

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SIVI SEVİYE DENETİMİ İÇİN BULANIK
DENETLEYİCİ PARAMETRELERİNİN KARINCA
KOLONİ VE GENETİK ALGORİTMA İLE
OPTİMİZASYONU

Türker Tekin ERGÜZEL

DOKTORA TEZİ
ELEKTRONİK - BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR VE KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Erbil AKBAY

2. DANIŞMAN
Prof. Dr. Burhanettin CAN

İSTANBUL, 2009

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SIVI SEVİYE DENETİMİ İÇİN BULANIK
DENETLEYİCİ PARAMETRELERİNİN KARINCA
KOLONİ VE GENETİK ALGORİTMA İLE
OPTİMİZASYONU

Türker Tekin ERGÜZEL
(141200420020342)

DOKTORA TEZİ
ELEKTRONİK - BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR VE KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Erbil AKBAY

2. DANIŞMAN
Prof. Dr. Burhanettin CAN

İSTANBUL, 2009

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 çalışmada öncelikle Gunt RT 512 sıvı seviye kontrol sisteminin Yapay Sinir Ağı (YSA) modeli oluşturulmuştur. Pnömatik oransal valf ve su tankından oluşan bu sistemin sıvı seviyesinin kontrolü Bulanık denetleyici ile gerçekleştirilmiştir. Bulanık denetleyicinin aksiyon değerleri ve üyelik fonksiyonu sınır değerleri esnek programlama yöntemi olarak bilinen genetik algoritma ve karınca kolonisi optimizasyon algoritması ile optimize edilmiştir. Yapılan bu çalışmanın modelleme, optimizasyon ve kontrol üzerine çalışma yapacak kişiler için bir kaynak niteliğinde faydalı olacağına inanıyorum.

Tez çalışmamın her aşamasında değerli vaktini ayırarak yardımcı olan danışmanlarım Yrd. Doç. Dr. Erbil AKBAY ve Prof. Dr. Burhanettin CAN'a, Doç.Dr. A.Fevzi BABA'ya, Doç. Dr. Arif ADLI'ya, desteğini esirgemeyen Yrd. Doç. Dr. Vedat TOPUZ'a, tez süresince birlikte çalıştığım arkadaşım, araştırma görevlisi Muhammed ÜNAL'a, ufku mu varlığıyla açık tutan babam Prof. Dr. M. Mehdi ERGÜZEL'e, manevi desteğiyle beni hep ayakta tutan annem Nurten ERGÜZEL'e, umutsuz günlerimde heyecanımı tazeleyen müstakbel eşim Merve Sarımuratoğlu'na, 9 yıllık öğretmenliğim süresince akademik ve bireysel gelişimime destek olan, samimiyetini ve sıcaklığını hergün hissettiğim okulum Yüzyıl Işıl Okulları'nın değerli kurucusu Lale ÜNALDI'ya teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
ÖZET	vi
ABSTRACT	vii
YENİLİK BEYANI.....	viii
KISALTMALAR.....	viii
ŞEKİLLER	xv
TABLolar	xviii
BÖLÜM I	1
GİRİŞ ve AMAÇ	1
I.1. GİRİŞ	1
I.2. LİTERATÜR TARAMASI	3
I.2.1 Yapay Sinir Ağları İle Sistem Modelleme	3
I.2.2 Genetik Algoritma ile Bulanık Denetleyici Parametre Optimizasyonu	5
I.2.3 KKO Algoritması ile Bulanık Denetleyici Parametre Optimizasyonu..	7
I.2.4 Su Seviye Sistemleri Üzerine Yapılan Çalışmalar	9
BÖLÜM II.....	11
Gunt RT 512 SIVI SEVİYE KONTROL SİSTEMİ.....	11
II.1. GİRİŞ.....	11
II.2. SİSTEMİN GENEL YAPISI.....	12
II.3 SIVI SEVİYE SİSTEMİ ELEMANLARI.....	14
II.3.1. Denetleyici	14
II.3.2. Pnömatik Oransal Valf (I/P).....	14
II.3.3. Basınç Regülatör Ünitesi.....	15
II.3.4. P / I Dönüştürücüsü	16

II.3.5. I / P Dönüştürücüsü (Pnematik Konumlandırıcı).....	16
II.4 SİSTEMİN DAVRANIŞI ve TASARIMI.....	17
II.4.1. Sistemin Çalışma Prensipleri.....	18
II.4.2. Sistemin Dolum / Boşalım Eğrileri	19
II.5. YÖRÜNGE TAKİBİ	21
II.5.1. Sistemde Kullanılan Yörünge Denklemi.....	22
BÖLÜM III	24
ESNEK PROGRAMLAMA.....	24
III.1. GİRİŞ	24
III.1. YAPAY SİNİR AĞLARI (YSA) ile SİSTEM TANIMLAMA.....	25
III.1.1. Biyolojik Sinir Ağları	26
III.1.2. Yapay Sinir Ağı Hücresi (Perceptron).....	27
III.1.3. Yapay Sinir Ağları Modelleri	30
III.1.4. Hatanın Geri Yayılımı Öğrenme Algoritması.....	33
III.1.5. YSA ile Sistem Modelleme	35
III.2. GENETİK ALGORİTMA.....	38
III.2.1. Basit Genetik Algoritma	38
III.2.2. Genetik Algoritma Operatörleri	40
III.2.3. Kısıtlı Optimizasyon Problemlerinin Çözümü.....	46
III.2.4. GA ile Optimizasyon	47
III.3. KARINCA KOLONİ OPTİMİZASYONU ALGORİTMASI (KKO)..	48
III.3.1. Gerçek Karıncalar	49
III.3.2. Yapay Karıncalar	50
III.3.3. Karınca Kolonisi Algoritması	51
III.3. BULANIK MANTIK DENETLEYİCİ.....	53
III.3.1 Bulanık Denetleyici Tasarımda Karşılaşılan Problemler.....	54
III.3.2 Bulanık Kümeler	55

III.3.3 Bulanık Denetleyicinin Temel Yapısı.....	56
BÖLÜM IV	63
Gunt RT 512 SİSTEMİNİN YAPAY SİNİR AĞLARI ile	
MODELLENMESİ	63
IV.1. GİRİŞ	63
IV.2. SİSTEMİN YSA ile MODELLENMESİ.....	63
IV.2.1. Kullanılan YSA Modeli.....	64
IV.2.2. Eğitim Verisinin Toplanması.....	64
IV.2.3. YSA Mimarisinin Seçilmesi	66
IV.2.4. YSA'nın Eğitimi	71
BÖLÜM V	75
SIVI SEVİYE DENETİMİ İÇİN BULANIK DENETLEYİCİ	
PARAMETRELERİNİN KARINCA KOLONİ VE GENETİK	
ALGORİTMA İLE OPTİMİZASYONU.....	75
V.1. GİRİŞ.....	75
V.2. BULANIK DENETLEYİCİ TASARIMI	75
V.2.1. Üyelik Fonksiyonları ve Kurallar	76
V.3. GENETİK ALGORİTMA ile BULANIKDENETLEYİCİ TASARIMI	81
V.3.1. Bulanık Denetleyici Parametrelerin Genetik Olarak Kodlanması	82
V.3.2. Bulanık Denetleyici Parametrelerin GA Tarafından Öğrenilmesi....	83
V.3.3. Sistem Çıkışının GA Uygunluk Değerine Dönüştürülmesi	83
V.3.4 Kullanılan Genetik Operatörler.....	85
V.4. KARINCA KOLONİ OPTİMİZASYONU ALGORİTMASI İLE	
BULANIK DENETLEYİCİ TASARIMI	89
V.4.1. KKO Algoritması Parametrelerinin Bulunması.....	93
V.5. RASTGELE GİRİŞ İŞARETLERİ İLE PARAMETRE	
OPTİMİZASYONU	97
BÖLÜM VI	100

GUNT RT 512 SİSTEMİ İLE YAPILAN DENEYLER.....	100
VI.1. GİRİŞ	100
VI.2. ÖRNEKLEME ZAMANI DENEYLERİ.....	100
VI.3. KKO - BULANIK DENETLEYİCİ DENEYLERİ	101
VI.3.1. Sistemin Basamak Giriş Cevapları	101
VI.3.2. Basamak Girişler Arası Geçişlerde Sistem Davranışı	104
VI.3.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı	106
V.1.4. KKO – Bulanık Denetleyicinin Yörünge Takip Cevapları.....	112
VI.2. GENETİK - BULANIK DENETLEYİCİ DENEYLERİ	116
VI.2.1. Basamak Girişlere Sistemin Davranışı	116
VI.2.2. Basamak Girişler Arasında Geçişte Sistemin Davranışı.....	118
VI.2.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı.....	120
VI.2.4. Genetik – Bulanık Denetleyicinin Yörünge Takip Cevapları.....	126
BÖLÜM VII.....	129
KKO - BULANIK, GENETİK – BULANIK DENETLEYİCİ DENEY	
SONUÇLARININ KARŞILAŞTIRILMASI	129
VII.1. GİRİŞ	129
VII.1. Farklı Referans Seviyeleri için KKO ve Genetik Algoritma	
Davranışları.....	130
VII.2. Yörünge Takibinde KKO ve Genetik Algoritma Davranışları	131
BÖLÜM VIII	133
TARTIŞMA ve DEĞERLENDİRME	133
VIII.1. GİRİŞ.....	133
REFERANSLAR.....	136
ÖZGEÇMİŞ.....	143

ÖZET

SIVI SEVİYE DNETİMİ İÇİN BULANIK DENETLEYİCİ PARAMETRELERİNİN KARINCA KOLONİ VE GENETİK ALGORİTMA İLE OPTİMİZASYONU

Endüstride kullanılan kontrol sistemlerinin parametre optimizasyonunda Genetik Algoritma (GA) ile PID veya bulanık denetleyici yaygın olarak birlikte kullanılmaktadır. Bu çalışmada ise, kontrol alanında henüz yeni olan Karınca Koloni Optimizasyonu (KKO) algoritması ile bulanık denetleyici birlikte kullanılmıştır.

Bu tez, Marmara Üniversitesi Elektronik-Bilgisayar Eğitimi Bölümü Sayısal Kontrol Sistemleri Laboratuvarında bulunan Gunt RT 512 sıvı seviye sisteminin kontrolü üzerinedir. Bulanık denetleyicinin aksiyon ve sınır değerleri GA ve KKO ile optimize edilerek ayrı ayrı bulunmuş ve her iki algoritmanın performansı simülasyon ve gerçek zamanlı deneyler ile karşılaştırılmıştır.

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 Bulanık denetleyici çıkış değişkeni aksiyon değerleri ve üyelik fonksiyonu sınır değerleri KKO algoritması ve GA ile optimize edilmiştir. KKO algoritması ve GA teknikleri ile elde edilen sonuçlar, yükselme zamanı, oturma zamanı kriterlerine ve ortalama karesel hataya (Root Mean Square - RMS) göre karşılaştırılmış, her iki optimizasyon algoritmasının performanslarının birbirlerine çok yakın olduğu gözlenmiştir.

Mart, 2009

Türker Tekin ERGÜZEL

ABSTRACT

PARAMETER OPTIMIZATION OF FUZZY CONTROLLER FOR WATER LEVEL CONTROLLER USING GENETIC AND ANT COLONY OPTIMIZATION ALGORITHMS

For the systems used in industry, genetic algorithm is usually used with either PID or fuzzy controllers together. But in this study, ant colony optimization algorithm, which is new for control systems, is used with fuzzy controllers.

This thesis describes a real time control algorithm, using genetic algorithm (GA) and ant colony optimization (ACO) algorithm for optimizing Fuzzy controller parameters which are action values and membership function boundary values developed for Gunt RT 512 Water Level Control Trainer System in the Digital Control Systems Laboratory of Technical Education Faculty at Marmara University. Optimized parameters of both algorithms are then compared with real time and simulation applications.

The dynamic model of the process to be controlled was obtained using Artificial Neural Network (ANN). To build the model, the system was run with diverse and random input values. Its output values were saved to relate them with the inputs. Using the input-output data set, models with various number of layers and neurons were created for the process. The final model was evaluated based on the performance functions. The model with minimum error was chosen as the dynamic model of the process.

Using the ANN model, the parameters of Fuzzy controller were optimized with ACO and GA algorithms. 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 were quite similar.

March, 2009

Türker Tekin ERGÜZEL

YENİLİK BEYANI

SIVI SEVİYE DENETİMİ İÇİN BULANIK DENETLEYİCİ PARAMETRELERİNİN KARINCA KOLONİ VE GENETİK ALGORİTMA İLE OPTİMİZASYONU

Sistem kontrolünde literatürde bir çok çalışma yapılmıştır. II. ve III. bölümde yer alan açıklamalar ve değerlendirmelerde ilgili çalışmalar referansları ile birlikte verilmiştir. Bu uygulamalarda genellikle sistemin modeli teorik yöntemle elde edilmiş, denetleyici parametreleri de simülasyon üzerinde kullanılmıştır. Değişik çalışmalarda sistem modelleme, denetleyici seçimi ve parametre optimizasyonu sırasında karşılaşılan sorunlar ve bunların çözümü için gerçekleştirilen yenilikler incelenmiş, bilimsel katkıları ortaya konmuştur. Bu çalışmada daha önceki çalışmalardan farklı olarak;

- GuntRT 512 sıvı seviye sisteminin gerçek zamanlı kontrolü için kullanılacak bulanık denetleyici üyelik fonksiyonu sınır değerlerinin ve aksiyon değerlerinin elde edilmesinde ilk defa KKO algoritması kullanılmıştır. Ayrıca sistemin kontrolü gerçek zamanlı (Real Time) olarak gerçekleştirilmiştir.
- GuntRT 512 sıvı seviye sistemi için bulanık denetleyici tasarımı gerçekleştirilmiştir. Bulanık denetleyicinin üyelik fonksiyonu sınır değerleri ile aksiyon değerleri hem KKO algoritması hem de Genetik algoritma ile elde edilerek karşılaştırılmıştır.
- Bulanık denetleyiciye ait üyelik fonksiyonu sınır değerleri ile aksiyon değerleri RMS (Root Mean Square) hatası değerini minimum seviyeye indirecek şekilde optimize edilmiştir. YSA (Yapay Sinir Ağları), bulanık denetleyici, KKO ve Genetik algotimaları, GuntRT 512 sıvı seviye sisteminin modellenmesi ve kontrolünde ilk olarak bu çalışmada kullanılmıştır.
- Literatür taraması neticesinde, simülasyon çalışmalarının gerçek sisteme daha yakın olması için sistemin fiziksel sınırlamalarının dikkate alınması, modellenme

işleminin bu fiziksel sınırların içerisinde yapılması gerektiği görülmüştür. Bu nedenle çalışmada, su pompası motorunun ısınmasından kaynaklanan performans düşüklüğü de göz önüne alınarak model verisi elde edilmiştir.

- Bulanık denetleyici parametrelerinin elde edilmesinde, model simülasyonuna gerçek sistemden alınan rastgele girişler referans olarak verilmiştir. Bu sayede sistem, gerçek zamanlı çalışmada bozucu etkisinden kaynaklanan kalıcı hal hatasını minimum seviyeye indirebilmiştir.
- Yapılan deney seti Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar ve Kontrol Eğitimi bölümünde ders içerikleri göz önünde bulundurulduğunda, müfredat içinde Endüstriyel Proses Kontrol ve Otomatik Kontrol gibi lisans ve yüksek lisans derslerinde kullanılabilecektir.

Yukarıda sayılan sebeplerden dolayı, bu tez çalışmasının kontrol bilimine, gerçek zamanlı uygulamalara, kontrolüne ve bilgisayar eğitimine katkı getireceğine inanıyoruz.

Mart, 2009

Prof. Dr. Burhanettin CAN

Yrd. Doç. Erbil AKBAY

Türker Tekin ERGÜZEL

SEMBOLLER

u_k	: Sayısal kontrol işareti
V_y	: Yörünge değeri
t_s	: Oturma zamanı
t_o	: Gecikme zamanı
V_o	: Başlangıç noktası
V_s	: İstenilen referans noktası
Y	: Sistem çıkışı
ϵ	: Yörünge eğim katsayısı
R	: İstenen yörünge çıkışı
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 katmanın duyarlık değeri
α	: YSA' nın öğrenme oranı
b	: YSA' nın bias(eşik) değeri
D	: YSA' nın zaman gecikmesi
e	: Hata vektörü
t	: Hedef vektörü
p	: Giriş vektörü
k	: Giriş sayısı
$\hat{y}$	: Tahmin edilen YSA çıkışı
N	: Genetik algorithmada (GA) popülasyon büyüklüğü
a	: Yapay sinir ağı hücresi çıkışı
Φ	: GA' da uygunluk değeri
P_c	: GA' da bir bireyin çaprazlama olasılığı
p_i	: GA' da bireyin seçilme olasılığı
p_k	: GA' da bir bireyin yaşama olasılığı
q	: GA' da turnuva büyüklüğü
r_u	: GA' da rulet tekerleği yönteminde bir bireyin seçilme ihtimali
n	: Yapay sinir ağı hücresi girişi
v	: Sigmoid fonksiyon eğim değeri
τ	: Karınca Kolonisi Optimizasyonu Algoritmasında feromen miktarı
λ	: KKO' da buharlaşma katsayısı
c	: KKO' da RMS hatası
Δ	: KKO' da her tur sonunda feromen miktarını tutan matris

KISALTMALAR

BM	: Bulanık Mantık
YSA	: Yapay Sinir Ağı
GA	: Genetik Algoritma
KKO	: Karınca Kolonisi Optimizasyonu
SGA	: Basit Genetik Algoritma
EA	: Evrim algoritmaları
NARX	: Nonlinear Autoregressive Network with Exogenous Inputs
PID	: Oransal Integral Türevsel
SC	: Esnek Programlama
RMS	: Karekök Ortalama (Root Mean Square)
MSE	: Karesel Hata Ortalaması (Mean Square Error)
SSE	: Karesel Hataların Toplamı (Sum Square Error)
ADC	: Analog Sayısal Çevirici
DAC	: Sayısal Analog Çevirici
TG	: Takagi – Gugen
mA	: Miliamper
V	: Gerilim
BR	: Bayesian Regularization eğitim fonksiyonu
LM	: Levenberg-Marquardt eğitim fonksiyonu
NB	: Negatif Büyük
NK	: Negatif Küçük
ZR	: Sıfır
PK	: Pozitif Küçük
PB	: Pozitif Büyük
TDL	: Tapped Delay Line

ŞEKİLLER

SAYFA NO

Şekil II. 1 Gunt RT 512 Sıvı Seviye Sistemi	12
Şekil II. 2 Gunt RT 512 İşlem Diagramı	13
Şekil II. 3 UDC 5000 Denetleyici	14
Şekil II. 4 Pnomatik oransal valf	14
Şekil II. 5 Pnomatik oransal valf, akım / valf açıklığı grafiği	15
Şekil II. 6 Basınç Regülatörü Şekil II. 7 PMC 133 basınç dönüştürücüsü	15
Şekil II. 8 Sıvı seviye tüpü seviye-gerilim grafiği	16
Şekil II. 9 Pnomatik Konumlandırıcı	16
Şekil II. 10 Tasarlanan kontrol sisteminin genel yapısı	17
Şekil II. 11 Kontrol edilen sistemin blok diyagramı	19
Şekil II. 12 Sistemin basamak girişlere verdiği cevaplar	20
Şekil II. 13 Sistemin farklı seviyelerden boşalma eğrileri	21
Şekil II. 14 (A) noktadan noktaya (B) yol takibi (C) yörünge takibi	22
Şekil II. 15 Çalışmada kullanılan yörünge grafiği	23
Şekil III. 1 Esnek programlamayı oluşturan programlama teknikleri	24
Şekil III. 2 Doğal sinir hücresi	26
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	29
Şekil III. 7 Tanjant hiperbolik fonksiyon	30
Şekil III. 8 İleri beslemeli YSA modeli	31
Şekil III. 9 Geri beslemeli YSA yapısı	31
Şekil III. 10 YSA' nın eğiticili öğrenme ile eğitilmesi	32
Şekil III. 11 İleri beslemeli çok katmanlı sinir ağı	34
Şekil III. 12 YSA ile modellemede kullanılan temel ileri tip blok şema	36
Şekil III. 13 Seri – paralel ileri tip modelleme blok şeması	37
Şekil III. 14 Genetik algoritma akış diyagramı	39
Şekil III. 15 İkili kodlama	40
Şekil III. 16 Rulet Tekerleği seçim yöntemi	43
Şekil III. 17 Çaprazlama operatörü	45
Şekil III. 18 Aritmetik çaprazlama	45
Şekil III. 19 Mutasyon operatörü	46

