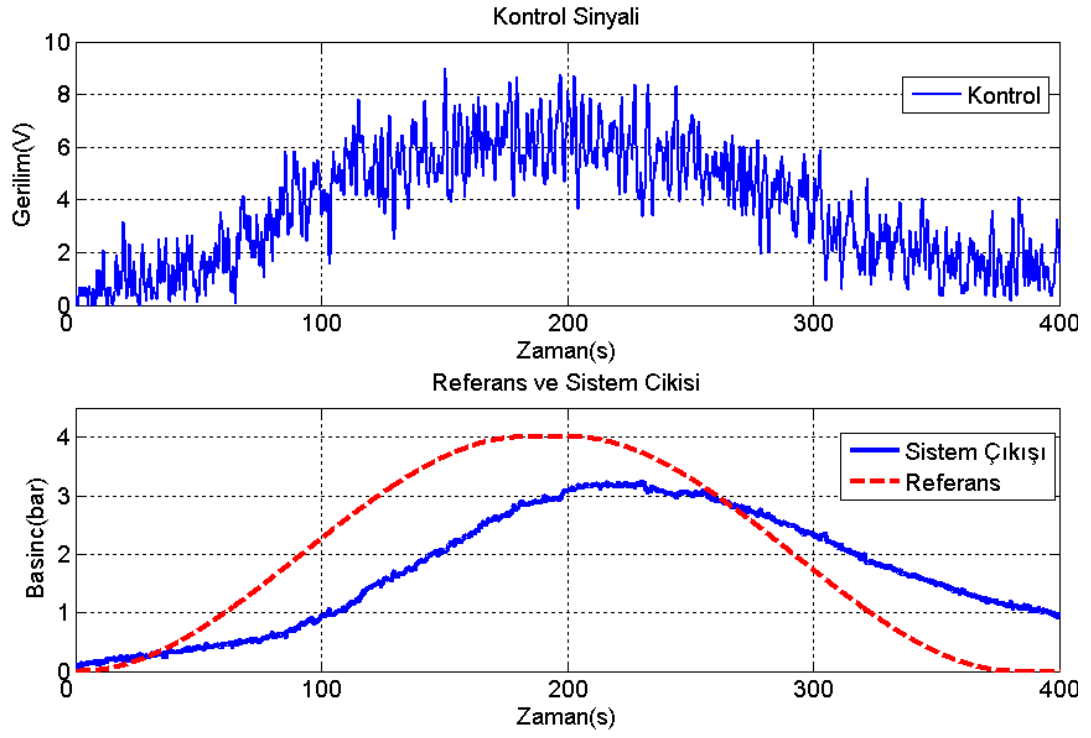
Çalışmanın bu kısmında Denklem II.4'de tanımlanan yörünge fonksiyonun, 0 bardan 4 bara kadar farklı eğim dereceleri ile oluşturulan yörüngelerin, sisteme referans yörünge olarak verilmesi ile sistemin davranışı incelenecektir.

Sistemin, başlangıç değerinden 4 bara kadar farklı eğimlerle takip etmesi istenmiş ve bunun için gerekli yörünge oluşturularak sisteme referans olarak verilmiştir. Sistem 4 bara geldiğinde ise aynı eğimle ilk konumuna gelmesi için gerekli yörünge oluşturularak sistemin boşalması sırasındaki cevabı incelenmiştir.