Şekil III. 20 Karıncaların izlediği yol	49
Şekil III. 21 Karıncaların bir engelle karşılaşması.....	49
Şekil III. 22 Engelle karşılaşan karıncaların seçimi	50
Şekil III. 23 Karıncaların kısa yolu bulmaları	50
Şekil III. 24 Bulanık denetleyicinin genel yapısı	56
Şekil III. 25 Bulanık denetleyici işlem adımları	57
Şekil III. 26 Çeşitli üyelik fonksiyonları ve üyelik fonksiyon denklemleri	58
Şekil III. 27 Mamdani Bulanık modeli	60
Şekil III. 28 Sugano Bulanık Modeli	61
Şekil IV. 1 Gerçekleştirilen ileri tip seri – paralel NARX model blok şeması.....	64
Şekil IV. 2 YSA eğitimi için uygulanan girişler.....	65
Şekil IV. 3 YSA eğitimi için elde edilen çıkışla	65
Şekil IV. 4 BR İle Eğitilmiş Tek Giriş.....	68
Şekil IV. 5 BR İle Eğitilmiş Çift Giriş	68
Şekil IV. 6 LM İle Eğitilmiş Tek Giriş	68
Şekil IV. 7 LM İle Eğitilmiş Çift Giriş	68
Şekil IV. 8 Kullanılan YSA modelin blok diyagramı	69
Şekil IV. 9 Gizli katmanındaki nöronların bağlantısı.....	70
Şekil IV. 10 50 iterasyonlu eğitim süresince SSE hatasının değişim eğrisi.....	71
Şekil IV. 11 Hedef veriler ile YSA Model çıkışının regresyon eğrisi.....	72
Şekil IV. 12 Hedef veriler ile YSA Model çıkışının hata eğrisi.....	72
Şekil IV. 13 Sistem ile YSA Modeli çıkış grafiği.....	73
Şekil IV. 14 Açık çevrimde sistem ile YSA çıkışının karşılaştırılması	74
Şekil V.1 Tasarlanan denetleyici bağlantı şeması.....	76
Şekil V.2 Oluşturulan bulanık denetleyici yapısı.....	77
Şekil V.3 Z-şekilli (zmf) üyelik fonk. Şekil V.4 S-şekilli (smf) üyelik fonk.....	78
Şekil V.5 []-şekilli (pimf) üyelik fonk.	78
Şekil V.6 Oluşturulan bulanık denetleyicinin hata girişine ait üyelik fonksiyonları.....	79
Şekil V.7 Bulanık denetleyicinin hatadaki değişim girişi üyelik fonksiyonları	80
Şekil V.8 GA ile Bulanık denetleyici parametre optimizasyonu	82
Şekil V.9 Bulanık Denetleyici için geliştirilen kromozom yapısı	83
Şekil V.10 Tipik bir birim basamak giriş için sistem cevabı.....	84
Şekil V.11 KKO Algoritması ile Bulanık Denetleyici Parametre Optimizasyonu.....	90
Şekil V.12 Bulanık denetleyici parametreleri için karıncanın izleyebileceği yollar ..	91
Şekil V.13 KKO-Bulanık denetleyici tasarımı ile kontrol edilen sistem şeması	93
Şekil V.14 Parametre optimizasyonunda bozucu ile kullanılan referans eğrisi	97
Şekil V.15 Farklı referans girişler ile optimize edilen parametrelerin sistem çıkışı ..	98

Şekil V.16	%70 bozucu ithali sonrası sistem davranışı ve kalıcı hal hatası.....	99
Şekil VI.1	30cm referans için gerçek sistem, simülasyon çıkışı ve kontrol işareti..	102
Şekil VI.2	45cm referans için gerçek sistem, simülasyon çıkışı ve kontrol işareti..	103
Şekil VI.3	60cm referans için gerçek sistem, simülasyon çıkışı ve kontrol işareti..	104
Şekil VI.4	Basamak geçişler sırasında sistem davranışı, kontrol işareti değişimi...	105
Şekil VI.5	Açık çevrimde sistemin bozuculara cevabı.....	106
Şekil VI.6	%40 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	107
Şekil VI.7	%50 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	108
Şekil VI.8	%60 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	109
Şekil VI.9	% 70 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	110
Şekil VI.10	% 80 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	111
Şekil VI.11	% 90 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	112
Şekil VI.12	Yörünge katsayısı 20 olan eğri için sistem cevabı ve kontrol işareti....	113
Şekil VI.13	Yörünge katsayısı 10 olan eğri için sistem cevabı ve kontrol işareti....	114
Şekil VI.14	Yörünge katsayısı 2 olan eğri için sistem cevabı ve kontrol işareti.....	115
Şekil VI.15	30 cm referans için sistem, simülasyon çıkışı ve kontrol işareti.....	116
Şekil VI.16	45 cm referans için sistem, simülasyon çıkışı ve kontrol işareti.....	117
Şekil VI.17	60 cm referans için sistem, simülasyon çıkışı ve kontrol işareti.....	118
Şekil VI.18	Basamak geçişler sırasında sistem davranışı, kontrol işareti değişimi.	119
Şekil VI.19	%25 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	120
Şekil VI.20	%40 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	121
Şekil VI.21	%50 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	122
Şekil VI.22	%60 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	123
Şekil VI.23	%70 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	124
Şekil VI.24	% 80 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	125
Şekil VI.25	% 90 bozucu uygulanan sistemin cevabı ve kontrol işareti.....	126
Şekil VI.26	Yörünge katsayısı 20 olan eğri için sistem cevabı ve kontrol işareti....	127
Şekil VI.27	Yörünge katsayısı 10 olan eğri için sistem cevabı ve kontrol işareti...	128
Şekil VI.28	Yörünge katsayısı 2 olan eğri için sistem cevabı ve kontrol işareti.....	129
Şekil VII.1	30cm referans seviyesi için gerçek sistem ve simülasyon çıkışları.....	130
Şekil VII.2	45cm referans seviyesi için gerçek sistem ve simülasyon çıkışı.....	130
Şekil VII.3	60cm referans seviyesi için gerçek sistem ve simülasyon çıkışları.....	130
Şekil VII.4	Farklı eğimlerdeki yörüngeler için sistem cevapları.....	132

TABLolar

SAYFA NO

Tablo III. 1 Bireylerin uygunluk deęerlerinin bilgisayar ortamına evrilmesi.....	43
Tablo IV. 1 YSA mimarisi iin yapılan deney sonuları.....	66
Tablo IV. 2 Sistem ıkıřından alınan gerilim / ykseklik dnřm tablosu.....	67
Tablo V. 1 Bulanık denetleyici kural tablosu.....	78
Tablo V. 2 Bulanık denetleyicinin <i>hata</i> giriři iin yelik fonksiyonu sınır deęerleri	79
Tablo V. 3 Bulanık denetleyicinin <i>dhata/dt</i> giriři yelik fonksiyonu sınır deęerleri.	80
Tablo V. 4 Farklı deęerler ile alıřtırılan genetik operatrler ve RMS deęerleri.....	87
Tablo V.5 Genetik Algoritma ile Optimize Edilen <i>hata</i> Giriři Sınır Deęerleri	88
Tablo V.6 Genetik Algoritma ile Optimize Edilen <i>dhata/dt</i> Giriři Sınır Deęerleri...	88
Tablo V.7 Genetik Algoritma ile Optimize Edilen Aksiyon Deęerleri.....	88
Tablo V.8 Farklı deęerleri ile alıřtırılan KKO operatrleri ve RMS deęerleri	95
Tablo V.9 KKO Algoritması ile Optimize Edilen <i>hata</i> Giriři Sınır Deęerleri	96
Tablo V.10 KKO Algoritması ile Optimize Edilen <i>dhata/dt</i> Giriři Sınır Deęerleri ..	96
Tablo V.11 KKO Algoritması ile Optimize Edilen Aksiyon Deęerleri.....	96
Tablo VI.1 Farklı rnekleme zamanları iin sistem ıkıřı RMS deęerleri	101
Tablo VII.1 Kalıcı, geici hal cevaplarına iliřkin parametrelerin karřılařtırılması.	131
Tablo VII.2 Farklı eęimlerdeki referans iin sistem ıkıřı RMS deęerleri	132

BÖLÜM I

GİRİŞ ve AMAÇ

1.1. GİRİŞ

Endüstride sıvı seviye sistemleri yaygın olarak kullanılmaktadır. Bu sistemler genellikle PID ile kontrol edilmektedir [1]. Bu yöntemlerle kontrol edilen sistemlerde aşım ve kararlı hal hatası oluşmakta, bu da sistemin hassasiyetini olumsuz yönde etkileyebilmektedir [2].

Sıvı seviye sistemleri çoğunlukla sistemin ataletine bağlı olarak gecikme gösteren ve lineer davranmayan sistemlerdir [3]. Bu sistemlerde, sıvının yoğunluğu, şebeke gerilimi, su pompası motorunun çalışma süresi ve bozucu etkiler gibi etkenler sistemin doğrusal olmayan bir davranış göstermesine sebep olur. Doğrusal olmayan sistem tanımlamaları için farklı teknikler kullanılmıştır. Bunların çoğu Wiener–Hammerstein modeli, Volterra serileri, wavelet ağları ve yapay sinir ağları gibi parametrelili doğrusal olmayan sistemlerdir [4].

Lineer olmayan sistemlerin ve karmaşık sistemlerin modellenmesinde yapay sinir ağlarının oldukça etkin sonuçlar ürettiği bilinmektedir [5]. Yapay sinir ağları iz tanıma, sistem modelleme gibi pek çok alanda da kullanılmıştır [6]. Bu sebeplerden dolayı sıvı seviye sistemin modeli çıkartılırken matematiksel model yerine, YSA modeli kullanılmıştır.

Sistem modellemede son yıllarda yapay sinir ağları ile birlikte Bulanık sistemleri birleştirilerek modelleme yapımı yaygın olarak da görülmektedir [7].

Bulanık denetleyici temel olarak insanların gündelik hayatta kullandığı yargıları “if-then” kuralları ile ifade ederek bu kuralların üyelik fonksiyonlara dahil olduğu oranda elde edilen sonuçlara göre bir cevap üretir [8]. Kullanılan bulanık denetleyici kuralları ve üyelik fonksiyonları genellikle sistemi tanıyan uzman tarafından deneme yanılma yöntemi ve tecrübesi ile ortaya çıkartılmaktadır. Dolayısıyla bu üyelik fonksiyonlarının aralıklarının ve türlerinin optimize edilmesi sistem performansı açısından önemlidir [9]. Geçmişte Bulanık parametrelerinin

optimize edilmesinde farklı algoritmalar kullanılmıştır. KKO'nun bu algoritmalara göre öne çıkan özelliği yerel minimumlarda takılmaması ve üyeler arası paralel çalışabilme becerisidir [10].

Bu çalışmada, GuntRT 512 sıvı seviye kontrol sisteminde kullanılan UDC5000 denetleyicinin yerine kullanılacak olan bulanık denetleyicinin üyelik fonksiyonu sınır değerleri ile aksiyon değerleri simülasyon ortamında Genetik ve KKO algoritması ile optimize edilerek bulunması ve her iki algoritmanın performansının karşılaştırılması amaçlanmıştır. Sistemin gerçek zamanlı çalışmasındaki davranışını elde etmek için sisteme farklı değerlerde girişler verilerek çıkış davranışı kaydedilmiş ve bu giriş-çıkış verileri YSA'ya eğitim verisi olarak uygulanıp sistemin dinamik modeli elde edilmiştir. Elde edilen dinamik YSA modeli üzerinden, bulanık denetleyicinin aksiyon değerleri ve üyelik fonksiyonu sınır değerlerinin optimizasyonu sağlanmaktadır.

Araştırmacı M. Dorigo kombinasyonel optimizasyon problemlerin çözümü için bir tür sürü zekası algoritması olan karınca koloni optimizasyonu algoritmasını önermiştir [11]. Karıncaların yuvadan yemek kaynağına giden en kısa yolu bulması mantığını esas alan yaklaşım ile veri madenciliği, gezgin adam problemi, network yönlendirmesi, makine öğrenmesi gibi klasik kombinasyonel optimizasyon problemleri başarı ile çözüme ulaştırılmıştır [12]. Son yıllarda kullanılan genetik, tabu araştırma gibi heuristic ve evrimsel yöntemlerden sonra KKO algoritması oldukça yeni bir yöntem olarak karşımıza çıkmaktadır [13].

En iyinin hayatta kalma ihtimalinin daha yüksek olması felsefesine göre çalışan Genetik algoritma denetleyicinin performansını, performans indeks katsayısı ile izlemektedir [14]. GA, canlıların en iyi olanı 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 ulaşır [15]. 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 [16].

Literatürde, bulanık denetleyici, evrim algoritmaları, yapay sinir ağları (YSA) gibi yapay zeka tekniklerinin birkaçının birlikte kullanılması esnek programlama olarak adlandırılır [17].

Bu çalışmada, esnek programlamayı oluşturan iki yöntem, bulanık denetleyici ile GA ve KKO algoritmaları, birlikte bir proses kontrol işleminde kullanılarak esnek programlama yaklaşımının proses kontrolünde uygulanabilirliği ve etkinliği gösterilmiştir. Elde edilen YSA modeli üzerinden, bulanık denetleyicinin üyelik fonksiyonu sınır değerleri ile aksiyon değerleri GA ve KKO algoritmaları ile bulunmuştur. Yol boyunca bulanık denetleyicinin üyelik fonksiyonu sınır değerleri ile aksiyon değerleri, *bulanık kontrol parametreleri* olarak adlandırılacaktır. Bulanık denetleyiciye ait parametreler RMS hatası değerini minimum seviyeye indirecek şekilde optimize edilmiştir. Bulunan değerler ile sistem gerçek zamanlı kontrol edilmiş davranışını test etmek için sisteme farklı referanslar değerleri verilmiştir. Elde edilen sistem davranışları da gerçek zamanlı çalışma çıktıları ile karşılaştırılmıştır.

1.2. LİTERATÜR TARAMASI

Literatürde, sıvı seviye sistemleri ile ilgili yapılmış kontrol uygulamaları ve kontrol aşamasında kullanılan YSA, GA, Genetik - Bulanık denetleyici, KKO ve KKO - Bulanık denetleyicilerle 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 [18].

1.2.1 Yapay Sinir Ağları İle Sistem Modelleme

Yapay sinir ağları dinamik sistemlerin modellenmesinde ve kontrolünde yaygın olarak kullanılmaktadırlar [19]. Özellikle kuvvetli doğrusal olmayan haritalama ve öğrenme becerileri yapay sinir ağlarının öne çıkan özellikleridir [20].

R. Çoban yaptığı çalışmada ITU Triga Mark-II reaktörünü ve reaktörün ayar kontrol çubuğunun servomotor sistemini yapay sinir ağları ile modellemiş ve reaktör için yörünge izleyen Bulanık denetleyici tasarlamış, denetleyiciye ilişkin üyelik fonksiyonlarını, aksiyon ağırlıkları genetik algoritma ile optimize etmiştir [21].

A. Uçar yaptığı çalışmada doğrusal olmayan sistemlerin modellenmesinde Yapay Sinir Ağları ve ANFIS modelleme tekniklerini kullanmıştır. Uçar farklı doğrusal davranmayan dinamik sistemlerden elde ettiği modelleri PID denetleyici kullanarak her iki modelleme yöntemin performanslarını karşılaştırmış, YSA ile elde edilen modellerin üstünlükleri ortaya konmuştur [22].

Dong-hai Zhai ve arkadaşları, Nero-Bulanık yapısı kullanarak doğrusal davranmayan bir sistemin modellemesini yapmışlardır. Modeli elde edilen nonlinear sistemin Additive-Multiplicative Fuzzy Neural Network (AMFNN) yapısı ile Bulanık parametreleri elde edilmiş ve model üzerinde kullanılmıştır [23].

Ming-Guang Zhang ve arkadaşı, online çalışan bir sistemde RBF (3 katmanlı ileri beslemeli 1 gizli katmanlı) sinir ağı kullanarak elde edilen model ve denetleyici ikinci derece bir sistemde üzerinde klasik bir PID denetleyici ile çalıştırılmış ve RBF sinir ağı ile elde edilen model, denetleyicinin üstünlüklerini ortaya konmuşlardır [24].

Manish Sagar ve arkadaşları NARX ve LRN (layer recurrent network) yapısındaki yapay sinir ağlarını Hodgkin-Huxley Modeli üzerinde kullanarak modelleme performanslarını karşılaştırmışlardır [25].

Eduard Petlenkov ve arkadaşı klasik NARX yapısı ile elde edilen model ile Anarx yapısı ile elde edilen modelleri nonlinear sistemler için uygulamış ve model performanslarını karşılaştırmalı olarak ortaya koymuşlardır [26].

Vladimir Koroman ve arkadaşı bir elektrohidrolik servo sisteminde ileri beslemeli yapay sinir ağı zaman gecikmesi kullanılarak modellemesi sistem normal çalışmasında ve sistemde yağ sızıntısı varken modellemiş, her iki model karşılaştırarak sistemden kaynaklanan hatanın ne olduğu grafiklerle ortaya koymuşlardır [27].

Stefan Minchev, akım ve gerilim transformatörlerini yapay sinir ağı ile modelleyerek dinamik sistemlerin modellenmesine dair farklı bir uygulama yapmışlardır. Stefan akım ve gerilim transformatörlerinin matematik modellerini de elde etmiş, yapay sinir ağları ile elde edilen modellerin gerçek zamanlı çalışmadaki üstünlüklerini ortaya koymuştur [28].

Waldemar Graniszewski ve arkadaşı, nonlinear bir kontrol sisteminde yerel doğrusal haritalama ve global yaklaşım sağlayan Fourier serileri tekniklerini 2 farklı modelleme aracı olarak Yapay Sinir Ağları için kullanmış ve karşılaştırmıştır [29].

L. Shi ve N. Sepehri pnomatik pistonun hareketinin kontrolünü yaptıkları çalışmada adaptif yapay sinir ağı tabanlı bulanık denetleyici ile gerçekleştirmiştir [30].

1.2.2 Genetik Algoritma ile Bulanık Denetleyici Parametre Optimizasyonu

Bulanık denetleyici kuralları sistem uzmanının tecrübesine dayanarak deneme yanılma yöntemi ile bulunmakta idi. Araştırmacılar bu kuralların daha kısa sürede daha doğru bulunması için üzerinde çalıştılar. Yapay Sinir Ağları bu parametrelerin elde edilmesinde kullanılmış fakat öğrenme sırasında yerel minimumlarda takılması sinir ağlarının zayıf noktası olarak karşımıza çıkmıştır [31]. Genetik biliminin ve doğal seçim kurallarının mantığına dayalı yapısıyla Genetik Algoritma, Bulanık denetleyici kural tablosunun ortaya çıkarılması ve pek çok kontrol uygulamasında, kontrol sistemlerin optimizasyonunda yaygın olarak kullanılmaktadır [32]. Bulanık denetleyici parametre optimizasyonuna ilişkin çalışmalar üyelik fonksiyonu sınır değerleri, üyelik fonksiyonu şekillerinin belirlenmesi ve aksiyon değerlerinin bulunması olarak 2 şekilde karşımıza çıkmaktadır [33].

Y. Işık yaptığı çalışmada uçuş kontrol sisteminde önemli bir yer tutan irtifa kontrolünün genetik bulanık denetleyici kullanılarak tasarımını incelemiş ve geniş gövdeli, dört motorlu jet yolcu uçağına ait elde edilen uygulama sonuçlarını, klasik PD denetleyici kullanılarak elde edilen sonuçlarla karşılaştırmıştır [34].

C. Şafak yaptığı çalışmada yapay sinir ağı metodu kullanarak pnomatik motoru modellemiş, kullandığı Sugeno tipi bulanık denetleyici kurallarını Genetik Algoritma ile optimize etmiştir. Şafak modelde elde ettiği aksiyon değerleri ile sistemi gerçek zamanlı çalışmada kullanmış ve yörünge takibi çalışması yapmıştır [35].

V. Topuz bir proses kontrol sistemi için (proses olarak tank sıvı seviye kontrol sistemi) çevrim dışı çalışma modelinde bulanık denetleyici üyelik fonksiyonlarının değerlerinin ve aksiyon ağırlık değerlerini genetik algoritma ile bulmuştur [17].

O. Kaan Erol yaptığı çalışmada doğrusal olmayan sistemlerin bulanık modellenmesi ve çarpışma yaklaşımlı genetik algoritma yöntemi ilgili sistem için kullanılacak denetleyici parametreleri optimizasyonu uygulaması yapmıştır. Çalışmanın son aşamasında elde edilen farklı denetleyiciler ikinci derece bir sistem modeli üzerinde çalıştırılarak karşılaştırılmıştır [36].

T. Um, Y. Joo, Y. Kong yaptıkları çalışmada, genetik algoritma kullanarak modellenmiş sistemin optimizasyonu gerçekleştirmişlerdir. Yapılan optimizasyon çalışmasında elde edilen modelin dinamik karakteristiğinden daha hızlı, kararlı olduğu gözlenmiştir [37].

J. Wang, N. Daw genetik algoritma kullanarak pnomatik silindirin sürtünmesi tanımlanmıştır. Çalışmada, evrim fonksiyonu olarak RMS değeri kullanılmıştır [38].

Teo Lian Seng ve arkadaşları yaptıkları çalışmada, bir arabanın kendi kendine park edebilmesi için kullandıkları Bulanık denetleyiciye ait üyelik fonksiyonu sınır değerlerini Genetik Algoritma ile optimize etmiş ve aynı sistemi PID ile kontrol edecek parametreleri Genetik algoritma ile elde ederek iki denetleyicinin performanslarını karşılaştırmışlardır [39].

JingYuan Zhang ve arkadaşı yaptıkları çalışmada, bir fırına ait sıcaklığın kontrol edilmesi deneyinde Bulanık denetleyiciye ait kuralları Genetik Algoritma ile optimize etmiş ve elde ettikleri ikinci derece transfer fonksiyonunda denetleyicilerin performanslarını karşılaştırmışlardır [40].

Xiao-Ping Zong ve arkadaşı yaptıkları çalışmada, 2. derece bir sistemin Bulanık denetleyici girişleri olan hata ve hatadaki değişim ile denetleyici çıkışına eklediği kazanç bloklarına ait katsayıları genetik algoritma ile optimize etmiş ve optimizasyon öncesi sistem performansı ile karşılaştırmıştır [41].

Dongrui Wu Woei Wan Tan yaptığı çalışmada, matematik modelini elde ettiği bir sıvı seviye sisteminin Bulanık denetleyici ile kontrol edilmesi işleminde giriş değişkeni üyelik fonksiyonu aralıklarını değişken kabul etmiş ve bu çalışma aralıklarını Genetik Algoritma ile optimize ederek sistemin referans eğriyi izlemesini sağlamıştır [42].

Hisao Ishibuchi ve arkadaşı yaptıkları çalışmada, veri madenciliğinde kullanılan Bulanık kuralların Genetik Algoritma ile optimizasyonunu sağlamışlardır. MOGA (Çok Amaçlı Genetik Algoritma) yapısıyla kural optimizasyonuna dayanan algoritma en az uzunluktaki / sayıdaki en etkin kuralları bulmayı hedeflemiştir [43].

Rubén Lagunas-Jiménez ve arkadaşı bir DC servo sistemin Bulanık denetleyici ile kontrolünü sağlamak için denetleyici kurallarını matematik modeli üzerinden çevrimdışı olarak optimize etmek için Genetik Algoritmayı kullanmış ve elde ettiği tasarım eğrilerini durum uzay eğrileri ile karşılaştırmıştır [44].

Jingzhe Yu ve arkadaşları, ikinci derece transfer fonksiyonunu çıkardıkları sarkaç düzeneğinin kontrolünde kullanılan Bulanık kurallarını ve üyelik fonksiyonu şekillerinin elde edilmesinde Genetik Algoritma kullanmışlardır [45].

Marcio A. T. de Sousa ve arkadaşı gerçek zamanlı olarak çalıştırdıkları bir sarkacın kontrolünde kullandıkları Takagi-Sugeno Bulanık denetleyicisinde kuralların ağırlıklarını optimize etmekte kullanmış ve genetik algoritmanın gerçek zamanlı çalışan kontrol sistemlerindeki performansına örnek vermişlerdir [46].

Khaled Belarbi ve arkadaşları yaptıkları çalışmada, Bulanık denetleyici ile yaygın olarak kontrol edilen araba-çubuk sisteminde kullandığı Mamdani türü denetleyicinin kural tabanının oluşturan kuralları Genetik Algoritma ile en az kural kullanacak şekilde optimize etmiş ve bu kurallar ile sistemi kontrol etmişlerdir [47].

P. T. Chan ve arkadaşları da gene klasik bir açık çevrim nonlineer problem olan araba çubuk sisteminde birleştirilmiş bulanık (fused Bulanık) yapısı kullanmış, denetleyicinin kurallarını kural tablosu dönüşümü yaparak Genetik algoritma ile optimize etmiş, genetik algoritma için kullanılan arama uzayını küçültmüşlerdir [48].

Francesco Cupertino ve arkadaşları yaptıkları çalışmada, 5 DOF CRS catalyst robotun yörünge izlemesi için tasarlanan Bulanık denetleyici üyelik fonksiyonu şekillerini ve çıkış aksiyon değerlerini Genetik algoritma ile optimize etmiş, PID denetleyici ile performanslarını karşılaştırmıştır [49].

Enwang Zhou ve arkadaşı yaptıkları çalışmada, Bulanık denetleyicinin giriş üyelik fonksiyonu aralıklarını ve kurallarını aynı anda Genetik Algoritma ile optimize etmişlerdir. Yapılan çalışma Iris ve Wine isimli iki veritabanının sınıflandırılmasında kullanılmıştır [50].

Ahmed Abdel Hadi ve arkadaşı yaptıkları çalışmada, iki eklemlili bir robot koluna farklı hedef fonksiyonlar vererek sistemi Bulanık denetleyici ile kontrol etmeyi amaçlamışlardır. Bulanık denetleyiciye ilişkin kurallar Genetik Algoritma ile optimize edilmiş ve diğer optimizasyon algoritmalarına göre gerek zaman gerek performans anlamında daha iyi sonuçlar elde etmişlerdir [51].

1.2.3 KKO Algoritması ile Bulanık Denetleyici Parametre Optimizasyonu

Karınca Koloni Optimizasyonu algoritması, üyelerin işbirliği halinde çalışması mantığını esas alır. Üyeler birbirleriyle doğrudan değil, takip ettikleri çözüm yoluna bıraktıkları iz ile haberleşirler. Bu yöntem gerçek yaşamda karıncaların izledikleri yollara bıraktıkları feromon mantığı ile ortaya çıkmıştır [52].

D. Taşkiran yaptığı çalışmada zaman kısıtlı araç rotalama problemini karınca kolonisi optimizasyonuna dayalı bir yaklaşımla çözmüştür. Çalışmada araç sayısı ve toplam gidilen yolun minimize edilmesi amaçlanmıştır. Bu minimizasyon problemini çözmek üzere biri araç sayısını, diğeri ise toplam katedilen yolu minimize etmeye odaklı iki karınca kolonisi feromon seviyeleri vasıtasıyla haberleşerek bir yardımlaşma anlayışı içerisinde çalışmışlardır [53].

M. Ünal yaptığı çalışmada YSA ile modellediği GuntRT 532 basınç prosesinin kontrolünde kullandığı PID parametreleri çevrim dışı olarak KKO algoritması ile elde etmiş, sonuçları GA sonuçları ile de karşılaştırmıştır [54].

Panta Lucic ve arkadaşı, yaptıkları çalışmada Bulanık üyelik fonksiyonlarını karınca feromeni ve mesafe seçerek şehir içi taşımacılıkta en kısa sürede ulaşımı sağlayacak bir model tasarlamışlardır. Yapılan çalışma karıncaların en kısa yolu izleme mantığını incelemesi bakımından önemlidir [55].

Michelle Galea ve arkadaşı, Bulanık üyelik fonksiyonları olarak tanımladığı hava gözlemi ve buna bağlı yapılabilecek aktivitelerin kural tabanının KKO ile optimize edilmesi üzerine yaptıkları çalışmada oldukça tatmin edici sonuçlar elde etmişlerdir [56].

J. Paetz atom bağları ile ilgili çalışmasında Neuro-Bulanık ile modellenmiş yapının Bulanık kurallarını KKO ile optimize ederek, optimizasyon algoritmalarının Tıp alanında da kullanılabileceğine dikkat çekmiştir [57].

Chia-Feng Juang, yaptıkları çalışmada FPGA (Field Programmable Gate Array) çip kullanarak bir su tankının sıcaklığını kontrol edecek Bulanık kurallarını KKO ile optimize etmişlerdir [58].

Üyelik fonksiyonlarının şekillerinin optimizasyonunda Tabu Search, Genetik ve Gradient Algoritma gibi yöntemler kullanılmıştır. Yongsheng Zhao ve arkadaşı ise 2. derece bir transfer fonksiyonu için kullandıkları Bulanık denetleyici üyelik fonksiyonu sınır değerlerini KKO ile optimize etmişlerdir [59].

Baojiang Zhao ve arkadaşı yaptıkları çalışmada ters çevrilmiş bir sarkaç düzeneğinin kontrolünde bulanık denetleyici kullanmış ve denetleyiciye ilişkin kuralların aksiyon değerlerini ve giriş üyelik fonksiyonu sınır değerlerini KKO ile optimize etmişlerdir [60].

Xiaoyu Song ve arkadaşı yaptıkları çalışmada Genetik Algoritmanın global arama becerisi ve KKO'nun çözüm arama etkinliğini bir arada kullandığı hibrid bir “iş alım planlama” problemi üzerinde denemiş etkin sonuçlar elde etmişlerdir [61].

1.2.4 Su Seviye Sistemleri Üzerine Yapılan Çalışmalar

Chengwei Li ve arkadaşı, proses kontrol alanında yaygın olarak kullanılan sıvı seviye sistemini PID denetleyici ile kontrol etmeyi amaçlamışlardır. Li ve arkadaşı kullandıkları PID parametrelerini ikinci derece transfer fonksiyonu üzerinde genetik algoritma kullanarak optimize etmiş ve sistemin referans eğriye oturmasını sağlamışlardır [3].

Tianyou Chai ve arkadaşı, kullandıkları buhar jeneratörünün sıvı seviyesini farklı denetleyiciler ile kontrol etmişlerdir. Üçüncü derece transfer fonksiyonu olarak ortaya koydukları sistemi GPE(Gaussian partition with evenly spaced midpoints) yaklaşımı ile tasarladıkları Bulanık denetleyici ile de kontrol etmişlerdir. Chai ve arkadaşı elde ettikleri parametreleri gerçek zamanlı olarak sistemde denemiş ve Bulanık denetleyicinin PID denetleyiciye göre üstünlüklerini ortaya koymuşlardır [62].

Man Gyun Na yaptığı çalışmada nükleer buhar jeneratörünün sıvı seviye kontrolünü 9 kurallı bir Bulanık denetleyici ile kontrol etmiştir. Üyelik fonksiyonları ve kural ağırlıklarını genetik algoritma ile optimize eden Na, elde ettiği sistem davranışlarını bir P-I denetleyici çıkışı ile karşılaştırmıştır. Kural sayısı oldukça az olmasına rağmen parametreleri optimize edilen Bulanık denetleyici daha hızlı ve referansa yakın cevaplar vermiştir [63].

Min-Soeng Kim ve arkadaşları nükleer güç ünitesi buhar jeneratörünün modelini Bulanık ile elde ettikten sonra, sıvı seviye kontrolünü Bulanık denetleyici ile sağlamışlardır. Kim ve arkadaşları denetleyicinin parametrelerinin genetik algoritma ile optimize etmiş ve elde ettiği değerleri gerçek zamanlı olarak sistem kontrolünde kullanmışlardır [64].

Satean Tunyasirut ve arkadaşı yaptıkları çalışmada yatay bir su tankındaki sıvı seviyesini Bulanık denetleyici ve MCS51 mikrodenetleyicisi ile kontrol etmişlerdir. Elde edilen denetleyici davranışı PID denetleyici davranışı ile oturma zamanı, aşım, kalıcı hal hatası değerlerine göre karşılaştırılmış ve Bulanık denetleyicinin üstünlükleri ortaya konmuştur [65].

E. Kaplanoğlu, Gunt RT 512 sıvı seviye kontrol sistemini PID denetimli UDC5000 panelini çıkartıp bilgisayar denetimli bir yazılım kullanmış ve Bulanık denetleyici ile kontrol etmiştir. Erkan, yaptığı çalışmada hata ve hatadaki değişimi

Bulanık denetleyici girişlerine vermiş ve ilgili parametreleri deneme yanılma yöntemi ile elde etmiştir [2].

V.Topuz, tasarladığı sıvı seviye tankında, bulanık denetleyici üyelik fonksiyonlarının değerlerinin ve aksiyon ağırlık değerlerinin genetik algoritma ile bulmuştur. Topuz, kullandığı sistemin transfer fonksiyonunu elde etmiş ve bu fonksiyon üzerinden genetik algoritma ile parametre optimizasyonu yapmıştır [17].

M. Özarslan, Gunt RT 512 su seviye kontrol sistemi için Bulanık denetleyici PicBasic diliyle yazmış ve PIC16F8 microdenetleyicisinde çalıştırarak sistemi kontrol etmiştir. Özarslan, denetleyici girişleri olarak hata ve hatadaki değişim almış, üyelik fonksiyonu, bunların sınır değerleri ve kural tabanını deneme yanılma yöntemi ile tasarlamıştır [66].

Önceki çalışmalardan da görüldüğü gibi, KKO algoritması ve YSA modelleme tekniği GuntRT 512 sıvı seviye sistemi üzerinde kullanılmamıştır. Bu çalışmada ise sistemin kontrolünde kullanılacak bulanık denetleyicinin aksiyon ve sınır değerleri hem Genetik algortima hem de KKO ile optimize edilerek ayrı ayrı bulunmuş, her iki algoritmanın performansı simülasyon ve gerçek zamanlı deneyler ile karşılaştırılmıştır.

Bölüm II' de Gunt RT 512 sıvı seviye sistemini oluşturan donanım yapısı, sistemin çalışma prensibi ve sistem üzerinde tasarlanan kontrol sistemi hakkında bilgi verilmiştir.

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

Bölüm IV'de YSA ile Gunt RT 512 sıvı seviye kontrol sistemin dinamik modeli çıkarılmış, GA ve KKO algoritması ile YSA modeli üzerinden çevrimdışı çalışmada Bulanık denetleyici için optimum değerler bulunmuştur.

Bölüm V'de Genetik-Bulanık denetleyici ve KKO-Bulanık denetleyici ile sisteme farklı referanslar verilerek gerçek zamanlı deneyler yapılmıştır.

Bölüm VI'da, elde edilmiş deney sonuçları ışığında GA ve KKO bulanık denetleyicilerin performansları farklı parametreler ile karşılaştırılmıştır.

Bölüm VII'de ise yapılan çalışmalar değerlendirilmiş, sonraki çalışmalar için öneri ve temenniler de bulunulmuştur.

BÖLÜM II

GUNTRT 512 SIVI SEVİYE KONTROL SİSTEMİ

II.1. GİRİŞ

Bu bölümde öncelikle Gunt RT sıvı seviye kontrol sisteminin çalışma prensibi anlatılmış, sistemde kullanılan parçaların görevleri, sistem üzerinde yapılan değişiklikler ve tasarlanan denetleyici ele alınmıştır. Daha sonra sistemin referans olarak izlemesi istenen yörünge fonksiyonu incelenmiştir.

Endüstri işletmelerin teknik eleman ihtiyacını karşılayan teknik ve endüstri meslek liselerine öğretmen yetiştiren Teknik Eğitim Fakültesi'nde uygulamalı eğitimin önemi büyüktür. Uygulamalarda kullanılan deney setlerinin endüstride kullanılan sistemlere uygun olması yetişen teknik öğretmenin, teknik elemanın kalitesini arttıracak ve endüstride karşılaşılabileceği sorunları en aza indirecektir.

GuntRT 512 sıvı seviye kontrol sistemi endüstride kullanılan sistemlere benzer bir sistemdir. Üzerinde enerji girişi pnomatik, kontrol girişi ise 4-20mA akım olan bir oransal valf, sıvı tankı, 1 fazlı asenkron motor tarafından sürülen bir pompa, dönüş vanası, basınç algılayıcı, basınç / akım dönüştürücüsü (2-6 bar / 4-20mA) 0-62 cm ölçekli seviye tüpü, UDC5000 dijital denetleyici bulunmaktadır. Bulanık kontrolü gerçekleştirmek amacıyla, sistemdeki PID ve ON-OFF kontrol işlemini gerçekleştiren UDC5000 denetleyicisi devre dışı bırakılmış, MATLAB yazılımı ile girişleri hata ve hata değişimi olan bir bulanık denetleyici tasarlanmıştır. Denetleyici çıkışından gelen gerilim bilgisi V/I dönüştürücüden oransal valfe gelerek valf açıklığını değiştirir. Sıvı seviye tüpüne giren sıvı miktarı da bu şekilde ayarlanmaktadır. Bu çalışma ile GuntRT 512 seviye ünitesine bulanık denetleyici ilave edilmiş, denetleyiciye ait parametreler iki farklı optimizasyon algoritması ile elde edilmiştir. Farklı denetleyicileri kullanmanın yanında denetleyicilerin parametrelerinin optimizasyonunda alternatif algoritmalar kullanarak bunların performanslarının karşılaştırılması öğrencilerin sistem üzerinde farklı yöntemlerle

alışmasına olanak saęlayacak, denetleyici davranışlarını aynı sistem üzerinde inceleyebilecektir.

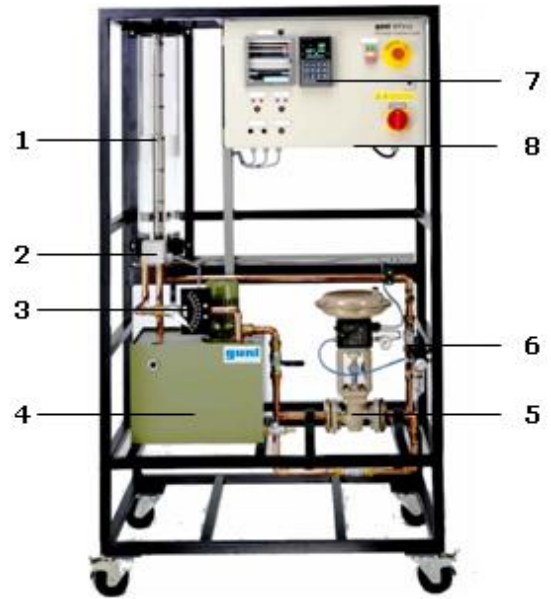
II.2. SİSTEMİN GENEL YAPISI

GuntRT 512 sıvı seviye kontrol sisteminin yapısını incelemek için sistem sekiz bölüme ayrılmıştır. Bu bölümler Şekil II.1’de ayrıntılı olarak gösterilmektedir.

0 - 62cm arasında deęişen yükseklik seviyesinde sıvı seviye kontrolü için kullanılan sistem denetleyici aracılığıyla pnomatik kontrol valfinden geçen debiyi dolayısıyla tüpe giden sıvı miktarını ayarlamaktadır. Temel elemanların görevleri ve sistemin alışması ile ilgili detaylar sonraki bölümlerde ayrıntısı ile ele alınacaktır.

II.2.1. Sistemin Temel Paraları :

1. Sıvı seviye tüpü
2. Basın dönüştürücüsü
3. ıkış Vanası
4. Su tankı
5. Pnomatik kontrol valfi
6. Basın ayar filtresi
7. Denetleyici
8. Sistem kontrol kasası

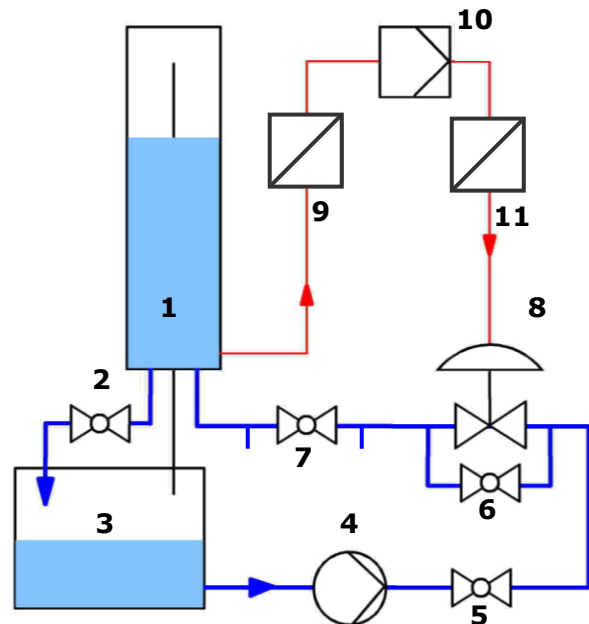


Şekil II. 1 Gunt RT 512 Sıvı Seviye Sistemi [2]

II.2.2 Sistemin Bağlantı Şeması

Şekil II.2’de bağlantı şeması verilen sistem temel olarak sıvı, hava ve elektrik işaretlerinin sistem içindeki değişimine göre davranmaktadır. Yazılımı tasarlanan denetleyici ile gönderilen kontrol işareti ile sıvı seviyesi kontrolü sağlanan sisteme çıkış valfindeki değişim ile bozucu verilebilmektedir. Sistem içerisindeki tüm elektriksel işaret akım bilgisi olarak iletilmekte ancak DAQ kartında veri haberleşmesi gerilim bilgisi ile yapılmaktadır.

- 1. Seviye silindiri**
- 2. Çıkış Valfi**
- 3. Sıvı Tankı**
- 4. Sıvı Pompa**
- 5. Konik Valf**
- 6. Kaskat için bypass**
- 7. Kaskat için hortum bağlantısı**
- 8. Pnömatik Kontrol Valfi**
- 9. P / I dönüştürücü**
- 10. Denetleyici**
- 11. Pnömatik konumlandırıcı**



Şekil II. 2 Gunt RT 512 İşlem Diagramı [2]

II.2.3 Sistemin Temel Özellikleri

Boyutlar :

W*l*h	1000*700*1735mm ³
Ağırlık :	110kg

SIVI pompasi :

Max. akış oranı: 110 litre/dk
Max. Pompa: 6.5 m

Sıvı Tankı :

Kapasitesi : 28 litre

Besleme :

Kompresör hava : 2 bar
Gerilim: 220V/50Hz

Sıvı Seviye Tüpü :

Sıvı kapasitesi : 7 litre
Yükseklik : 0 - 62 cm

II.3 SIVI SEVİYE SİSTEMİ ELEMANLARI

II.3.1. Denetleyici

Sistem üzerinde üretici firma tarafından konulmuş, mikroişlemci tabanlı bir Honeywell marka UDC 5000 denetleyici vardır. (Şekil II.3) Bütün ölçümler sayısaldır, sayısal olarak işlenir ve analog sinyale çevrilerek aktuatör elemana gönderilir. Denetleyicinin giriş ve çıkış değerleri 4-20mA arasındadır. Denetleyici parametreleri, kontrol paneli üzerindeki menüden basmalı butonlar yardımı seçilerek değiştirilebilir. UDC 5000 ile seçilen denetleyici tipine göre (PID, P, PI, PD, ON-OFF) sistem kontrol edilebilir. Yapılan çalışmada, PC tabanlı denetleyici kullanıldığından dolayı bu denetleyici devre dışı bırakılmıştır.



Şekil II. 3 UDC 5000 Denetleyici

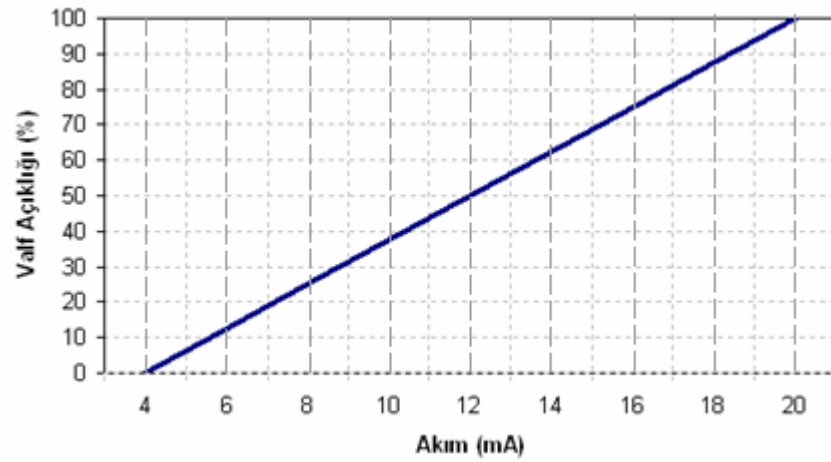
II.3.2. Pnematik Oransal Valf (I/P)

Kullanmış olduğumuz sistemde Samson firmasının üretmiş olduğu tip 3241-7 pnematik oransal valf kullanılmaktadır. Bu pnematik valf, 5 bar basınç değerinde bir hava ile beslenerek, 4-20 mA'lık bir kontrol sinyali ile kontrol edilmektedir. Denetleyici çıkışındaki işarete göre sisteme verilecek su miktarı pnematik valf üzerinden ayarlanır. Şekil II.4'de gösterilen valfın kumanda devresi elektrikli (4-20mA), güç devresi ise pnematiktir.