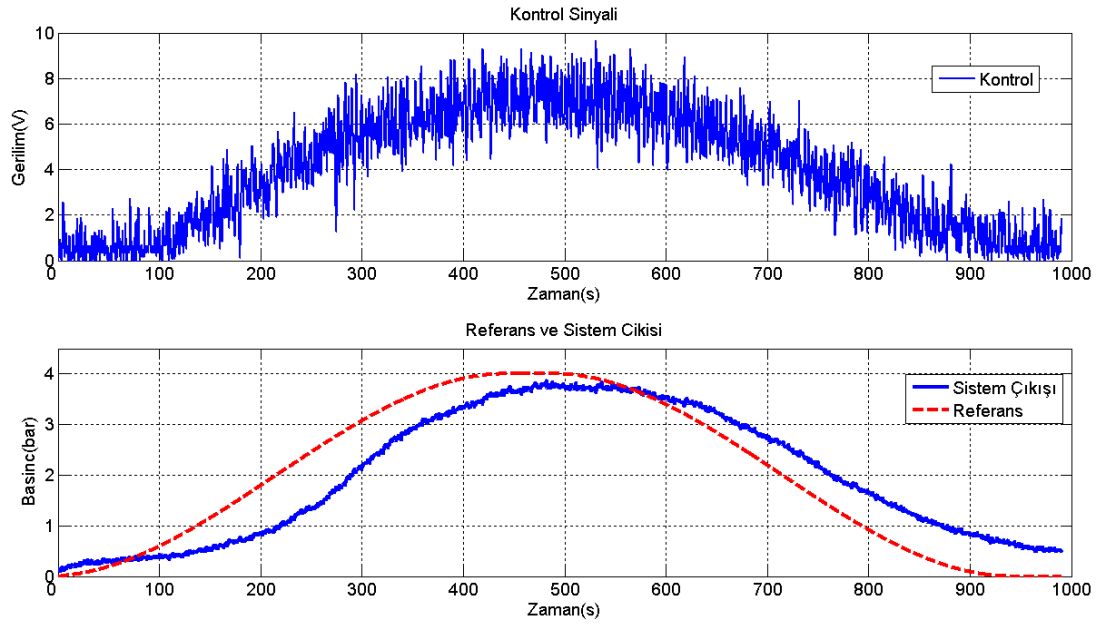
Sistemin, başlangıç değerinden 4 bara kadar sırasıyla 6, 10, 15, 20 ve 30 değerinde eğimlerle takip etmesi sırasında gerçek sistem davranışı Şekil IV.53, Şekil IV.54, Şekil IV.55, Şekil IV.56, Şekil IV.57'de gösterilmektedir. Ancak verilen grafiklerden de anlaşılacağı gibi verilen yörüngeyi ZN-PID kontrolörlü sistem yeterince iyi takip edememiştir.



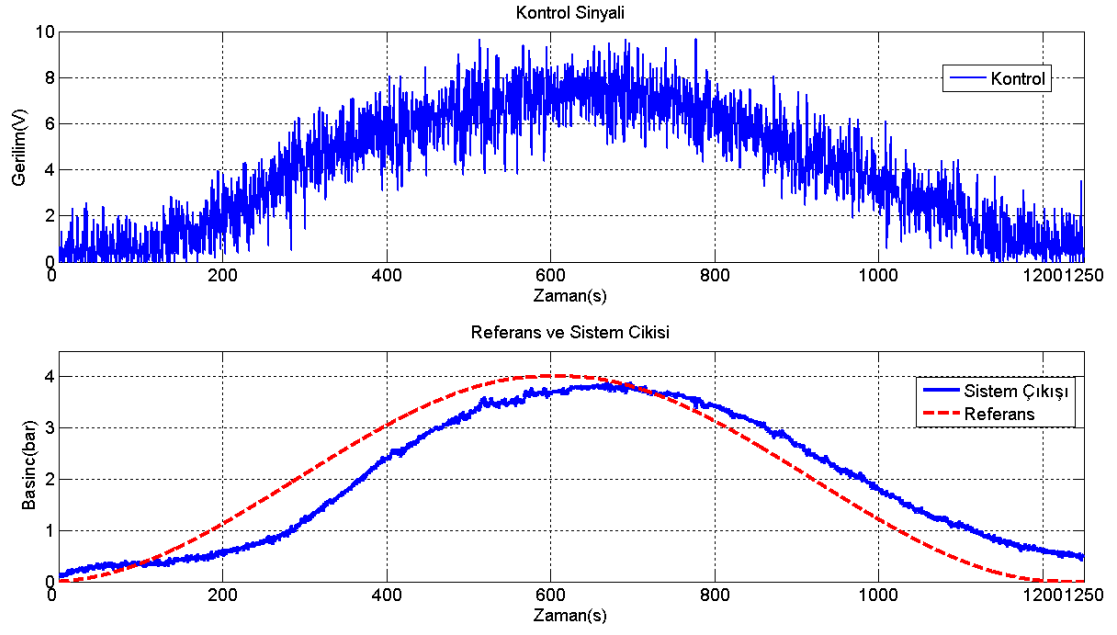
Şekil IV. 53 Eğimi 6 olan referans yörüngenin sistem tarafından takibi