Şekil II. 4 Pnematik oransal valf

Pnematik valf sistemde uygulanan kontrol sinyaline göre oransal olarak açılıp kapanarak sıvı seviye tüpüne giden suyun debisini ayarlamaktır. Pnematik valf 4-20mA arası doğrusal olarak çalışmaktadır. Sistemin kontrolünde kullanılan DAQ kartı gerilim çıkışlı olduğu için veri toplama kartından gönderilen gerilimi akıma dönüştürmek için V-I dönüştürücü modül (Phoenix Mcr) kullanılmıştır. Bu modül ile veri toplama kartından çıkan 2-10V'luk gerilim, 4-20mA'lık akıma dönüştürülerek pnematik oransal valfe uygulanmaktadır. Valfin 2–10V arası çalışması Şekil II.5'de gösterilmektedir. Şekil II.5'den anlaşılacağı gibi pnematik valf 4mA'den sonraki akımlarda açılmaya başlamakta ve 20mA'de maksimum açıklığa ulaşmaktadır.



Şekil II. 5 Pnematik oransal valf, akım / valf açıklığı grafiği

II.3.3. Basınç Regülatör Ünitesi

Şekil II.6'da gösterildiği gibi sistem üzerinde oransal pnematik valfe bağlı bir basınç regülatörü bulunmaktadır. Kompresörden gelen basınçlı havadaki titreşimleri ve fazla basıncı, istenen basınç değerinde düzenler. Sistemin beslemesi için basınç değeri 6 bar, valf için 5 bardır.



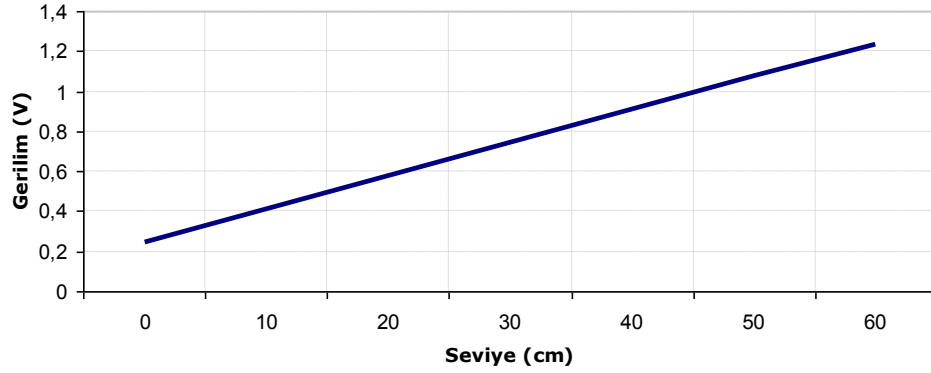
Şekil II. 6 Basınç Regülatörü



Şekil II. 7 PMC 133 basınç dönüştürücüsü

II.3.4. P / I Dönüştürücüsü

Endress+Hauser Firmasına ait dönüştürücü Şekil II.7’de gösterilmektedir. Dönüştürücü, sıvı seviye tüpünden aldığı seviye/basınç bilgisini 4–20 mA akım değerine çevirir.



Şekil II. 8 Sıvı seviye tüpü seviye-gerilim grafiği

Şekil II.8’de bu dönüştürücünün sıvı seviye tüpündeki su miktarına göre 64Ω ’luk bir direnç üzerinde oluşturduğu gerilim değerleri gösterilmiştir. Grafikten de anlaşılabileceği gibi tüpteki sıvı seviyesi 0cm’de iken 0,245V, 62cm’de iken 1.275V’luk gerilim sistemden okunmakta ve sıvı seviyesinin artmasıyla basınç dönüştürücüsünden gelen gerilim doğrusal olarak artmaktadır. Sıvı seviye sisteminden okunan gerilim bilgisi olduğu için gerek YSA ile oluşturulan modelde gerek referans yörünge eğrilerinde yükseklik bilgisi gerilim olarak verilmekte ancak kullanıcı bu gerilim değerinin yükseklik olarak karşılığını programın ara yüzünde görebilmektedir.

II.3.5. I / P Dönüştürücüsü (Pnmatik Konumlandırıcı)

Şekil II.9’da gösterilen pnmatik konumlandırıcı, denetleyiciden gelen akım bilgisini elektropnmatik dönüştürücüden geçerek basınç bilgisine çevrilir. Gelen basınç, bir yay ile dengelenen diyafram üzerinde bir kuvvet oluşturur ve diyaframa bağlı valf kolunun açıklığını azaltır veya artırır.



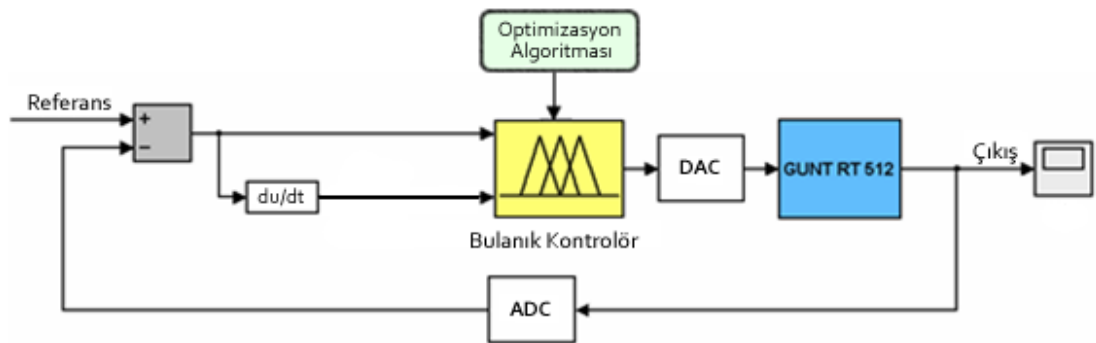
Şekil II. 9 Pnmatik Konumlandırıcı

II.4 SİSTEMİN DAVRANIŞI ve TASARIMI

GuntRT 512 sıvı seviye sistemi bilgisayarla kontrol edileceğinden dolayı üzerinde bazı değişiklikler yapılması gerekmiştir. Bu değişiklikler sistemin temel çalışmasını değiştirmemekte, sadece denetleyicinin çalışmasını düzenlemektedir. Sistemde yapılan en önemli değişiklik UDC 5000 denetleyicinin devre dışı bırakılarak yerine Matlab programı aracılığıyla kullanılacak olan bir bulanık denetleyici eklenmesidir.

Sıvı seviye 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 PCI 1711) bulunmaktadır.

Sistemin basınç bilgisi, basınç dönüştürücüsünden akım olarak alınmaktadır. Veri toplama kartına girişin gerilim olması gerektiğinden dolayı basınç dönüştürücüsünden alınan akım, 64Ω 'luk bir direnç üzerinden gerilime dönüştürülmüştür. Ayrıca pnomatik oransal kontrol valfi akım kontrollü olduğu için veri toplama kartından çıkan gerilim, Phoenix marka gerilim-akım dönüştürücü modül ile akım bilgisine dönüştürülerek pnomatik oransal valfin kontrolü sağlanmıştır. GuntRT 512 sıvı seviye ünitesinin kontrolü bulanık denetleyici ile gerçekleştirilmiştir. Bulanık denetleyicinin girişleri, hata ve hatadaki değişim olarak seçilmiştir. Denetleyici çıkışı ile oransal valfin konumu değiştirilecek, verilecek referansı takip etmesi sağlanmıştır.



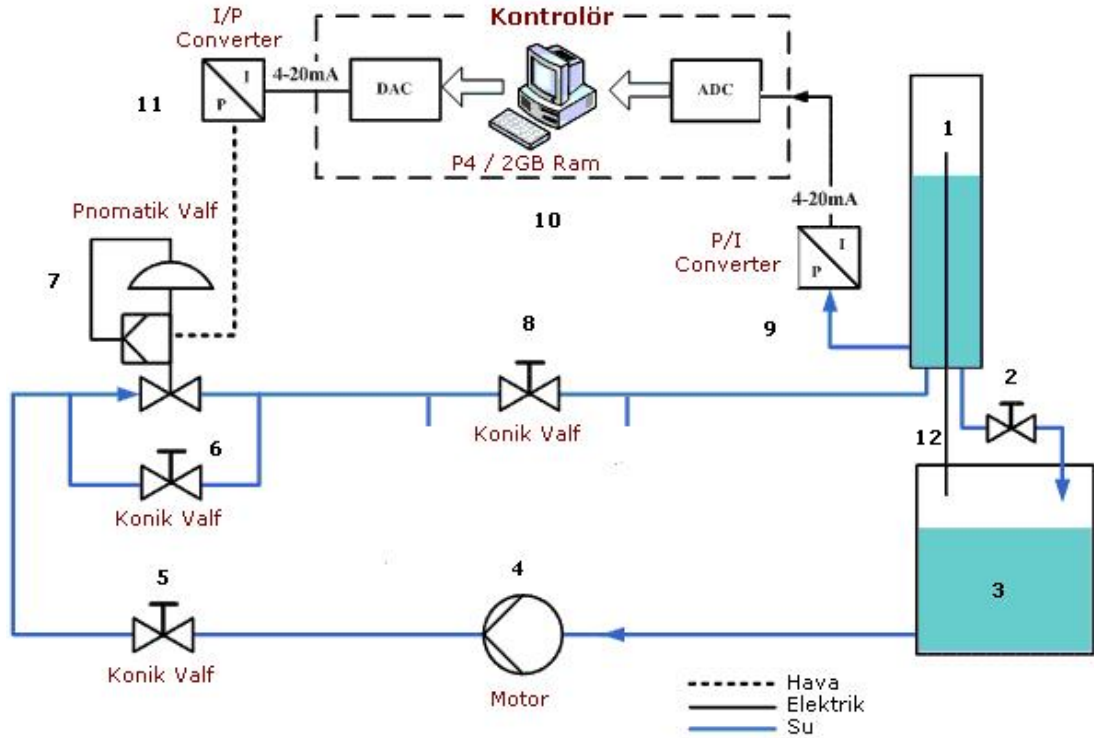
Şekil II. 10 Tasarlanan kontrol sisteminin genel yapısı

II.4.1. Sistemin Çalışma Prensibi

Endüstride sıvı seviye sistemleri yaygın olarak kullanılmaktadır. Bu sistemler genellikle PID (Proportional, Integral, Derivative) ve ON-OFF kontrol yöntemleriyle kontrol edilmektedir. Bu yöntemlerle kontrol edilen sistemlerde aşım ve kararlı hal hatası oluşmaktadır, bu da sistemin hassasiyetini olumsuz yönde etkilemektedir. Dolayısıyla sistemde bulunan UDC 5000 denetleyicisi yerine girişleri hata ve hatadaki değişim olan bir bulanık denetleyici (10) kullanılmıştır. Sistemin işlem diyagramı Şekil II.11’de verilmiştir. Sistemde amaç; sıvı seviye tüpü (1) içerisindeki sıvının istenilen referans değerine ulaşmasıdır.

Referans yörünge sisteme verildikten sonra, denetleyici sıvı seviye tüpüne bağlı basınç dönüştürücüsünden (9) gelen bilgiyi, sıvının o andaki seviyesini algılar ve referans değeri ile karşılaştırarak gerekli kontrol sinyali V/I dönüştürücüden geçtikten sonra akım/basınç dönüştürücüsüne (11) , buradan da pnomatik kontrol valfine (7) gönderir. Sıvı tankındaki (3) sıvı, sıvı pompası (4) tarafından sistemin çalışmasıyla birlikte sürekli olarak pnomatik kontrol valfinin girişine gönderilmektedir.

Pnomatik kontrol valfini denetleyiciden gelen sinyale göre oransal olarak açılıp kapanarak girişinde bulunan sıvıyı sıvı seviye tüpüne gönderir. Bu arada sıvı seviye tüpünün alt kısmındaki tahliye borusu (12) sıvının sıvı tankına boşalmasını sağlar. Sistemdeki tam açık konumda bulunan ve çıkış valfi (2) sıvının boşalması elle ayarlanabilmekte ve sistem çalışırken bozucu etki ithalinde valf kullanılmaktadır. Dolayısıyla sıvı seviyesi referans değere ulaştığında dahi çıkış valfi tam kapalı duruma getirilmeyecek, gelen sıvı ile giden sıvı dengede tutan bir açıklıkta kalacaktır.



Şekil II. 11 Kontrol edilen sistemin blok diyagramı

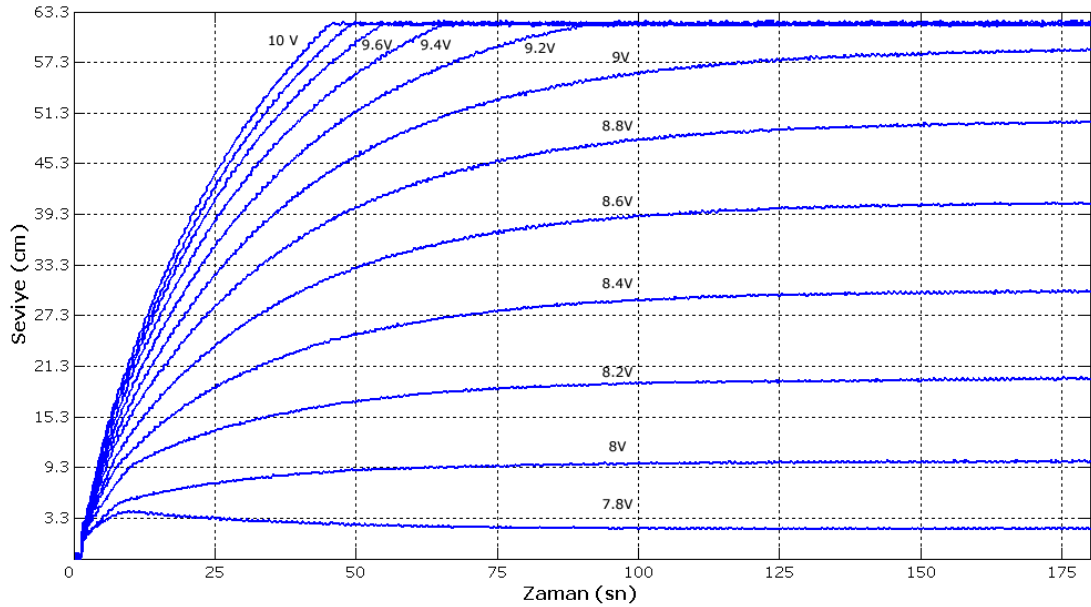
- | | |
|--------------------------------|--|
| 1. Sıvı Seviye Tüpü | 7. Pnömatik Kontrol Valfi |
| 2. Sıvı çıkışı için konik valf | 8. Kaskat bağlantı için konik valf |
| 3. Sıvı Tankı | 9. Basınç / Akım Dönüştürücü |
| 4. Sıvı Pompası | 10. Denetleyici (PC, DAQ Kartı, V / I Dönüştürücü) |
| 5. Konik Valf | 11. Akım / Basınç Dönüştürücü |
| 6. Bypass için Konik Valf | 12. Sıvı Tahliye Borusu |

II.4.2. Sistemin Dolum / Boşalım Eğrileri

GuntRT 512 sıvı seviye kontrol sisteminin bulanık kontrol tasarımının gerçekleştirilmesi için öncelikle sistemin davranışının iyi anlaşılması gerekir. Dolayısıyla sistem üzerinde bulunan pnömatik kontrol valfine çeşitli gerilim sinyalleri uygulanarak, sıvı seviye tüpündeki seviye değişimleri gerilim cinsinden incelenmiştir. Bu işlemin sonucunda sistemin açık çevrim cevabı elde edilmiştir.

Elde edilen verilerden, sistem değişkenleri olarak kontrol girişi ve seviyedeki değişim ele alınmıştır. Pnomatik oransal valfin çalışma aralığı 4-20mA'dır.

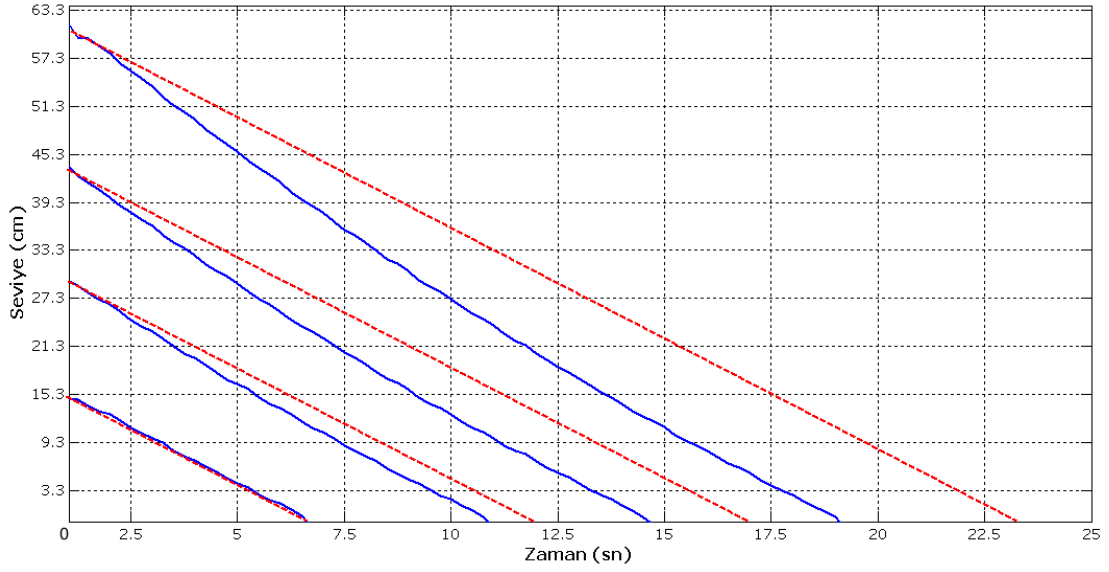
Ancak kullanılan akım gerilim dönüştürücü ile 2-10V çalışma aralığı elde edilmiştir. Çeşitli kontrol sinyalleri için sistemin davranışı Şekil II.12' de görülmektedir. Sistem, verilen kontrol sinyali 7.6V'un üzerinde olduğu zaman sıvı seviye tüpüne dolum yapabilmektedir. Sıvı seviye tüpünün altında bulunan ve her sıvı seviyesinde tahliye yapan tahliye borusundan sürekli su çıkışı olduğu için bu gerilim değerinin (7.6V) altındaki girişlerde sistem dolum yapamamaktadır. Ayrıca Sıvı seviye tüpünde 1.275V (62cm)'den sonra da tahliye borusundan sıvı tahliyesi olduğu için sıvı seviyesi bu seviyenin üzerine çıkamamaktadır.



Şekil II. 12 Sistemin basamak girişlere verdiği cevaplar

Şekil II.13'de ise sistemin boşalma eğrileri verilmiştir. Eğrilerinden de görüleceği gibi farklı seviyelerdeki sıvının boşalma hızları / süreleri seviye ile orantılı değildir. 0.5V (15cm)'den boşalma süresi yaklaşık 7 saniye iken 1.245V (60cm)'den boşalma süresi yaklaşık 19sn sürmektedir. Dolayısıyla tüp içerisindeki sıvı seviyesinden kaynaklanan basınç, sıvının boşalma süresini de etkilemektedir. Şekil II.13'de verilen grafiklerde sürekli çizgi ile verilen eğriler 0.5 V (15cm), 0.745V (30cm), 1V (45cm) ve 1.245V (60cm) seviyelerinden başlayan boşalma

eğrilerini, kesik çizgiler ise boşalmanın oransal olması durumunda izlenecek yolu göstermektedir.

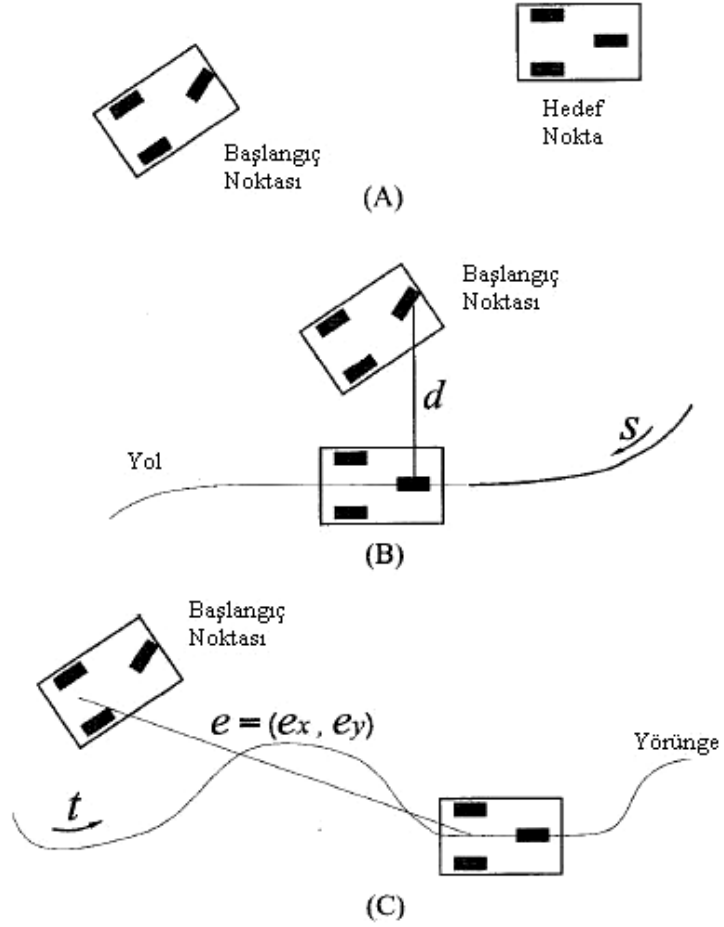


Şekil II. 13 Sistemin farklı seviyelerden boşalma eğrileri

II.5. YÖRÜNGE TAKİBİ

Sistemin kontrolü yapılırken referans olarak basamak giriş, basamak geçişleri gibi sabit referanslar verilebilir. Ancak sistemin ilk konumundan itibaren istenilen seviye değerlerinde seyretmesi için yörünge şeklinde referans verilebilir. Yörünge referansı ile sistem, ilk konumdan son konuma ve ara değerlerde hangi konumda olması gerektiğini bilecektir. Bu denetleyici performansındaki sürekliliği görebilmek için de faydalı olacaktır.