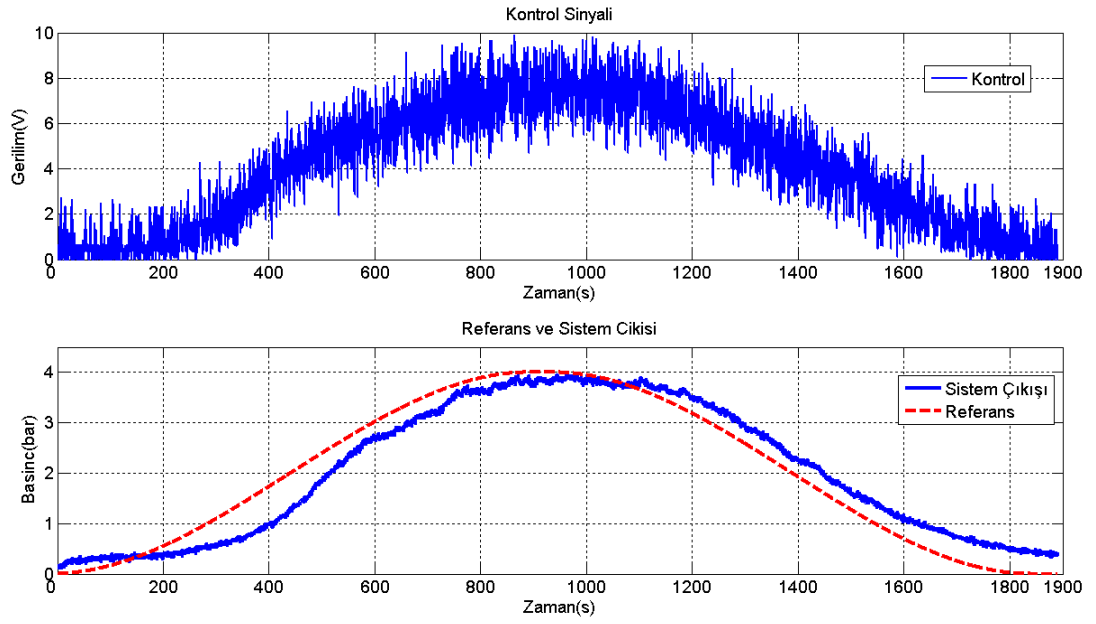
Şekil IV. 54 Eğimi 10 olan referans yörüngenin sistem tarafından takibi



Şekil IV. 55 Eğimi 15 olan referans yörüngenin sistem tarafından takibi

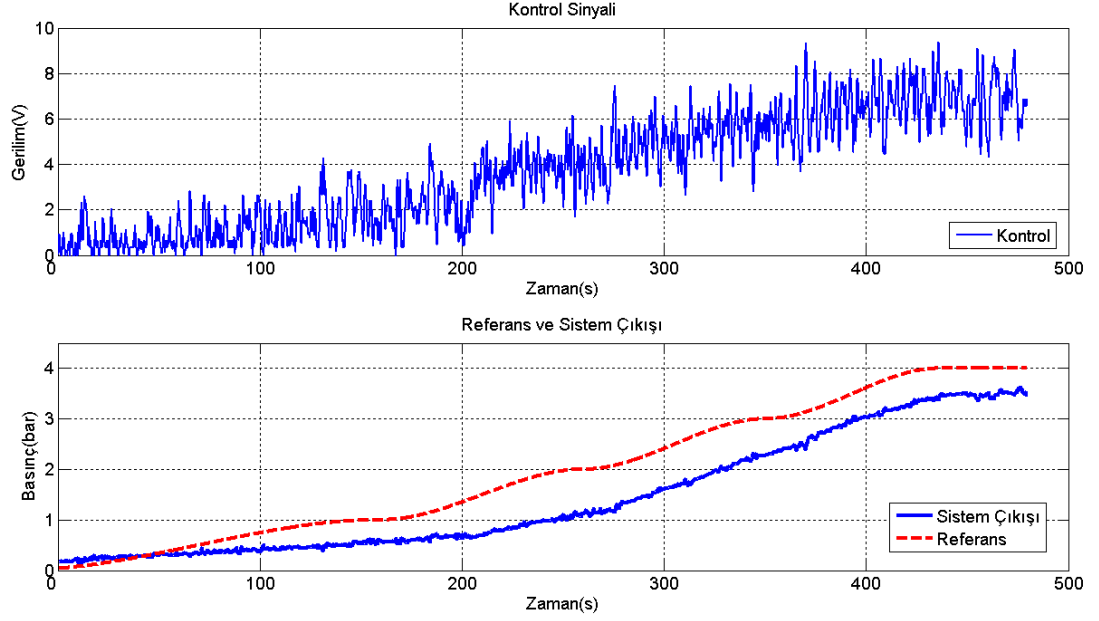


Şekil IV. 56 Eğimi 20 olan referans yörüngenin sistem tarafından takibi



Şekil IV. 57 Eğimi 30 olan referans yörüngenin sistem tarafından takibi

ZN-PID kontrolörlü sistemin farklı eğimli yörünge değerleri arasında geçişi sırasında başarısını görmek için Şekil IV.58’de gösterildiği gibi referans yörünge verilmiştir. Şekil IV.58’den anlaşılabacağı gibi sistem yörüngeler arasında geçişi istenilen şekilde gerçekleştirememiştir.