Başlangıç seviyesinden son seviyeye kadar hareketi esnasında yer değiştirme yollarını belirleyen noktalar kümesine yörünge denir [67]. Hareketin aktivitesine göre, yörünge planlaması, noktadan noktaya hareket, yol takibi veya yörünge takibi şeklindedir. Şekil II.14'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 [21].



Şekil II. 14 (A) noktadan noktaya (B) yol takibi (C) yörünge takibi [35]

Yörünge tanımında birçok matematiksel fonksiyon kullanılabilmektedir. Kübik fonksiyon, hareket eğrisinin üretilmesinde kullanılmaktadır. Bu eğri, düşük dereceli eğriler yerine polinom eğrilerinin ara değer kestirimi için daha tahmin edilebilirdir [68].

II.5.1. Sistemde Kullanılan Yörünge Denklemi

Sıvı seviye kontrol sisteminde, seviye tüpünün içerisindeki sıvının yüksekliğinin kontrolü gerçekleştirilmektedir. Burada istenen, seviye tüpünün içindeki sıvının ilk değerinden istenilen seviye değerine, Şekil II.15’de 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 matematisel ifadesi Denklem II.1 de verilmiştir [69].

$$\begin{aligned}
V_y &= V_0 & t < t_0 \\
V_y &= f_3 t^3 + f_2 t^2 + f_1 t + f_0 & t_0 < t < t_s \\
V_y &= V_s & t > t_s
\end{aligned} \tag{II.1}$$

Burada V_y , yörünge değeri, t_0 gecikme zamanı, f_0, f_1, f_2 ve f_3 katsayılar, t ise zamandır. Katsayılar sırasıyla;

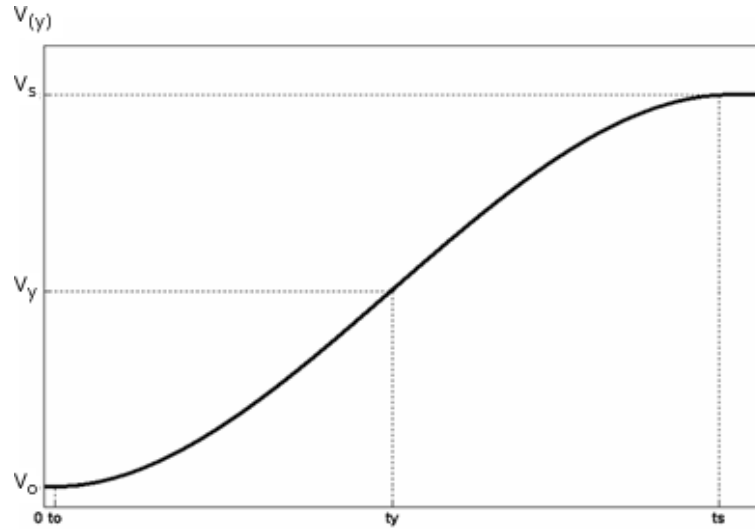
$$t_s = 3\epsilon \frac{V_s - V_0}{V_s + V_0} \tag{II.2}$$

$$f_0 = V_0 \quad f_1 = 0$$

$$f_2 = -3 \frac{V_0 - V_s}{t_s^2}$$

$$f_3 = 2 \frac{V_0 - V_s}{t_s^3}$$

Burada; t_s oturma zamanı, ϵ eğimle ters orantılı katsayı değeri, V_s istenilen referans seviyesi ve V_0 başlangıç noktasıdır. Yapılan çalışmada kullanılan yörünge grafiği Şekil II.15’de gösterilmiştir.



Şekil II. 15 Çalışmada kullanılan yörünge grafiği [69]

BÖLÜM III

ESNEK PROGRAMLAMA

III.1. GİRİŞ

Esnek programlamayı (SC) oluşturan yapay sinir ağları, bulanık mantık ve evrim algoritmalarından bir veya birkaç tanesinin birlikte kullanılması ile oluşan yöntemlerin uygulanması son yıllarda gittikçe önem kazanmıştır. Bu yöntemlerden yapay sinir ağları insanın sinir ve öğrenme yapısını, Genetik algoritma doğal genetiği, KKO var olan çözümlerden en kısa olanın bulunmasını, bulanık mantık ise insanın düşünme biçimini modelleyerek çözüm arar. Şekil III.1’de esnek programlamayı oluşturan yöntemler görülmektedir [17]. Esnek programlama, geleneksel (hard) programlamadan farklı olarak kesin olmayan, yaklaşım ile elde edilen çözümlere yer veren insan zekasını rol model olarak alır ve yeni durumlara adapte olabilir. Hard programlama ise önceden belirlenmiş bir algoritmayı kullanan klasik anlamdaki programlama metodudur. İşletim sistemleri, ofis yazılımları, oyunlar bu gruba girer. Bu tür uygulamalar yazıldığı şekilde çalışan süreç içinde alternatif üretmeyen yazılımlardır. Dolayısıyla bir işi belirli bir şekilde değil, değişen ve gelişerek işlem yapan yazılımlar esnek programlama sınıfına girerler.



Şekil III. 1 Esnek programlamayı oluşturan programlama teknikleri [17]

III.1. YAPAY SINIR AĞLARI (YSA) İLE SİSTEM TANIMLAMA

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 gözlemleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin sebebi insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir. Yapay sinir ağları ile ilgili ilk gelişme McCulloch ve Pitts'in 1943 yılında oluşturdukları temel sinir hücresi tasarımı ile ortaya çıkmıştır. [70]

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 bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur.

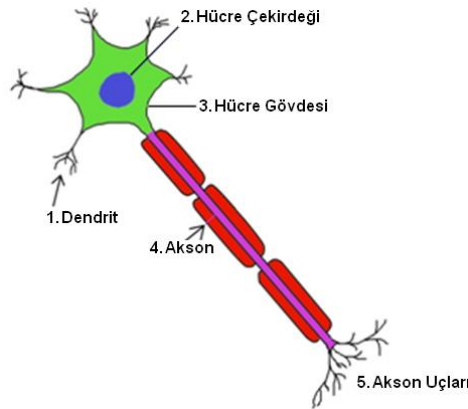
YSA'lar, belirli katsayılar ile ağırlıklandırılarak birbirlerine bağlanmış birçok işlem biriminden (nöronlar) oluşan matematiksel sistemlerdir. Bir işlem birimi, aslında sık sık transfer fonksiyonu olarak anılan bir denklemdir. Bu işlem birimi, diğer nöronlardan sinyalleri alır; bunları birleştirir,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

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. Başka bir ifadeyle, fonksiyonun temel yapısını ağırlıkların büyüklüğü ve işlem elemanlarının işlem şekli belirler. YSA'ların davranışları, yani girdi veriyi çıktı veriyeye nasıl ilişkilendirdikleri, ilk olarak nöronların transfer fonksiyonlarından, nasıl birbirlerine bağlandıklarından ve bu bağlantıların ağırlıklarından etkilenir [71].

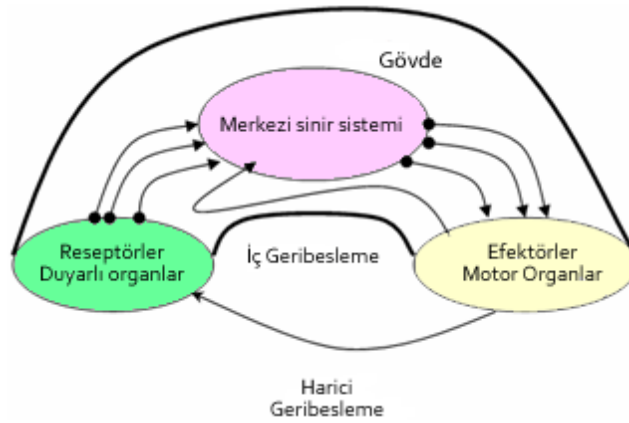
III.1.1. Biyolojik Sinir Ağları

Biyolojik sinir ağının temel inşa 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. Şekil III.2'de 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 uçlarından diğer hücrelere iletilir. Akson uçları ile dendrit bağlantısı ise sinaps olarak adlandırılır. Sinaps, nöronlar arasında elektrokimyasal bağlantıyı sağlamaktadır.



Şekil III. 2 Doğal sinir hücresi [54]

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.3’de gösterildiği gibi sinir sisteminde bilgi akışı üç ana kısımdan oluşmaktadır: reseptörler, sinir ağı ve efektörler [68].

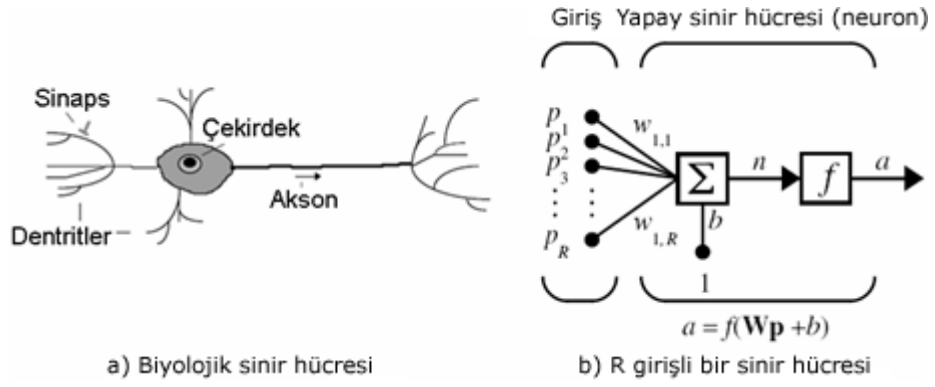


Şekil III. 3 Sinir sisteminde bilgi akışı [54]

Şekil III.3’ 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.1.2. Yapay Sinir Ağı Hücresi

Tek katman (layer) ve R adet girişten oluşan, 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. Neuron (sinir hücresi) ve sinir hücresi girişlerinden oluşan temel yapıya perceptron adı verilmektedir [72].



Şekil III. 4 Biyolojik ve yapay sinir hücresi [72]

Yapay sinir ağı hücresinde temel olarak dış ortamdan ya da diğer nöronlardan alınan veriler yani girişler(**p**), ağırlıklar(**w**), toplama fonksiyonu, aktivasyon fonksiyonu(**f**) ve çıkışlar(**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 ilgili ağırlıkların çarpımının bir sonucudur. Aktivasyon fonksiyonu(**f**) işlem süresince net çıkışını hesaplar ve bu işlem aynı zamanda nöron çıkışını(**a**) verir. Genelde aktivasyon fonksiyonu doğrusal olmayan bir fonksiyondur. Şekil III.4 (b)'de görülen **b**(bias) bir sabittir, bias veya aktivasyon fonksiyonunun eşik değeri olarak adlandırılır. Ağın giriş sayısı **R** ile, aktivasyon fonksiyonu **f** ile gösterilirse tek neuronlu ağın çıkışı Denklem III.1'deki gibi yazılır.

$$a = f(w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b) \quad (III.1)$$

Burada ; **W** = w1, w2, w3, . w_S ağırlık matrisi, (S: neuron sayısı)
p = p1, p2, p3, ..., p_R giriş vektörü, (R: Giriş sayısı)
b = b1, b2, b3, ..., b_S bias vektörüdür.

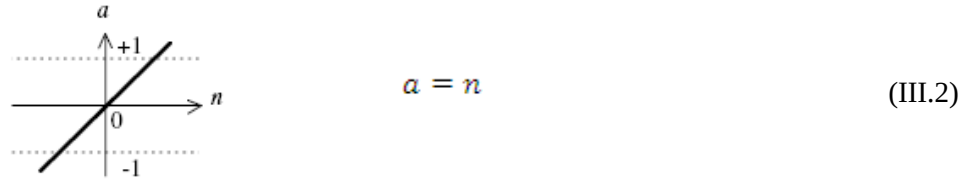
III.1.2.1. Aktivasyon Fonksiyonu

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. Yapay sinir ağının davranışı ağırlıkların ve aktivasyon fonksiyonunun değerine bağlıdır. Hücre modellerinde, hücrenin gerçekleştireceği işleve göre değişik aktivasyon fonksiyonları kullanılabilir. Aktivasyon fonksiyonları sabit / uyarlanabilir parametrelili seçilebilir ve fonksiyon seçimi sinir ağının verilerine ve ağın neyi öğreneceğine 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. Doğrusal olmayan transfer fonksiyonlarının kullanılması, YSA'ların karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır.

1.a. Doğrusal Aktivasyon Fonksiyonu

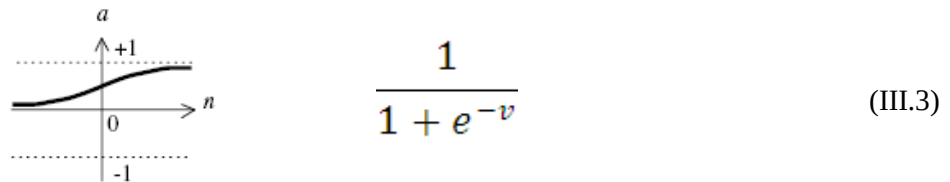
Doğrusal bir problemi çözmek amacıyla kullanılan doğrusal hücre ve YSA'da ya da genellikle katmanlı YSA'nın çıkış katmanında kullanılan doğrusal fonksiyon, hücrenin net girdisini doğrudan hücre çıkışı olarak verir. Doğrusal aktivasyon fonksiyonu Şekil III.5'te ve denklem III.2'de gösterildiği gibi girişe uygulanan değeri çıkış olarak verir. Burada **a** YSA hücresi çıkış, **n** ise YSA hücresi giriş değeri olan $(\mathbf{W} \cdot \mathbf{p} + \mathbf{b})$ 'dir.



Şekil III. 5 Doğrusal aktivasyon fonksiyonu [54]

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 0 ile 1 arasında bir değer üretir. Şekil III.6'da sigmoid aktivasyon fonksiyonun grafiği görülmektedir.

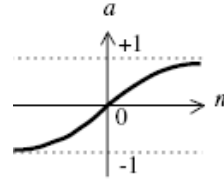


Şekil III. 6 Sigmoid fonksiyon [54]

Denklem III.3'de verilen sigmoid fonksiyonda v sabit bir sayıdır ve eğimini belirler.

1.c. Hiperbolik Tanjant Fonksiyonu

Hiperbolik tanjant fonksiyonu, sigmoid fonksiyonunun biraz farklı şeklidir. Giriş uzayının genişletilmesinde etkili bir aktivasyon fonksiyonudur. Sigmoid fonksiyonun çıkış aralığı 0 ve 1 olurken, hiperbolik tanjant fonksiyonu çıkışı -1/+1 aralığında olmaktadır. Fonksiyon Şekil III.7’de verilmiş, denklem III.4 gösterilmiştir.


$$\frac{1 - e^{-2v}}{1 + e^{2v}} \quad (III.4)$$

Şekil III. 7 Hiperbolik tanjant fonksiyonu [54]

Yukarıda anlatılan aktivasyon fonksiyonlarından başka;

- Basamak Fonksiyonu
- Kutuplamalı Basamak Fonksiyonu
- Parçalı Doğrusal Fonksiyon aktivasyonları kullanılabilir [73].

III.1.3. Yapay Sinir Ağları Modelleri

YSA modellerini sınıflandırarak iki kısımda inceleyebiliriz.

A. Mimari yapılarına göre

- İleri beslemeli (statik)
- Geri beslemeli (dinamik)

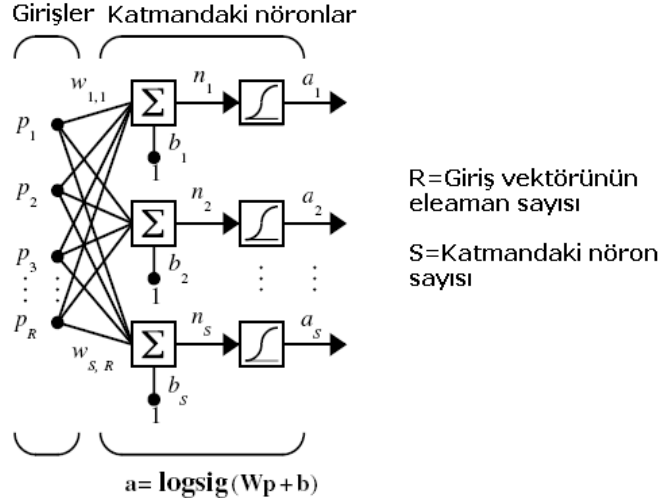
B. Öğrenme yaklaşımlarına göre

- Danışmanlı öğrenme
- Danışmansız öğrenme
- Takviyeli öğrenme

III.1.3.1. Mimari Yapılarına Göre YSA

1.a. İleri Beslemeli (Statik) 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.



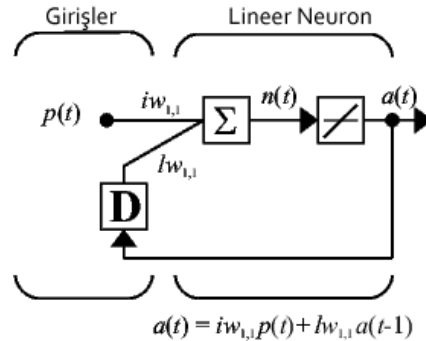
Şekil III. 8 İleri beslemeli YSA modeli [75]

2.b. Geri Beslemeli (Dinamik) 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. [75]

$$\begin{aligned}
 a(0) &= p(0), \\
 a(t) &= iw_{1,1}p(t) + lw_{1,1}a(t-1)
 \end{aligned}
 \tag{III.5}$$

Denklem III.5'deki 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 [75]

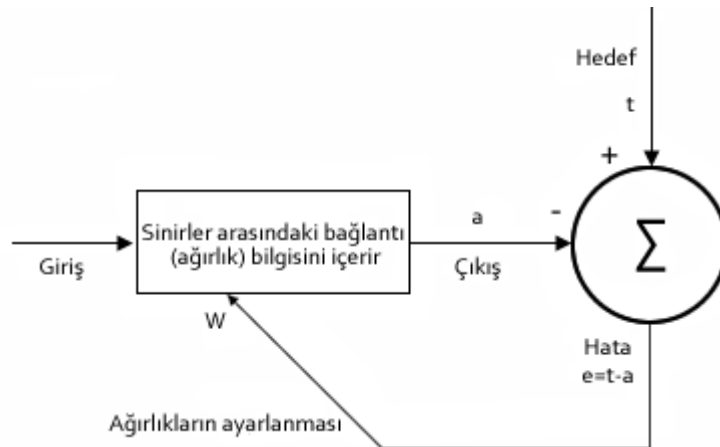
Şekil III.9'da **D(delay)** harfi ile gösterilen zaman gecikmesi, bir örnekleme zamanı önceki bilgiyi hafızada tutar ve bir önceki çıkışı, YSA' nın girişi olarak uygular.

III.1.3.2. 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 ağına da aktivasyon fonksiyonundan geçirilerek bir tepki çıkışı üretilir. 1990'lı yıllardan bugüne gelinceye kadar birçok öğrenme algoritması geliştirilmiştir. Öğrenme algoritmaları eğitici, eğiticişiz ve takviyeli olmak üzere üç grupta toplanmaktadır.

2.a. Eğitici öğrenme

Eğitici öğrenmede, girişler ağı uygulanıp aktivasyon fonksiyonundan geçilerek çıkışlar elde edilir. Bu çıkışlar, tecrübeyle verilen çıkışla karşılaştırılarak hata bulunur. Çeşitli öğrenme algoritmalarıyla hata azaltılıp gerçek çıkışa yaklaşılmaya çalışılır. Bu çalışma süresince yenilenen yapay sinir ağına ağırlıklarıdır. Şekil III.10'da gösterildiği gibi ağırlıklar her bir çevrimde yenilenerek amaca ulaşmaya çalışılır. Amaca ulaşmanın veya yaklaşmanın ölçüsü de yine dışarıdan verilen bir değerdir. Ağırlıkların güncellenmesi süresince eğitici, ödüllendirme-cezalandırma şemasını ağı uygulayarak hatayı azaltır. Bu öğrenme modelinde giriş ve çıkış örnekleri kümesi eğitim kümesi olarak adlandırılır [72].



Şekil III. 10 YSA' nın eğitici öğrenme ile eğitilmesi [54]

Eğer yapay sinir ağı verilen giriş-çıkış çiftleriyle amaca ulaşmış ise ağırlık değerleri saklanır. Ağırlıkların sürekli yenilenip istenilen sonuca ulaşılan kadar geçen zamana öğrenme adı verilir. Yapay sinir ağı öğrendikten sonra daha önce verilmeyen girişler verilip, sinir ağı çıkışıyla gerçek çıkışı yaklaşımları incelenir. Eğer yeni verilen örneklerle de doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir. Sinir ağına verilen örnek sayısı optimum değerden fazla ise sinir ağı işi öğrenmemiş ezberlemiştir. Genelde eldeki örneklerin %80'i ağı verilip ağı eğitilir, daha sonra geri kalan % 20'lik kısım verilerek ağın davranışı incelenir, yani ağı test edilir.

2.b. Eğiticişiz öğrenme

Eğiticişiz öğrenmede, eğiticişli öğrenmedeki gibi arzu edilen y çıkışları bilinmemektedir. Bu yüzden kesin bir hata bilgisini ağıın davranışını değıştirmekte kullanmak mümkün değıildir. Cevabın doğruluğı veya yanlışlığı hakkında bilgi sahibi olunmadığı için öğrenme, girişlerin verdiğı cevaplar gözlenerek başarıya ulaşılır. Eğiticişiz 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 [72].