Şekil IV. 58 Yörüngeler arası geçişe sistemin cevabı

BÖLÜM – V

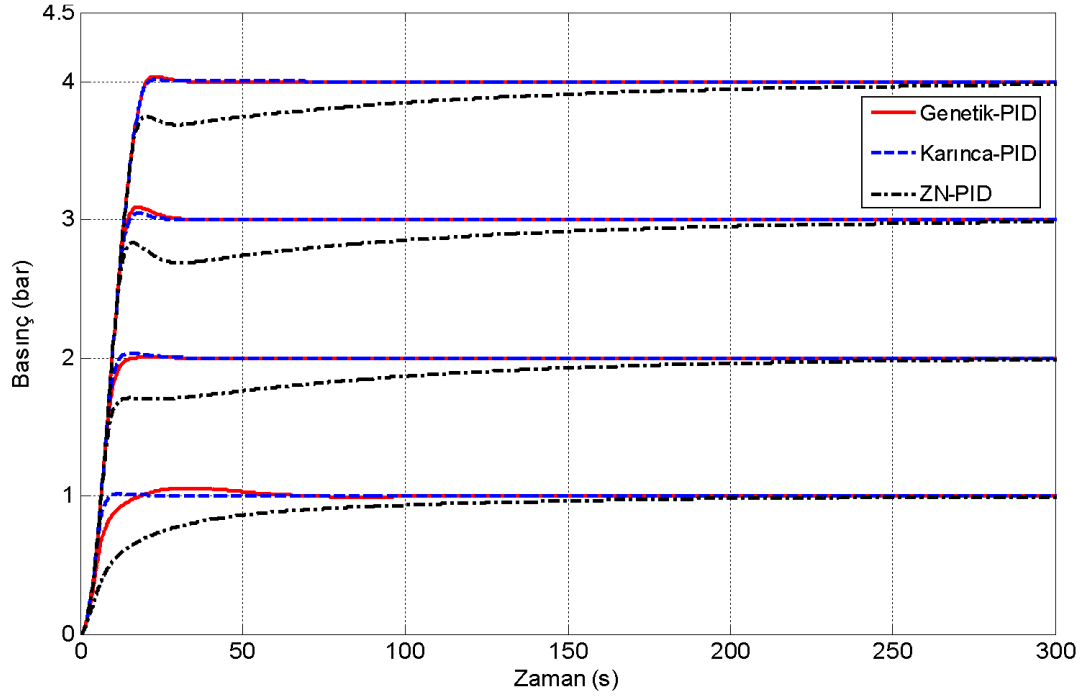
SONUÇLAR

Gunt RT 532 Basınç Proses Kontrol Sisteminin, verilen YSA modeli üzerinden optimize edilmesi ile elde edilen Genetik-PID, Karınca-PID ve ZN-PID kontrolör katsayıları Tablo.1’de gösterilmektedir.

Tablo V. 1 Elde edilen PID katsayıları

	K_p	K_i	K_d
Genetik-PID	8.5829	0.1871	1.7277
Karınca-PID	14.55	0.1875	14.4
ZN-PID	8.16	0.1	2.5

Tablo V.1’de gösterilen PID katsayıları ile sistemin YSA modeline farklı değerlerde basamak girişler verildiğinde, Genetik-PID, Karınca-PID ve ZN-PID kontrolörlü sistemin cevapları Şekil V.1’de gösterilmektedir.