III.1.4. 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 ağı belirli bir sırada sunulur. Her bir çalışma örneğı çıktı örneğı hesaplanana kadar katman katman ileri yayılır. Hesaplanan çıktı daha sonra hedef çıktı ile karşılaştırılarak hata sinyali bulunur.

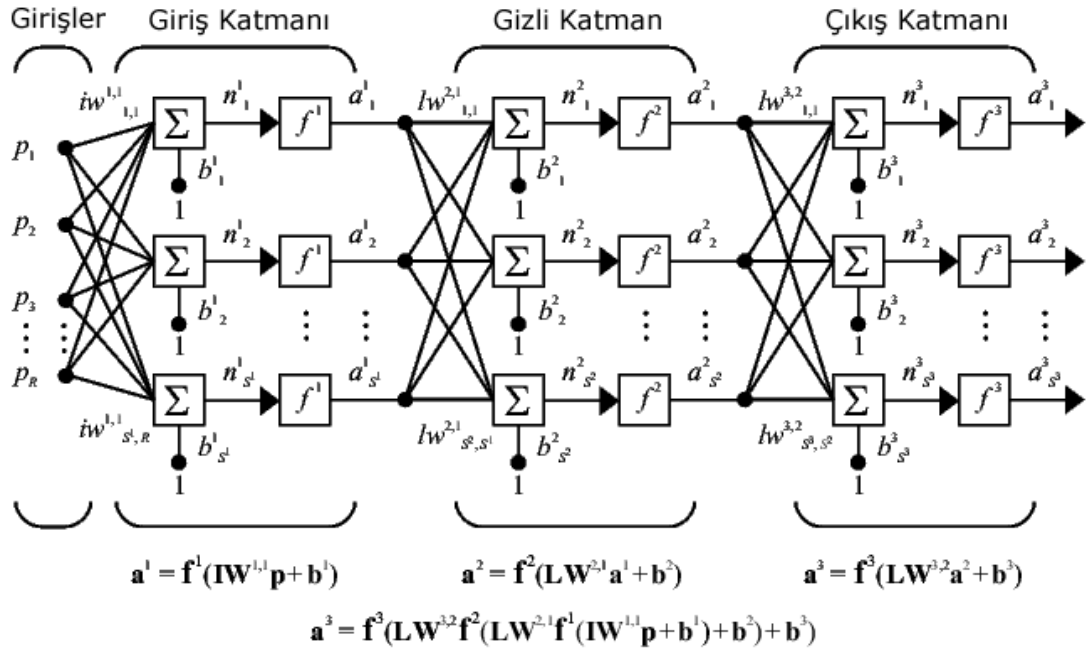
Denklem III.6'da verildiğı gibi hata $e(k)$, arzu edilen çıkış($y(k)$) ile sinir ağıının çıkışı ($a(k)$) arasındaki farktır.

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

Hatalar, katman katman sinaptik ayarlamaların yapıldığı geri besleme bağlantılarına “girdiler” olarak sunulur. Geriye yayılım öğrenme algoritması kullanıldığında, sonraki katmanların hataları kullanılarak gizli katmanın ağırlıkları

ayarlanır. Böylece çıkış katmanında hesaplanan hatalar için önce son gizli katman ile çıkış katmanı arasındaki ağırlıklar ayarlanır. Bu işlem, ilk gizli katmana kadar geriye doğru 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 [74].

Şekil-III.11’de birçok sinir hücresinin bir birine bağlandığı ileri yönlü çok katmanlı bir yapay sinir ağı görülmektedir [75]. 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 [75]

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

1) İleri Yön Değerleri Hesaplanır;

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{LW}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \quad ; m=1,2, \dots, M-1 \quad (\text{III.7})$$

Burada L=layer çıkışı, m=katman sayısıdır.

2) Hatanın bulunması;

$$e = t - a \quad \text{Burada } t \text{ hedef çıkış ve } a \text{ ağı çıkışıdır.}$$

3) Hatanın Geriye Yayılması;

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

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

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

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

4) Ağırlıkların ayarlanması aşağıdaki gibi yapılabilir;

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

Burada α öğrenme oranıdır ve 0 ile 1 arasında bir değerdir. Tüm öğrenme algoritmalarında şu adımlar izlenir. İlk önce denklem III.7 kullanılarak giriş ileri yönde yayılır, sonra denklem III.8 kullanılarak geri yayılır ve son olarak ağırlıklar denklem III.11 kullanılarak güncellenir [72].

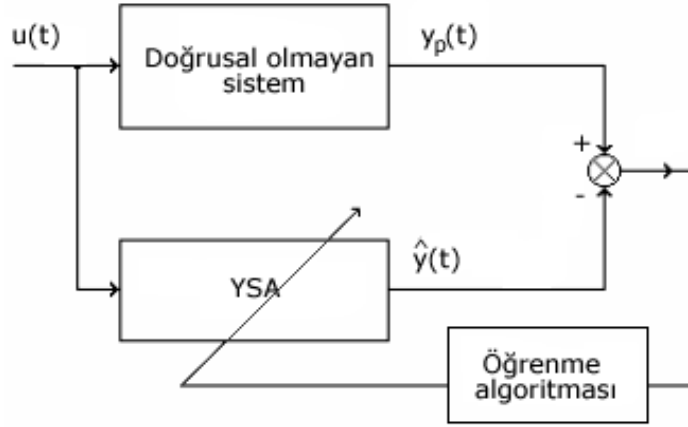
III.1.5. YSA ile Sistem Modelleme

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 bir kaç 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.1.5.1. İleri Modelleme

Zamanla değişen, dinamik bir sistemin giriş ve çıkışı 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.12' de bu metoda ait blok şema görülmektedir.



Şekil III. 12 YSA ile modellemede kullanılan temel ileri tip blok şema [54]

Şekil III.12'de Yapay Sinir Ağı (YSA) bloğu, lineer olmayan sisteme paralel olarak yerleştirilmiştir. Sistem çıkışı y_p ile YSA çıkışı $\hat{y}(t)$ arasındaki hata $e(t) = y_p - \hat{y}$ YSA'nın öğretilmesinde kullanılmaktadır. Çok katmanlı ileri beslemeli YSA modelinde, hatanın geri yayılım algoritması kullanılır [76]. Buna göre YSA'nın sistem dinamiğinin öğrenmesinde eğitici öğrenme prosedürü kullanılmaktadır. Sistemin aşağıda verilen lineer olmayan ayrık zamanlı fark denklemi ile belirtildiği farz edilsin.

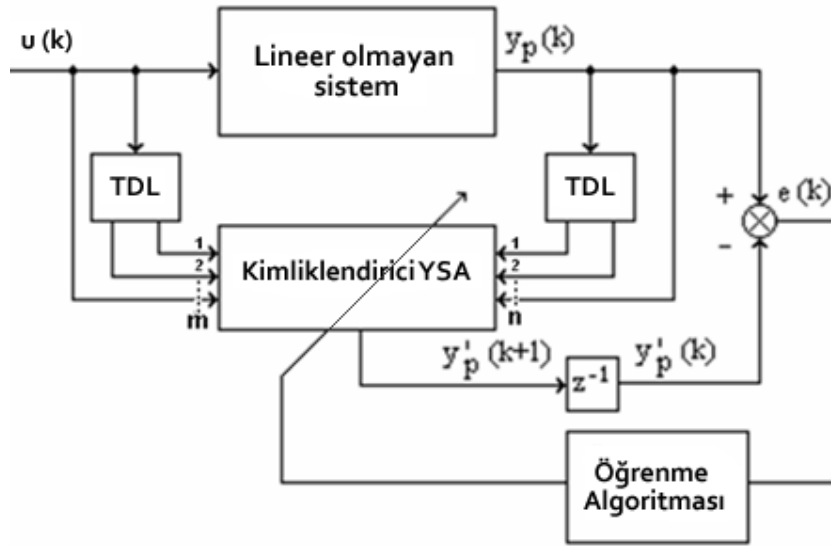
$$y_p(k+1) = F[y_p(k), \dots, y_p(k-n+1); u(k), \dots, u(k-m+1)] \quad (\text{III.12})$$

Denklem III.12'de, $y_p(k+1)$, 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. Bu yüzden, YSA modeli çıkışı şu şekilde belirtilebilir [77].

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

Denklem III.13'de, $N(\cdot)$, YSA'nın lineer olmayan (nonlinear) 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 [77]

Şekil III.13, denklem III.12 ve III.13 nolu bağıntıları kullanarak oluşturulan YSA 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 [77, 78]. Seri – paralel modelleme ile elde edilen yapı daha sonra NARX model yapısına dönüştürülerek sistem sadece giriş bilgisi ile simüle edilen bir model haline getirilir. Bu sayede sisteme herhangi bir anda verilen giriş işareti için sistemden çıkış bilgisi alınır.

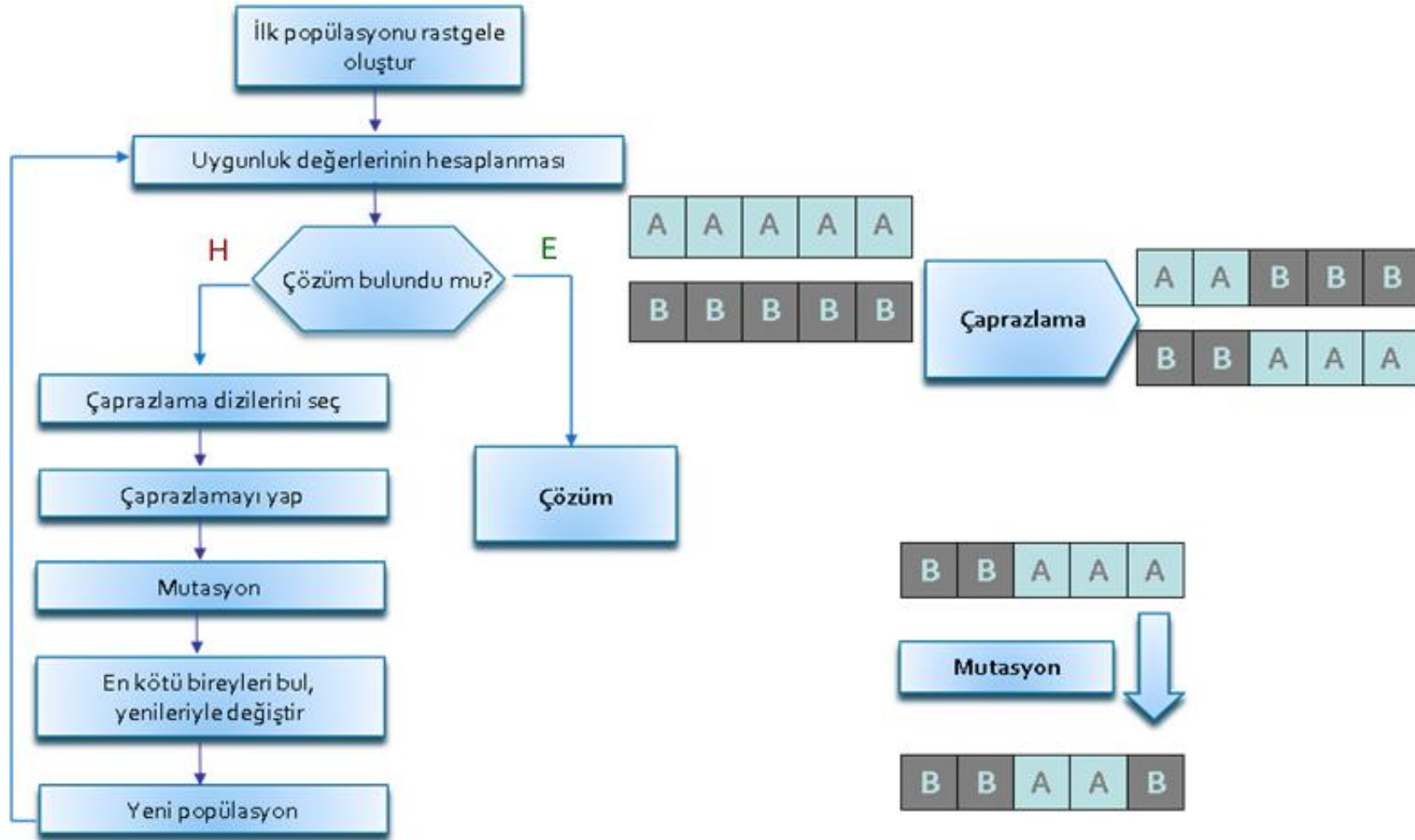
III.2. GENETİK ALGORİTMA

Günümüzün karmaşık ve zor koşulları problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle hard optimizasyon teknikleri yerine, esnek programlama ve evrimsel algoritma kullanımı ön plana çıkmıştır. Genetik algoritma, KKO, bulanık kontrol bu yaklaşımda önemli bir yer tutarlar. Uygulama başarıları artan ve sürekli geliştirilmeye çalışılan GA diğer esnek programlama yöntemleri ile birlikte kullanılarak hibrid çözümler geliştirilmesine çalışılmaktadır.

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 ulaşır [79]. 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 [80].

III.2.1. Basit Genetik Algoritma

Pek çok genetik algoritma uygulamasında basit genetik algoritma (BGA) modeli kullanılır. BGA, D.E.Goldberg tarafından tanımlanmıştır ve diğer ardışık genetik algoritma türleri için referans oluşturan bir algoritmadır. BGA'da ikili kodlama biçimi, rulet tekerleği seçim yöntemi, bütünüyle yer değiştirme üreme yöntemi, tek noktadan çaprazlama ve bit mutasyon operatörleri kullanılmıştır [81]. Genetik algoritma işleyişine ait ait akış diyagramı Şekil III.14'de gösterilmektedir.



Şekil III. 14 Genetik algoritma akış diyagramı [17]

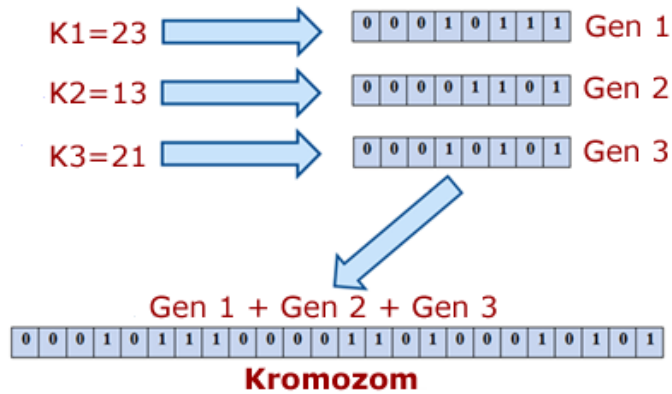
Genel olarak GA, çözüm bilgisinin hiç olmadığı veya çok az olduğu bir durumla aramaya başlar. Çözüm çevreden gelen etkileşime ve genetik operatörlere bağlıdır. GA, aramaya paralel bir şekilde, birbirinden bağımsız noktalardan başlar. Bu nedenle, GA'nın alt optimal çözümlere takılma olasılığı azdır ve 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 özellikler;

- GA, parametre kodlarıyla uğraşır, parametrelerin kendileriyle doğrudan ilgilenmez.
- GA, tek bir alana 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, permutasyon veya ağaç biçimini kullanan genetik algoritma türleri olduğu gibi, farklı sembolik alfabeler kullanan GA çalışmaları da mevcuttur [17].



Şekil III. 15 İkili kodlama [17]

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 [82].

Kodlama biçimi, GA'nın performansını oldukça önemli oranda etkiler, fakat kodlama biçimi probleme bağlı olduğundan bütün problemler için geçerli en uygun kodlama biçimini söylemek mümkün değildir. Z.Michalewicz belli bir problem tipi için yapmış olduğu çalışmada gerçel sayı gösteriminin daha çabuk sonuca ulaştığını göstermiştir [17].

Gerçekleştirilen çalışmamızda bulanık kontrol parametreleri 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 heuristic olarak çeşitli sezgisel yöntemler geliştirilebilir.

III.2.2.3. Popülasyon Büyüklüğü

Popülasyonun büyüklüğü, problemin çözüm süresini etkilemektedir. Fazla sayıda kromozom içeren popülasyon problemin çözüm süresini uzatırken, az sayıdaki popülasyonda 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 aktarılacak bireylerin (kromozomların) 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 nesilde 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 [83, 84].

Bunlar:

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

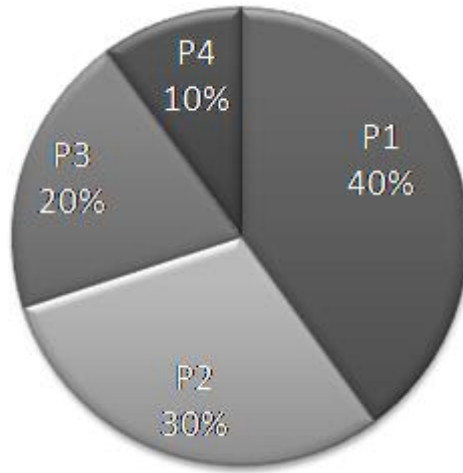
1.a Rulet Tekniği Seçim Yöntemi

Bu seçim yöntemi uygunluk değeri orantılı seçim yöntemine girmektedir. Bu seçim yönteminde, bireyin beklenen seçim olasılığını, bireyin uygunluk değerinin neslin toplam uygunluk değerine bölümü belirler. Yöntem, genel olarak rulet oyununa benzetilir. Nesildeki bireylerin toplam uygunluk değeri rulet tekerleğinin büyüklüğünü verir ve her bir birey rulet tekerleğinde uygunluk değeri kadar yer kaplar [29]. 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 nesile aktarılacak olan bireylerden bir tanesi seçilmiş olur. Benzer şekilde diğer bireylerinde belirlenmesi ile uygunluk değeri en iyi olan bireyler eşleştirme havuzuna (matching pool) alınır. Bu işlem sonra diğer nesle ait bireyler elde edilir ve genetik operatörlerin uygulanmasıyla yeni nesil elde edilir. Aynı işlem her döngüde devam ederek nesil devamı sağlanır [30].

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.

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

Denklem III.14'den elde edilen seçilme olasılığı sırasıyla $p_1=\%40$; $p_2 = \%30$; $p_3 = \%20$; $p_4= \%10$ olarak bulunur. Şekil III.16'da her bir bireyin yüzdesine göre oluşturulmuş çember görülmektedir.



Şekil III. 16 Rulet Tekerleği seçim yöntemi [17]

Seçim işlemi için Rulet Tekerleği kullanılır. Rulet çevrilir, hangi bireyin üzerinde durur ise o birey seçilecektir. Popülasyon içerisindeki 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 [17].

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	$0.7 < r_u \leq 0.9$
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 nesilde 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. Bireyin seçilme olasılığı Denklem III.15' de verilmiştir. [85]

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

Burada; N = popülasyon büyüklüğü, p_i , i. bireyin seçilme olasılığı, q=turnuva büyüklüğü olarak verilir. i ise kaçınıcı bireyi seçeceğimizi gösteren indistir.

1.c Sıralı Seçim Yöntemi

Her kromozom uyumluluk fonksiyonuyla hesaplanmış uyum değerine göre sıralanır. Daha sonra kromozomlar en kötü uyum değerine sahip olandan en iyi değere sahip olana doğru sıralanır. En uyumsuz kromozomun değeri 0, en uyumlu kromozomun değeri is N (toplam kromozom sayısı) kadardır. Böylece kromozomların seçilme olasılığı, doğrusallaştırılmış artan fonksiyon haline getirilir. Bu çeşitlilik yaratmak açısından Rulet Çemberinden daha iyi sonuç veren bir seçim yöntemidir. Uyumluluk değerinin büyüklüğüne göre kromozomların seçilme ihtimali uyumluluk değeri ağırlıklarına göre olacaktır.

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.

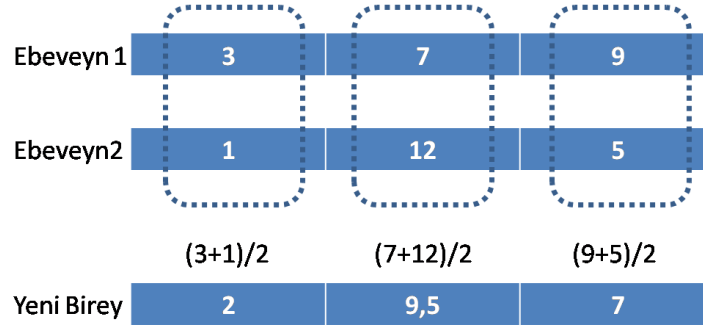
İkili kodlamada genel olarak çaprazlama noktası bir tanedir. Ancak çaprazlama noktası birden fazla seçilebilmektedir. Şekil-III.17'de çaprazlama operatörünün bir noktada ve iki noktada uygulanarak oluşturulan yeni nesiller görülmektedir. Çaprazlama olasılığı (P_c) ise bir popülasyon içindeki ebeveynlerin yeni bireyler

üretmek için hangi olasılıkla çaprazlanacağını belirtir. Örnek olarak ($P_c=1$) alınırsa, bütün ebeveynlerin çaprazlanacağını gösterir.



Şekil III. 17 Çaprazlama operatörü [17]

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

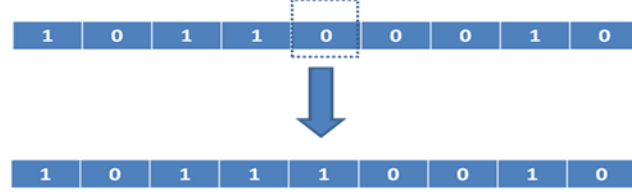


Şekil III. 18 Aritmetik çaprazlama [54]

III.2.2.7. Mutasyon(Değişim)

GA’da sistem, belirli bir döngü değerine geldikten sonra bireyler birbirlerine gitgide benzemektedir. Bu da 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 1 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 [54]

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, 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 değerine göre genin değeri azalabilir veya artabilir

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 [17]. 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 algorithmada 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 [84]. 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 arttıracakını belirtmiştir.

Burada kullanılacak ceza fonksiyonu denklem III.16 ve III.17'deki gibi tanımlanır;

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

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

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, süresiz ve gürültü içeren fonksiyonları çözmeye 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 [17].