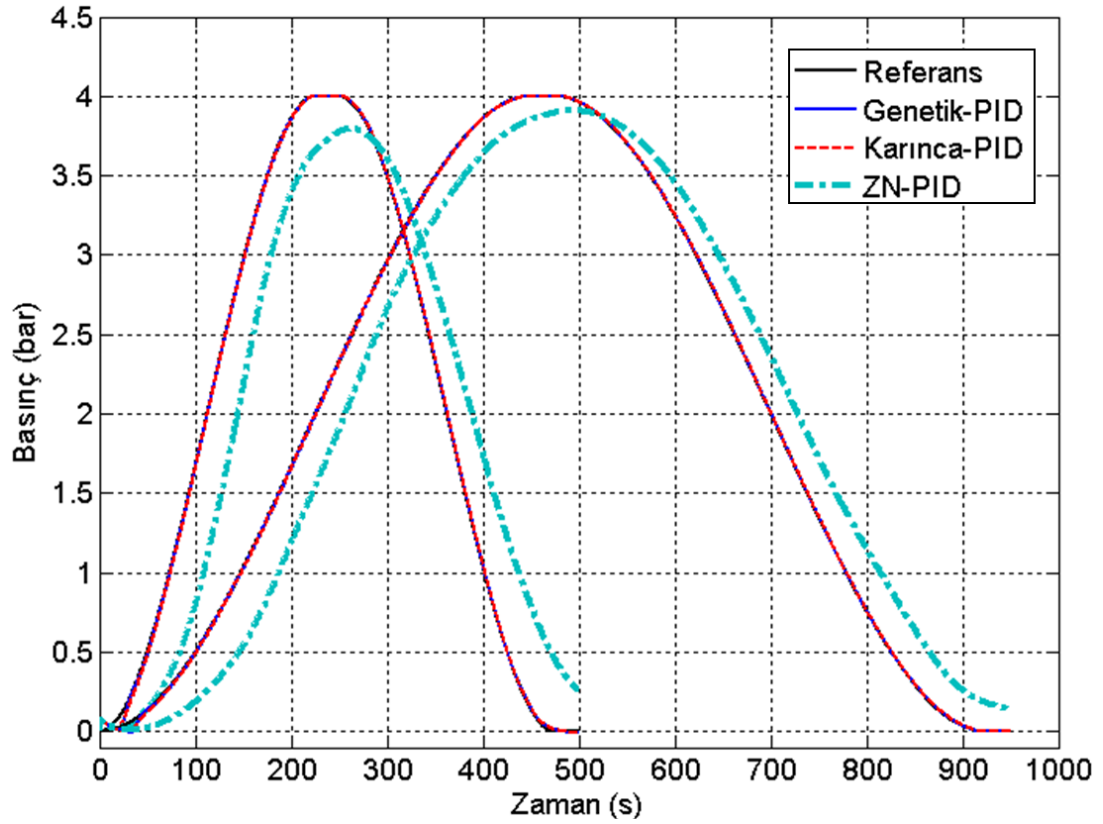
Şekil V.1 Farklı değerlerde basamak girişlere sistemin cevabı

Gunt RT 532 Basınç Proses Kontrol sisteminin YSA modeli üzerinden GA, KKA ve ZN ile optimize edilen PID kontrolörün performansları aşım miktarına (O_s), yükselme zamanına (T_r) ve oturma zamanına (T_s) göre Tablo V.2’de gösterilmektedir. Tablo V.2’ye göre Karınca-PID ve Genetik-PID kontrolörün ZN-PID kontrolöre nazaran daha iyi sonuç verdiği görülmektedir.

Tablo V. 2 PID kontrolörlerin kontrol karakteristikleri

	Referans 1bar			Referans 2bar			Referans 3bar			Referans 4bar		
	GA	KKA	ZN	GA	KKA	ZN	GA	KKA	ZN	GA	KKA	ZN
O_s (%)	5.7023	1.5162	-	0.1655	1.7152	-	3.0343	1.4994	-	0.9266	0.3261	-
T_r	9.338	5.0449	67.417	7.3431	6.8488	63.143	8.9744	9.0067	9.6084	12.475	12.475	12.682
T_s	60.973	16.438	253.54	12.781	11.143	200.65	18.236	13.734	144.27	18.066	18.067	95.128

Sisteme referans olarak verilen 15 ve 30 derece eğimli yörüngeler, sırasıyla Genetik-PID, Karınca-PID ve ZN-PID kontrolörler ile takip edilmiş ve sistem cevapları Şekil V.2’ de gösterilmiştir.



Şekil V.2 Farklı yörünge değerleri için sistem cevapları

Şekil V.2’ de gösterilen, üç farklı kontrolörle elde edilen sistem çıkışı - referans yörüngeleri arasındaki RMS hatası Tablo V.3’ de verilmiştir. Üç farklı kontrolörün RMS hataları karşılaştırıldığında Genetik-PID ve Karınca-PID kontrolörleri ZN-PID kontrolöründen daha başarılı sonuçlar verdiği görülmektedir.

Tablo V.3 Yörünge değerleri için RMS hataları

Eğim	RMS Hataları		
	Genetik-PID	Karınca-PID	ZN-PID
15	0.0109	0.0143	0.4824
30	0.0056	0.0073	0.2755

V.1. TARTIŞMA VE DEĞERLENDİRME

Yapılan çalışmada basınç kontrol ünitesi üzerinde SC tekniği ile Genetik-PID ve Karınca-PID kontrolör, ZN metodu ile ZN-PID kontrolör tasarlanmıştır. Sistemin kontrolü için öncelikle YSA ile sistemin dinamik modeli çıkartılmıştır. Elde edilen bu model ile çevrimdışı çalışma sırasında, sisteme farklı referans değerleri giriş olarak verilmiş ve sistemin çıkışını, girişi ile arasında en az hata ile takip etmesini sağlayan Genetik-PID ve Karınca-PID kontrolörler tasarlanmıştır. Elde edilen kontrolörler ile basınç kontrol ünitesi gerçek zamanda kontrol edilerek, kontrolörlerin performansları incelenmiştir.

Sistemin gerçek zamanlı çalışması değerlendirildiğinde tasarlanan Genetik-PID ve Karınca-PID kontrolörlerin verilen referansları başarılı bir şekilde takip ettikleri, ZN-PID kontrolörün ise diğer kontrolörlere nazaran kötü sonuç verdiği görülmektedir. Böylece gerçek bir sistem için kontrolör tasarlanırken sistemin modelinin elde edilmesi sayesinde kısa sürede optimizasyon teknikleri ile çok fazla sayıda deney yapılarak ve deneyler sırasında sistemi yormadan gerekli kontrol katsayılarının elde edilebileceği gösterilmiştir.

GA ile PID kontrolörün katsayıları bulunurken, verilen arama uzayında genetik operatörler ile elde edilen bir önceki değerlerden yeni değerler elde edilebilir. KKA ile optimizasyon yapılırken, verilen arama uzayında KKA' nın en kısa yolu arama yöntemi ile kombinasyon yaparak çözüm üretir ve verilen sistem için başarılı PID katsayıları bulunabilir. Ancak ZN ile optimizasyon sırasında diğer yöntemlere nazaran yeni PID katsayıları elde edilememekte ve en iyi çözüme doğru bir yöneliş olmamaktadır. Bu özellikte ZN' nin SC' ye nazaran eksikliğini gösterir.

Tez sürecinde yaşanan bazı zorluklar vardır. Bunlar;

Sistemin dinamik modelini çıkarmak için kullanılan YSA yönteminde farklı katman ve nöron sayılı YSA yapıları eğitilmiş ve sonuçları değerlendirilmiştir. Ancak katman ve nöron sayısının artması ile elde edilen YSA modelinin eğitimi yaklaşık 10-12 saate çıkmıştır. Bu da farklı gizli katman ve nöron sayılı modellerin deneme süresini uzatmıştır.

GA ve KKA ile PID kontrolörün optimize edilirken, uygunluk ve maliyet fonksiyonlarının, sistem için çok iyi tasarlanması gerekmektedir. Aksi takdirde sistem için farklı PID katsayıları üretilebilmekte ve istenilen sonuca gidilememektedir. Aynı zamanda GA ve KKA ile optimizasyon yapılırken yaklaşık

500-1000 deneme civarında yapılmakta ve bu denemelerin sonucunda istenilen optimum PID katsayıları elde edilmektedir. Bu süreçte vakit almaktadır.

Bu tezde elde edilen sonuçlardan sonrada çözülmesi gereken problemler vardır. Bunlardan bazıları:

Sistem doğrusal bir davranış göstermediği için klasik matematiksel yöntemler ile transfer fonksiyonu bulmak zordur. Çünkü sistemin çalışması sırasında oluşan gürültüler modelin çıkarılmasını zorlaştırmaktadır. Bu yüzden doğrusal olmayan yöntemlere yani YSA gibi yöntemlerin kullanımını gerektirmektedir.

GA ve KKA ile optimizasyon yaparken, sistem için gerekli olan kontrol parametrelerinin hangi değer aralıkları arasında bilinmesi gerekmektedir. Aksi takdirde kontrol için sağlıklı parametreler bulunamayabilmir. Parametre aralığının bilinmesi içinde sistem üzerinde çeşitli deneyler yapılması gerekir.

Yapılan bu çalışmaya ilave olarak yapılacak bazı ekler, ileriki çalışmalarda daha verimli sonuç alınmasını sağlayabilir. Bunlar;

- Farklı GA kodlama yapıları (gerçel kodlama, gray kodlama.gibi..) ve parametreleri (çift halkalı kromozom yapısı, gibi) geliştirilerek GA' nın performansı araştırılabilir.
- GA' nın yapısında doğal bir paralellik vardır. Bunun getirdiği avantajlardan yararlanarak paralel işlem yapan bilgisayarda uygun paralel GA'lar gerçekleştirilebilir.
- Sistemin modeli çıkartılırken elman ağları gibi ağlar kullanılabilir. Çünkü elman ağı, gizli katman çıkışından gizli katmanın girişine geri besleme alınan iki katmanlı bir ağıdır. Bu yenilenebilir yapı zamanla değişen yani dinamik modelin oluşmasına izin vermektedir. Böylece sistem için performansı daha yüksek bir model elde edilmiş olunabilir.
- Sistemin modeli çıkartılırken, YSA' nın öğrenme özelliğinin yanında bulanık mantığın karar verme özelliğinin birlikte kullanıldığı uyarlanabilir(adaptif) sinirsel-bulanık sistem kullanılabilir. Bu teknik ile sistem için uygun bir model elde edilebilir.
- YSA' nın performansını geliştirmek için gizli katman sayısı, gizli katmandaki nöron sayısı, öğrenme oranı, momentum sabiti ve eğitim algoritmasının seçimi için birçok deney yapılması gerekir. Bu parametrelerin seçiminde GA

veya KKA kullanılması kullanıcıya zaman kazandırabilir ve daha hassas değerler bulunarak sistem için daha iyi bir model elde edilebilir.