Kontrol uygulamalarında GA'nın kullanımının önemli bir bölümünü ise denetleyici parametrelerinin oluşturduğu geniş arama uzayından, optimum denetleyici parametrelerinin bulunmasına yönelik yapılan çalışmalar oluşturur [17]. Gerçekleştirilen çalışmada, sistemi verilen referansı optimum şekilde kontrol eden Bulanık denetleyicinin katsayıları GA ile bulunmuştur.

III.3. KARINCA KOLONİ OPTİMİZASYONU ALGORİTMASI (KKO)

KKO algoritmaları model tabanlı arama algoritmaları sınıfındandır. Bu sınıf ayrık optimizasyon problemlerini çözmede gittikçe artan popüleriteye sahip algoritma sınıfıdır. KKO teknikleri, optimizasyon problemlerini çözmek için karıncaların sosyal davranışını modeller. Karınca algoritmaları, gerçek karıncaların davranışının gözlenmesinden türetilen modelleri inceler [79]. Bazı problemleri çözmenin tek yolu cevapla ilgili bütün ihtimallerin tek tek test edilmesi ile gerçekleşir. Buna verilebilecek klasik bir örnek, şehir şehir dolaşan bir işadınının 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 Enstitüsü'nde 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 [86]. 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 [87].

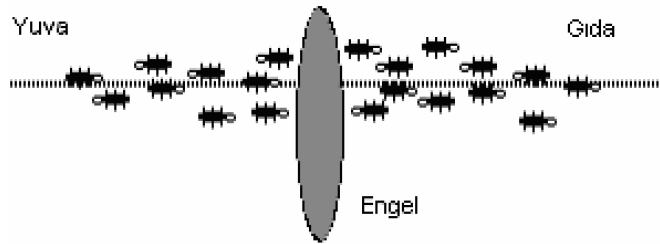
III.3.1. Gerçek Karıncalar

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.



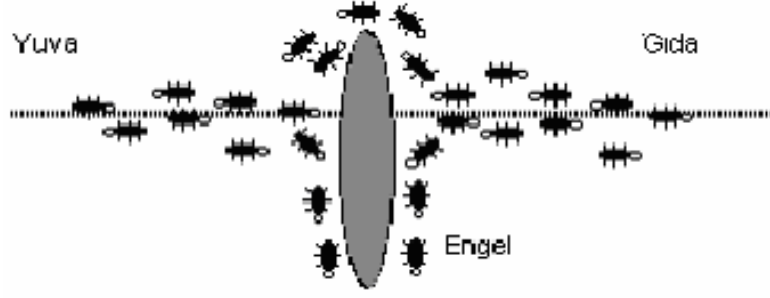
Şekil III. 20 Karıncaların izlediği yol [79]

Karıncaların bu yolu bulmak için kullandıkları araç feromonlar yani bıraktıkları kimyasal salgılardır. 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.



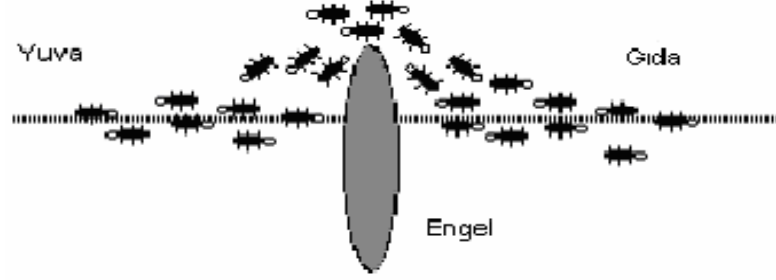
Şekil III. 21 Karıncaların bir engelle karşılaşması [79]

Bu konumda, karıncanın yeni yön seçeneklerinin seçilme olasılıkları eşittir. Yani karınca sağ ve sol yönlerinden birisini seçebiliyorsa, bu yönlerin seçilme ihtimalleri 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 [79]

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



Şekil III. 23 Karıncaların kısası bulmaları [79]

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.

III.3.2. Yapay Karıncalar

Gerçek karıncalar, 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 ayrı 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 algoritmalarda temel fikir, basit iletişim mekanizmalarını kullanan yapay akıllı araçların, birçok karmaşık problem için çözümler üretebilmesidir [79].

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 feromen bırakırlar. Bu feromenler, 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, feromen ve seçilen yol verilerini tutan global veri yapısını aynı anda değiştirememelidir. Bu uygulamada temel amaç, KKO 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.

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

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

Burada;

- $\tau_{ij}(t)$: i ve j numaralı noktalar arasında t. tur içindeki feromen miktarını,
- η_{ij} : iki nokta arasındaki uzaklığı,
- α : problemde feromen ize verilen bağıl önemi,
- β : problemde görünürlük değerine verilen önemi,
- j : henüz seçilmemiş olan düğüm noktaları kümesini gösterir.

Karıncalar bu olasılık bağıntısına göre bir sonraki seçimlerini yaparlar. Problemdeki tüm düğüm noktaları gezildikten sonra bir tur veya iterasyon tamamlanmıştır. Bu noktada denklem III.19'a göre feromen miktarı güncellenir [88].

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

Burada;

ρ : t ile t+n süreleri arasında iz buharlaşma katsayısı ($0 < \rho < 1$)

$\Delta\tau_{ij}$: Karıncanın bir turu boyunca (i,j) bağlantısını seçmelerinden dolayı bu kenarda oluşan iz miktarını gösterir. Oluşan iz denklem III.20'ye göre hesaplanır.

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

Burada;

m : Toplam karınca sayısı

T_{ij} : k. karıncanın (i,j) köşesine bıraktığı feromen iz miktarı

Denklem III.20 ise her bir (k) karıncasının herhangi bir (i,j) köşesindeki feromen iz miktarına ne kadarlık katkı yapacağını gösterir [89].

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

Burada;

Q : Sabit bir değerdir

L_k : k. karıncanın tur uzunluğu

Eğer karınca tur boyunca (i,j) köşesini kullandı ise denklem III.21'e göre bıraktığı iz miktarı hesaplanır. Aksi halde iz miktarı sıfır olur [90].

III.3.3.1. Feromen Buharlaşması

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

III.3. BULANIK MANTIK DENETLEYİCİ

Aristotle'nin doğru (1) ve yanlış (0) gibi iki seviyeli mantığından sonra Lukasiewicz, doğru(1), yanlış (0) ve nötr (1/2) olmak üzere 3 seviyeli bir mantık önermiştir. Burada nötr yarı doğruyu veya yarı yanlışı temsil etmektedir. Zadeh 1965 yılında yayınladığı bir makalesi ile bulanık mantığın sahip olabileceği doğruluk değerlerinin sayısına bakmaksızın herhangi bir mantığın temeli olduğunu göstermiştir. Bir bulanık küme esnek sınırlara sahiptir. Şöyle ki böyle kümelerin elemanları sadece siyah ve beyaz renklerini temsil etmez aynı zamanda ikisi arasındaki gri rengin bir spektrumunu da temsil eder. Güncel olarak bulanık mantık uygulamalarının en aktif sahalarından biri kontrol sistemleridir. Bulanık kontrol sistemleri sistemden gelen işaretlerin etkilerini ayarlayacak bir kontrol karar mekanizmasını temsil eden bulanık kurallar takımını içeren kural tabanlı sistemlerdir. Bulanık kontrol sistemlerinin amacı becerikli bir operatörün yerine bulanık kural tabanlı bir sistemi koymaktır. [91]

Günümüzde bulanık mantığın uygulandığı pek çok alan mevcuttur. Kontrolü yapılan sistemlerin ortak özelliği, sistemlerin doğrusal olmaması veya sistemin matematiksel modelinin çok karmaşık veya hiç elde edilememesidir [17].

Bulanık mantık, bulanık üyelik kavramı kullanarak insanın düşünme biçimi modellemeye çalışır ve dilsel değişkenler kullanır. Bulanık denetleyici, özellikle geleneksel sayısal tekniklerle analizi çok karmaşık olan sistemlerde ya da elde edilebilir bilgi kaynakları sayısal olmayan veya kesin olmayan sistemlerde çok etkindir [92].

III.3.1 Bulanık Denetleyici Tasarımda Karşılaşılan Problemler

Bulanık kontrol, pek çok karmaşık endüstriyel işlemde başarıyla kullanılmasına rağmen, bulanık denetleyicilerin performansı pekçok durumda objektif olarak elde edilemeyen, sezgisel veya deneysel bilgilere ihtiyaç duyar. Bulanık tasarımda çözülmesi gereken temel problemler ;

- Durum uzayının belirlenmesi,
- Kontrol kurallarının üretilmesi,
- Üyelik fonksiyonlarının tiplerinin ve sınırlarının belirlenmesidir.

Geleneksel metodlarda, yukarıda tanımlanan sorunları çözmek için deney ve tecrübeye ihtiyaç vardır. Bulanık denetleyici tasarımda geleneksel yaklaşım, zaman harcamak ve uzun işgücü harcamayı gerektirir. Bu yaklaşımın diğer sakıncalarını söyledir ;

- Operatörler genellikle, kendi bilgi ve deneyimlerini bir otomatik kontrol stratejisine dönüştürmek için gereken kural-tabanlı biçime veya bir algoritmik ifadeye kolayca dönüştüremezler.

- Bu uzman bilgisi her zaman elde edilmeyebilir.
- Çok değişkenli bulanık denetleyici tasarlamak oldukça zordur.

Optimizasyon algoritmaları ile yukarıda tanımlanan temel problemleri çözmek için Makine Öğrenmesi (Machine Learning) kavramından yararlanılır.

III.3.1.1 Makina Öğrenmesi (Machine Learning)

Makina öğrenmesi, öğrenme yeteneğine sahip bilgisayar programları olarak tanımlanır. Bu bilgisayar programları giriş bilgilerini kullanarak yeni bir bilginin yaratılması veya mevcut olan bilgiyi kullanarak geliştirebilme yeteneğine sahiptir. Bu yöndeki çalışmalardan çoğu, algoritmik yapılardan ziyade sezgisel yapılar kullanırlar. Optimizasyon algoritmaları ile makine öğrenmesinde 2 temel yaklaşım vardır. Birinci yaklaşım, Holland'ın bir sınıflandırma problemi için geliştirdiği Michigan Yaklaşımı (Michigan Approach), diğeri DeJong ve talebelerinin Pittsburgh üniversitesinde geliştirdikleri Pittsburgh Yaklaşımı (Pitts Approach) metodudur.

III.3.1.1.2 Michigan Yaklaşımı

Bu yaklaşım, bir sınıflandıma probleminin kural tabanlı sistem ile çözümü için geliştirilmiştir. Kural tabanlı sistemlerin genel özellikleri, kuralların paralel olarak işlenmesine, yeni kuralların üretilmesi ile uyarlanmasına ve var olan kuralların etkinliğinin test edilmesine dayanır. Sınıflandırıcı sistemlerde ise, bir popülasyon içerisinde kurallar bit dizisi şeklinde kodlanır ve çevreden gelen teşvik ve kuvvetlendirme sonucunda değerlendirilir. Sistem, uygun bir teşvik olduğu sürece öğrenmeye devam eder. Bir sınıflandırıcı sistemde kurallar, zaman içerisinde gelişen bireylerin bir popülasyonundan oluşur. Bu yaklaşım ile bulanık denetleyici tasarlamak için, bulanık denetleyici parametrelerinin ayrı bir kromozom içinde kodlanması gereklidir. Bulanık denetleyicide, denetleyici performansı sistemin toplam performansı olarak tarif edildiğinden ve bulanık denetleyiciye ait her bir parametre birbiri ile ilişkili olduğundan dolayı, bu yaklaşım bulanık denetleyici tasarımında tercih edilmez.

III.3.1.1.3 Pittsburg Yaklaşımı

Pittsburg yaklaşımı, popülasyondaki her bir bireyin, bütün kuralların kümesini temsil etmesini benimser. Bu şekilde oluşturulan bireyler kendi aralarında rekabet ederler ve zayıf bireyler ölürken, güçlü bireyler yaşar ve yeniden üretilir. Bu işlem, yeniden üretim, çaprazlama, mutasyon ve uygunluk değeri ile orantılı seçim yöntemleri ile yapılır. Böylece, Pittsburg yaklaşımı genetik algoritmayı öğrenme problemine uygular. Bu nedenden dolayı Pittsburg yaklaşımı genetik algoritma ile bulanık denetleyici tasarımında oldukça sıklıkla kullanılır. [17]

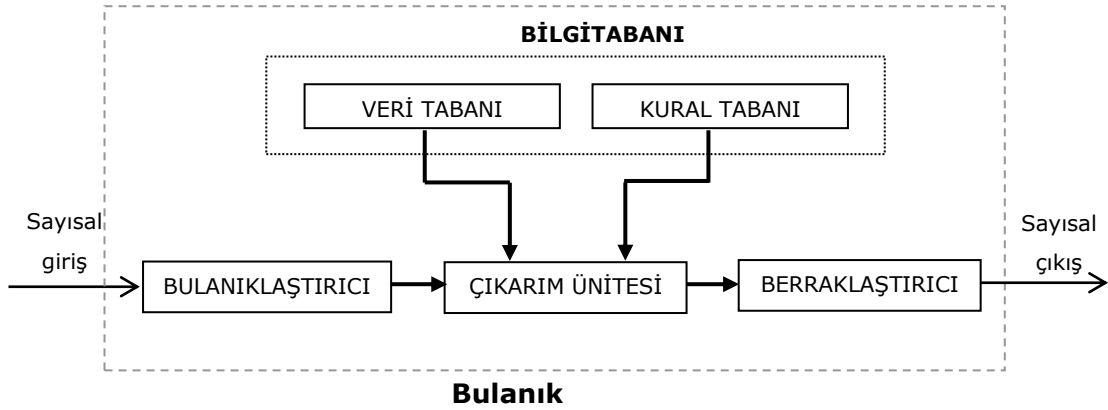
III.3.2 Bulanık Kümeler

Bir bulanık kümede, kümedeki bir elemanın kümeye aitlik derecesi (μ), 0 ile 1 arasında verilir [23]. Böylece bir elemanın kısmi bir üyeliğinden söz edebiliriz. Klasik kümelerde bir eleman ya o kümenin bir elemanıdır ya da o kümenin bir elemanı değildir. Bulanık kümede ise bir eleman bir kümeye aitlik derecesi kadar o kümenin elemanı olacaktır. Eğer aitlik derecesi 1 ise, bu eleman bulunduğu kümeye tam üyedir, aitlik derecesi 0 ise, bulunduğu kümeye üye değildir; ancak aitlik derecesi bu iki değer arasında olursa o elemanı için yüzdesel veya oransal bir üyelikten bahsedebiliriz [65].

III.3.3 Bulanık Denetleyicinin Temel Yapısı

Bulanık kontrol sistemleri, kural tabanlı (rule-based) sistemlerdir. Bu kontrol sistemlerinin amacı, sistemi kontrol eden uzman kişinin yerine bulanık kural tabanlı bir sistem yerleştirmektir.

Şekil-III.24’de bu dört birimden oluşmuş bir bulanık mantık denetleyicinin yapısı görülmektedir [24].



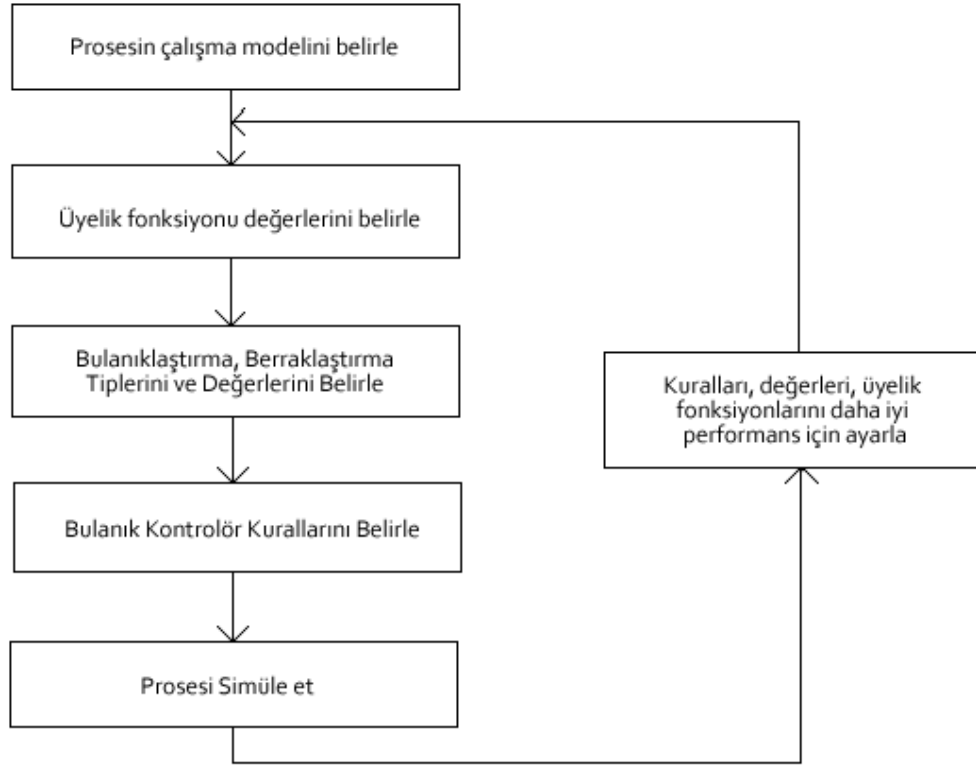
Şekil III. 24 Bulanık denetleyicinin genel yapısı [24]

Bulanık denetleyici, algılayıcılardan gelen bilgileri, bulanık üyelik fonksiyonlarını kullanarak dilsel değişkenlere çevirir. Bu işleme bulanıklaştırma denir. Bu işlemden sonra kural tabanı ve veri tabanı içeren çıkarım ünitesinde, bulanık “Eğer....Öyle ise...” (If – Then) kurallarına göre değerlendirilerek mantıksal çıkarımlar, bulanık çıkışlar elde edilir. Elde edilen bulanık çıkış değerleri, berraklaştırma yöntemlerinden biri kullanılarak kesin bir sayısal değere çevrilir. Elde edilen bu sayısal değer kontrol işaretini oluşturur ve sisteme uygulanır [25].

Bulanık denetleyici genel olarak 4 ana bölümden oluşmaktadır.

- 1- Bulanıklaştırma birimi
- 2- Bulanık kural ve veri tabanı
- 3- Çıkarım ünitesi
- 4- Berraklaştırma birimi

Bulanık denetleyici tasarım işlemi genellikle bir deneme yanılma döngüsünden oluşur ve bu işlem için gerekli adımlar Şekil-III.25’de verilmiştir [65].



Şekil III. 25 Bulanık denetleyici işlem adımları [17]

Yukarıdaki verilen işlem adımlarını maddeleştirirsek, bulanık denetleyici tasarlarken aşağıdaki maddelere dikkat edilmelidir.

- a- Kontrol edilecek sistemin çalışma modeli belirlenir.
- b- Bulanık üyelik tipleri ve sınır değerleri belirlenir.
- c- Üyelik fonksiyonları belirlenir.
- d- Uzman kişi bilgisine göre bulanık kurallar oluşturulur.

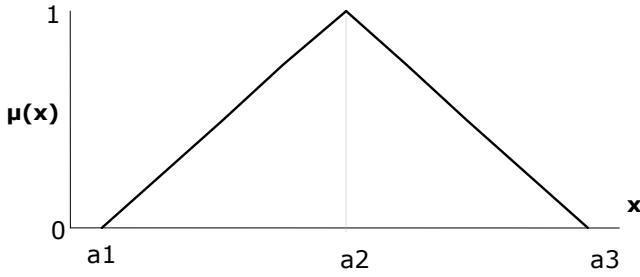
Bulanık denetleyici tasarımı, kabul edilebilir bir performansı elde etmek için sonuçları ardışık değişikliği ve sistem davranışı ve sistem davranışını analize dayanan sezgisel deneme ve yanılma yaklaşımını gerektirir.