REFERANSLAR

- [1] Mok, H. S.; Kim, G. T.; Park, M. H.; Rhew, H. W.: "PI Controller Gains Tuning Of The Pressure Control System By Open-Loop Frequency Response," *IEEE*, (1988).
- [2] Namba, R.; Yamamoto, T.; Kaneda, M.: "Robust PID Controller and Its Application," *IEEE*, (1997).
- [3] Chengyi, G.; Qing, S.; Wenjian, C.: "Real-Time Control Of AHU Based On A Neural Network Assisted Cascade Control System," *Conference on Robotics, Automation and Mechatronics*, Singapore, (2004).
- [4] Jian, W.; Wenjian, C.: "Development Of An Adaptive Neuro-Fuzzy Method For Supply Air Ressure Control In Hvac System," *IEEE*, (2000).
- [5] Ünal, M.; Erdal, H.; Topuz, V.; Ergüzel, T. T.: "Isac Robot Kolunun Yörünge Takibi İçin PID Kontrolörün Genetik Algoritma ile Optimizasyonu," *UMES, İzmit, Kocaeli*, (2007).
- [6] Salami, M.; Cain, G.: "An Adaptive PID Controller Based on Genetic Algorithm Processor," *Genetic Algorithms in Engineering Systems: Innovations and Applications*, (1995).
- [7] Jones, A. H.; DeMouraOliveira, P. B.: "Genetic Auto-Tuning of PID Controllers," *Genetic Algorithms in Engineering Systems: Innovations and Applications*, (1995).
- [8] Gündoğdu, Ö.: "Optimal-Tuning of PID Controller Gains Using Genetic Algorithms," *Pamukkale Üniversitesi Mühendislik Fakültesi Mühendislik Bilimleri Dergisi*, 1 (2005).
- [9] Dandıl, B.; Gökbulut, M.; Ata, F.: "Genetik Algoritma ile Model Referans-PID Denetleyici Tasarımı," *Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 14 (2002) 33-43.
- [10] Alli, H.; Kaya, M.: "Genetik Algoritma Kullanarak PID Kontrol Parametrelerinin Bulunması," *Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 13 (2001) 1-8.
- [11] Mitsukura, Y.; Yamamoto, T.; Kaneda, M.: "A Design of Self-Tuning PID Controllers Using a Genetic Algorithm," *American Control Conference*, (1999) 1361-1365.
- [12] Duan, H.-B.; Wang, D.-B.; Yu, X.-F.: "Grid-Based ACO Algorithm For Parameters Tuning Of NLPID Controller and Its Application in Flight Simulator," *International Journal of Computational Methods*, 3 (2006) 163-175.
- [13] Hsiao, Y.-T.; Chuang, C.-L.; Chien, C.-C.: "Ant Colony Optimization For Designing of PID Controllers," *Intemational Symposium on Computer Aided Control Systems Design*, Taiwan, (2004).
- [14] Ergüzel, T. T.; Akbay, E.: "ACO (Ant Colony Optimization) Algoritması ile Yörünge Takibi," *UMES, İzmit, Kocaeli*, (2007).
- [15] Varol, A.; Bingül, Z.: "A New PID Tuning Technique Using Ant Algorithm," *American Control Conference*, (2004).

- [16] Ogata, K.: *Modern Control Engineering, 3rd Edition*, New Jersey, Prentice Hall (1997).
- [17] Özçalık, H. R.; Küçüktüfekçi, A.: "Dinamik Sistemlerin Yapay Sinir Ağları ile Düz ve Ters Modellenmesi," *KSÜ Fen ve Mühendislik Dergisi*, 6 (2003) 26-35.
- [18] Åström, K. J.; Hägglund, T.: *PID Controllers Theory, Design and Tuning*, USA, Instrument Society of America (1995).
- [19] Tonbul, T. S.; Sarıtaş, M.: "Beş Eksenli Bir Edubot Robot Kolunda Ters Kinematik Hesaplamalar Ve Yörünge Planlaması," *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 18 (2003).
- [20] Coban, R.; Can, B.: "Identification and Control of ITU Triga Mark-II Nuclear Research Reactor Using Neural Networks and Fuzzy Logic " *Advances in Artificial Intelligence*, 3809/2005 Springer Berlin / Heidelberg, (2005) 1057-1062.
- [21] Şafak, C.: "Pnömatik Motorun Hız Kontrolü İçin Genetik-Bulanık Kontrolör Tasarımı," Yüksek Lisans Tezi, *Fen Bilimleri Enstitüsü*. İstanbul Marmara Üniversitesi (2005).
- [22] Can, B.: "The optimal control of ITU Triga Mark II Reactor," *The Twelfth European Triga User's Conference*, NRI Bucuresti-Pitesti, Romania, (1992).
- [23] Topuz, V.: "Bulanık Genetik Proses Kontrolü," Doktora Tezi, *Fen Bilimleri Enstitüsü*. İstanbul Marmara Üniversitesi (2002).
- [24] http://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1. "Yapay Sinir Ağları," (25.05.2007).
- [25] Hagan, M. T.; Demuth, H. B.; Beale, M.: *Neural Network Design*, Boston, PWS (1997).
- [26] Saraç, T.: "Yapay Sinir Ağları," Gazi Üniversitesi Endüstri Mühendisliği Bölümü, Ankara (2004).
- [27] Elmas, Ç.: *Yapay Sinir Ağları (Kuram, Mimari, Uygulama)* Seçkin Yayınları (2003).
- [28] Matlab. "Neural Network." (2006).
- [29] Psaltis, D.; Sideris, A.; Yamamura, A. A.: "A Multilayer Neural Network Controller," *IEEE*, 8 (1985) 17.
- [30] Narendra, K. S.; Parthasarathy, K.: "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, 1 (1990).
- [31] Şafak, K. K.: "Identification of A Universal DC Motor Dynamics Using Feedforward Neural Networks," *Fen Bilimleri Enstitüsü*. İstanbul Boğaziçi Üniversitesi (1997).
- [32] Dorigo, M.: "Ant Algorithms for Discrete Optimization," *Artificial Life*, 5 (1999) 137.
- [33] Engin, O.: "Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu," (2001).
- [34] Karr, C. L.; Freeman, L. M.: *Industrial Applications of Genetic Algorithms* CRC Press (1999).
- [35] Back, T.; Homeister, F.: "Extended Selection Mechanism in Genetic Algorithms," *The Fourth International Conference on Genetics Algorithms*, USA, (1991).
- [36] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs* (1996).

- [37] Goldberg, D. E.; Sastry, K.: "A Practical Schema Theorem For Genetic Algorithms Design and Tuning," University of Illinois, USA, 2001017, (2001).
- [38] Biroğul, S.: "Genetik Algoritma Yaklaşımıyla Atölye Çizelgeleme," *Fen Bilimleri Enstitüsü*. Ankara Gazi Üniversitesi (2005).
- [39] Coley, D. A.: *An Introduction to Genetic Algorithms For Scientist and Engineers* (1999).
- [40] Bonabeau, E.; Dorigo, M.; Theraulaz, G.: "Inspiration For Optimization From Social Insect Behaviour," *Nature*, 406 (2000).
- [41] Gambardella, L. M.; Dorigo, M.: "Ant-Q: A Reinforcement Learning approach to the traveling salesman problem," *ML-95, Twelfth Intern. Conf. on Machine Learning*, (1995) 252-260.
- [42] Coloni, A.; Dorigo, M.; Maniezzo, V.: "Distributed Optimization by Ant Colonies," *The First European Conference on Artificial Life*, Paris, France, (1992).
- [43] Alaykiran, K.; Engin, O.: "Karıncalar Kolonileri Metasezgiseli ve Gezgin Satıcı Problemleri Üzerinde Bir Uygulaması," *Gazi Üniv. Müh. Mim. Fak. Der.*, 20 (2005) 69-76.
- [44] Sanchez, E.; Shibata, T.; Zadeh, L. A.: *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives* (1997).
- [45] Yeniay, Ö.: "An Overview of Genetic Algorithms," *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*, 2 (2001) 37-49.
- [46] Altıparmak, F.; Dengiz, B.; Smith, A. E.: "An Evolutionary Approach For Reliability Optimization in Fixed Topology Computer Networks," *Transactions On Operational Research*, 12 (2000) 57-75.

ÖZGEÇMİŞ

1983 yılında Denizli’ de doğdu. İlkokul’u Ankara’ da Abidin Paşa İlköğretim Okulunda, ortaokul ve liseyi ise Denizli Türk Eğitim Vakfı Anadolu Lisesinde okudu. 2005 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü, Bilgisayar ve Kontrol Öğretmenliği bölümünden mezun oldu. 2005 yılında aynı bölümde Araştırma Görevlisi olarak göreve başladı ve halen devam etmektedir.