III.3.3.1 Bulanıklaştırma

Algılayıcılardan gelen sayısal bilgiyi, dilsel değişkenlere çevrildiği birimdir [25]. Kesin olmayan ölçümleri ifade eden her giriş için, bulanıklaştırma işlemi tanımlanır [65]. Bulanıklaştırma küme içerisindeki tanımlanan dilsel ifadeler ile ölçülen bulanık olmayan girişlerini haritalama işlemidir ve haritalanmış veriyi uygun dilsel terimleri üyelik değerlerine çevirir [15]. Üyelik değerleri, sistemin bulanık değişkenlerini ifade etmek üzere kullanılan ve genellikle biçimsel tanımlamalardır. Üyelik

fonksiyonlarının şekli üçgen, trapez veya n. dereceden fonksiyonlar olabilir. Kullanılacak üyelik fonksiyonu ve sayısı, tasarımcının kontrol edeceği sistem ile ilgili isteklerine ve tecrübesine bağlıdır. Kullanılabilecek örnek üyelik fonksiyonları Şekil-III.26’da ilgili üyelik fonksiyon deklemleri ise şekillerin yanındaki denklemler ile verilmiştir.

a) Üçgen üyelik fonksiyonu

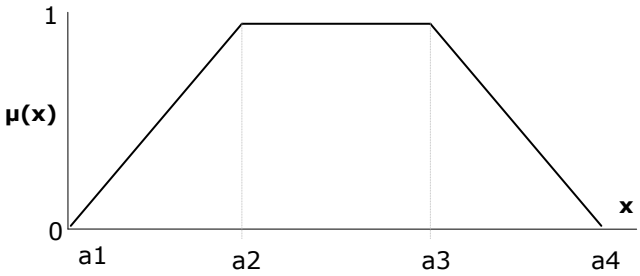


$$\mu(x) = 0, x < a_1$$

$$\mu(x) = \frac{x - a_1}{a_2 - a_1}, a_1 \leq x \leq a_2$$

$$\mu(x) = \frac{a_3 - x}{a_3 - a_2}, a_2 \leq x \leq a_3$$

b) Trapez üyelik fonksiyonu



$$\mu(x) = 0, x < a_1$$

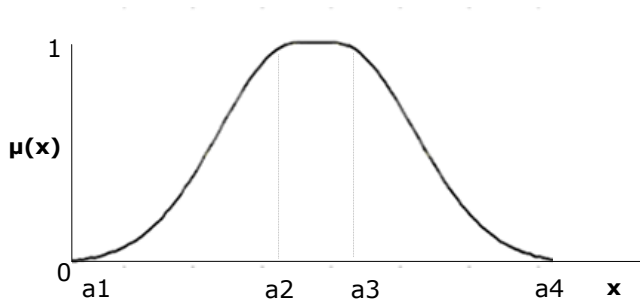
$$\mu(x) = \frac{x - a_1}{a_2 - a_1}, a_1 \leq x \leq a_2$$

$$\mu(x) = 1, a_2 \leq x \leq a_3$$

$$\mu(x) = \frac{a_4 - x}{a_4 - a_3}, a_3 \leq x \leq a_4$$

$$\mu(x) = 0, a_4 > 0$$

c) 2. Dereceden üyelik fonksiyonu



$$\mu(x) = 0, x \leq a_1 \text{ ve } a_4 \leq x$$

$$\mu(x) = \frac{-(x - a_1)^2}{(a_2 - a_1)^2} + \frac{2(x - a_1)}{(a_2 - a_1)},$$

$$a_1 \leq x \leq a_2$$

$$\mu(x) = 1, a_2 \leq x \leq a_3$$

$$\mu(x) = \frac{-(x - a_4)^2}{(a_3 - a_4)^2} + \frac{2(x - a_4)}{(a_3 - a_4)},$$

$$a_3 \leq x \leq a_4$$

Şekil III. 26 Çeşitli üyelik fonksiyonları ve üyelik fonksiyon denklemleri

III.3.3.2 Kural Tabanı

Kural tabanı, bulanık denetleyici davranışı belirleyen kontrol kurallarını içerir. Uzman kişinin bilgi ve tecrübelerinden yararlanılarak oluşturulmuş, bulanık denetleyicinin davranışları belirleyen kontrol kurallarıdır [25]. Kural tabanı *eğer... ise...* (if... then) biçiminde oluşturulmuş dilsel tanımlamalardır. Kurallardaki “*eğer*” terimi giriş değişkenlerinin bulanık ifadeleri, “*ise*” terimi her kurula ait kontrol davranışının sayısal değerlerini içermektedir. Örnek bir kural aşağıdaki gibi gösterilebilir.

Eğer Hata = sıfır ve hata değişimi = sıfır ise kontrol sinyali = 0.

Bu ifadede robot kolunun bulunduğu konumda istenilen referansa göre hata ve hata değişimi bulanıklaştırılacak giriş değişkeni, kontrol sinyali ifadesi ise kurala ait kontrol davranışdır.

Kural tabanının oluşturulmasında kullanılan beş değişik yöntem aşağıda maddeler halinde sıralanmıştır [25].

1. Uzman kişinin bilgi ve tecrübesine göre bulanık model oluşturulur.
2. Sistem çalıştırılarak davranışları gözlenir. Gözlenen davranışlara göre bulanık model oluşturulur.
3. Sistemin matematik modelinden hareketle bulanık model oluşturulur.
4. Kurallar denetleyici tarafından öğretilir.
5. Kurallar genetik algoritma gibi bir optimizasyon algoritması kullanılarak üretilir.

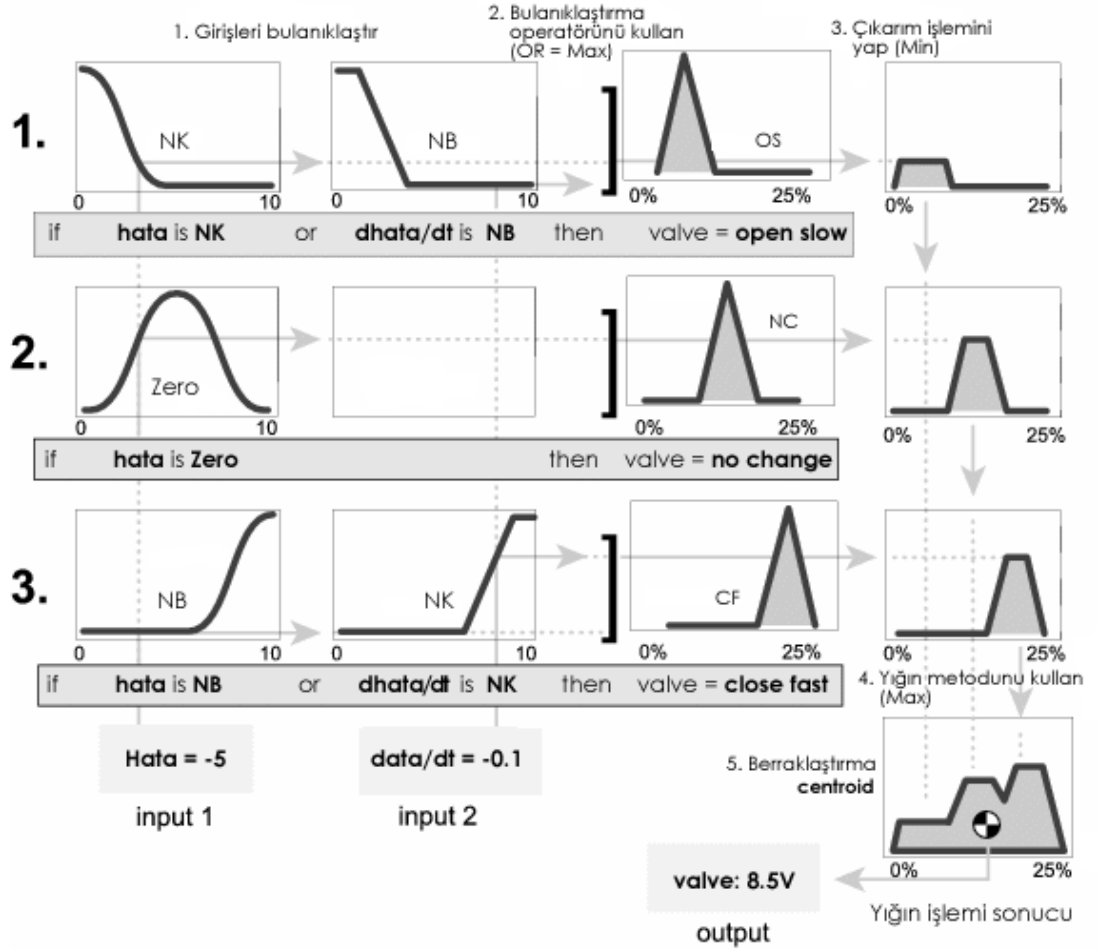
III.3.3.3 Çıkarım Ünitesi

Bu ünitenin amacı, giriş değişkenlerini, ilgili bulanık kurallar ile birleştirerek bulanık kontrol işaretinin bulunmasıdır. Farklı çıkarım modelleri bulunmaktadır, fakat günümüzde birçok uygulamada Mamdani Bulanık Modeli ile Sugano Bulanık Modeli kullanılmaktadır.

a) Mamdani Bulanık Modeli

İlk oluşturulan bulanık çıkarım modelidir. Mamdani bulanık modelinde çıkarım işlemi, A, B ve C bulanık üyelik değerleri olmak üzere;

“Eğer $x=AI$ ve $y=BI$ ise $z=CI$ ” formunda yazılabilir. Bu modele göre oluşturulan ve minimum (ve), maksimum (veya) çıkarım operatörlerinin kullanıldığı ile bir çıkarım ünitesi Şekil-III.10 da görülmektedir [65].



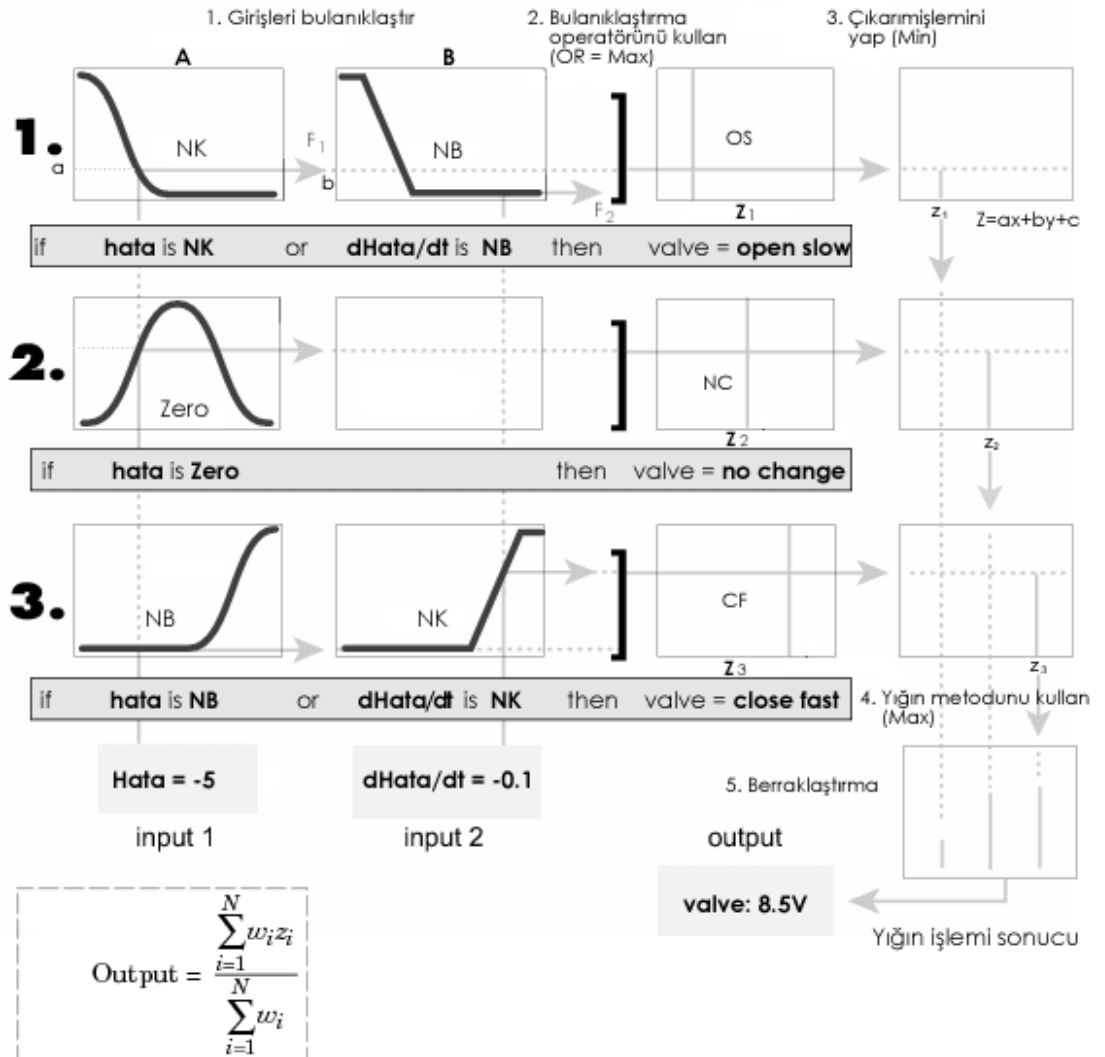
Şekil III. 27 Mamdani Bulanık modeli [93]

b) Sugano Bulanık Modeli

Bu model, Sugano tarafından giriş çıkış veri bilgilerine dayanan bir sistemde bulanık denetleyici tasarlamak amacıyla geliştirilmiştir. Bu modelde çıkarım işlemi, A, B bulanık üyelik değerleri, Z gerçel değer ve $f()$ x,y arasında bir fonksiyon olmak üzere; “If Input 1 = x and Input 2 = y, then Output is $z = ax + by + c$ ” formunda bir lineer denklem olarak yazılabileceği gibi sabit katsayılar olarak da atanabilir.

Burada a ve b sırasıyla A ve B bulanık kümelerinin ağırlık değerleridir. Bu modele göre oluşturulan maksimum (veya) çıkarım operatörlerinin kullanıldığı ile bir çıkarım ünitesi Şekil-III.11 de görülmektedir. Burada $W_i = AndMethod (F_1(x), F_2(y))$

ile elde edilen ve her bir kurala ait ağırlık değerleridir. Dolayısıyla W_i ile elde edilen her bir kurala ilişkin girişlerden gelen minimum ağırlık değeridir.



Şekil III. 28 Sugano Bulanık Modeli [93]

III.3.3.4 Berraklaştırma Birimi

Mamdani bulanık modelinde elde edilmiş bulanık kontrol sinyalinin, kontrol edilen sisteme uygulayabilmek için, sayısal değere dönüştürülmesi gerekmektedir. Berraklaştırma işlemi için yaygın olarak kullanılan üç metot vardır. Bunlar:

a) *Maksimum Berraklaştırma Metodu:*

Aktif olan kuralların en büyük üyelik derecesi, sayısal kontrol işareti olarak alınır.

b) *Maksimum Ortalama Metodu:*

Aktif olan kurallardan elde edilen maksimum kontrol işaretinin ortalaması, kontrol işareti olarak alınır. Fonksiyonel aksiyon değerleri değil sabit değerlerin kullanıldığı seçimlerinde kullanılır.

$$u_k = \sum_{i=1}^n \frac{w_i}{n} \quad (\text{III.22})$$

Denklem III.22’de w_i , $\mu(w_i)$ üyelik fonksiyonunun maksimum seviyeye ulaştığı değeri, u_k , sayısal kontrol işaretini ve n ise aktif kural sayısını göstermektedir.

c) *Alan Merkezi Metodu:*

Bu metod alan ağırlık merkezi yöntemi olarak da bilinmektedir. Uygulamalarda en yaygın olarak kullanılan metottur. Aktif kuralların bulanık çıkışlarına ilişkin üyelik fonksiyonu değerleri ile gerçekleştirme oranı çarpılarak toplanır. Elde edilen değer, üyelik fonksiyonu değerinin toplamına bölünmesi ile sayısal kontrol işareti bulunur. Ağırlık merkezi yöntemi matematiksel ifadesi denklem III.23’deki gibidir [93].

$$u_k = \frac{\sum_{i=1}^n w_i \cdot \mu(w_i)}{\sum_{i=1}^n \mu(w_i)} \quad (\text{III.23})$$

Denklem III.23’de $\mu(w_i)$ üyelik fonksiyonunu, n kural sayısını göstermektedir. Bu yöntem çoğunlukla 1. derece fonksiyonların kontrol işaretini elde etmek için kullanılır.

BÖLÜM IV

GuntRT 512 SİSTEMİNİN YAPAY SİNİR AĞLARI İLE MODELLENMESİ

IV.1. GİRİŞ

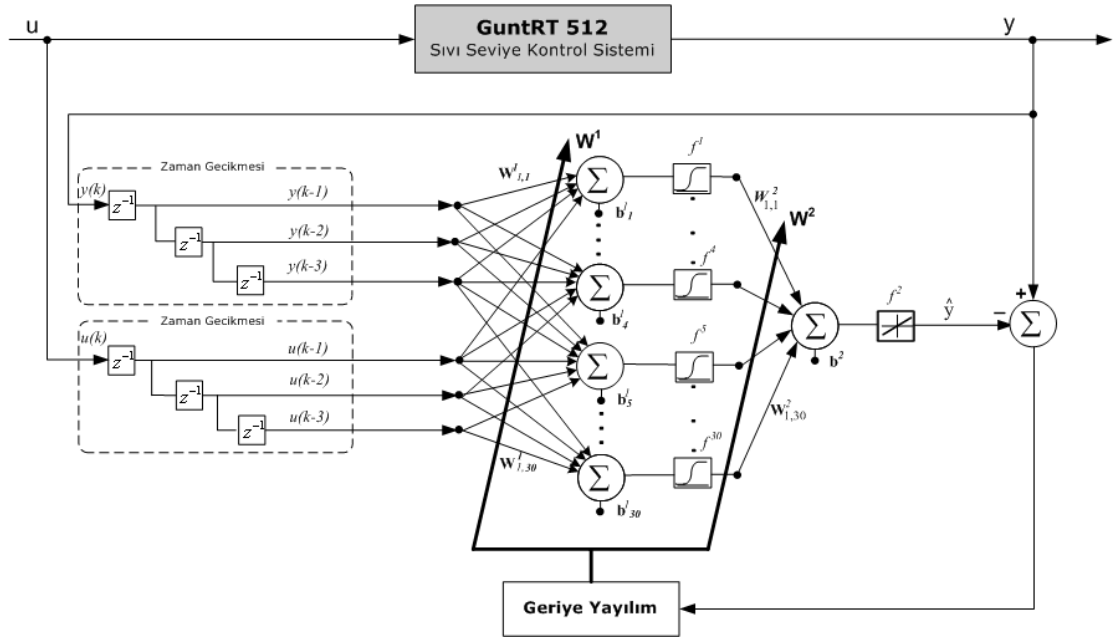
Bu bölümde, GuntRT 512 su seviye kontrol sisteminin YSA ile modellenmesi gerçekleştirilmiştir. Kontrol sistemlerinin davranışını sergileyecek bir model oluşturma ihtiyacı, özellikle tekrar sayısının çok olduğu optimizasyon problemlerinde ortaya çıkar. Bu amaçla GuntRT 512 su seviye kontrol sistemini modellenmesi ve elde edilen model üzerinde Genetik, KKO algoritmalarının ve denetleyicinin davranışının simülasyon olarak izlenmesi zaman ve maliyet açısından avantaj sağlayacaktır.

IV.2. SİSTEMİN YSA İLE MODELLENMESİ

Modelleme işleminde sisteme verilen kontrol işaretine karşılık sıvı seviye tüpünde görülen seviye değişimi ve bu seviye değişimiyle ortaya çıkan basınç değişim davranışı, zaman serisi modellemeye dayanan seri-paralel YSA ile iki farklı şekilde modellenmiştir. Birinci model için tahliye vanası %100 açıklıkta tutulmuş, ikinci model için ise tahliye vanası %70 kapatılarak sistem modellenmiştir. İlk modelimiz simülasyon sırasında sistem davranışını gösterirken, ikinci model sisteme bozucu ithal etmek için kullanılmıştır. Bu sayede gerçek zamanlı çalışmada sistemin bozucuya gösterdiği davranış simülasyon üzerinde de görülebilmektedir. Sistemin YSA ile modeli çıkartılmadan önce matematik modeli üzerinden lineer kabul edilen bölge üzerinden ilgili deneyler yapılmış ancak gerçek sistem davranışı ile örtüşmediği görülmüştür. Doğrusal davranmayan sistem için elde edilen doğrusal model ile yapılan deneyler sonucunda transfer fonksiyonunun sistemin simülasyonu için uygun görülmemiş sistemin verilen giriş işaretleri için üretilen çıkış işaretleri ile birlikte yapay sinir ağı ile modellenmesine karar verilmiştir.

IV.2.1. Kullanılan YSA Modeli

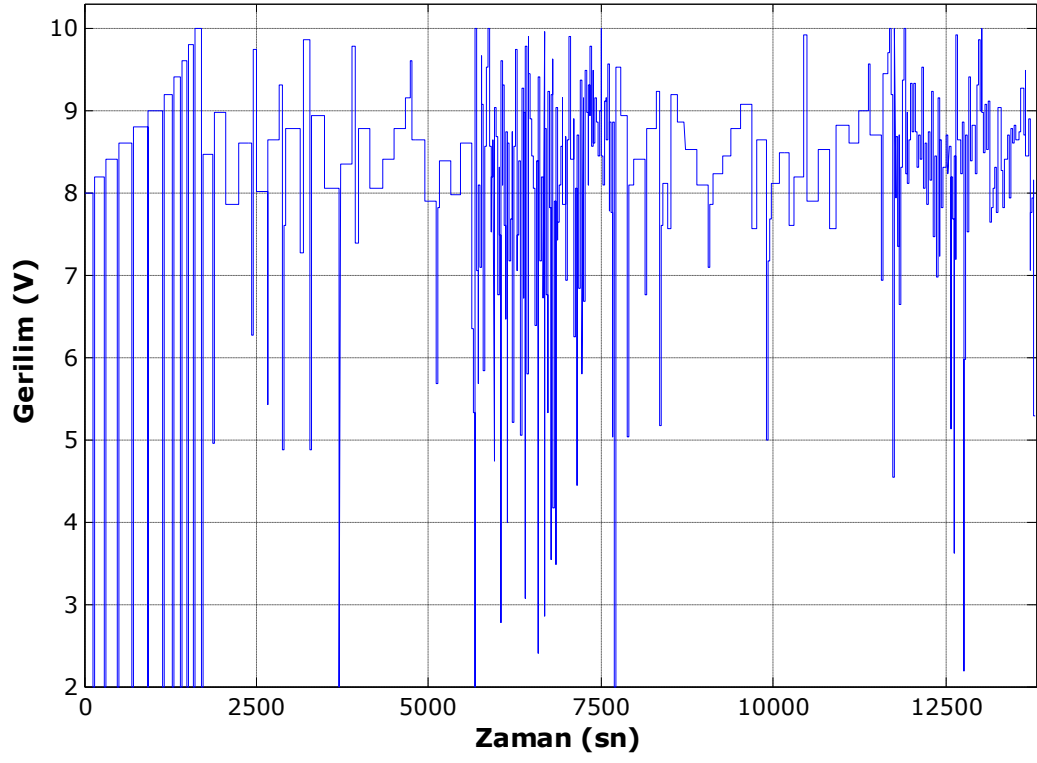
Sistemin modellenmesi, Şekil IV.1'deki blok diyagramda gösterildiği gibi seri-paralel modelleme kullanılarak gerçekleştirilmiştir. Seri-paralel modelleme sisteminde; sistem çıkışı ve onun geriye kaymış modları YSA modeline giriş olarak sağlanmaktadır. Yapılan çalışmada 3 adet zaman gecikmesi kullanılmış ve YSA'nın önceki 3 girişi ve 3 çıkışı, ağırlıklı giriş olarak uygulanmıştır.



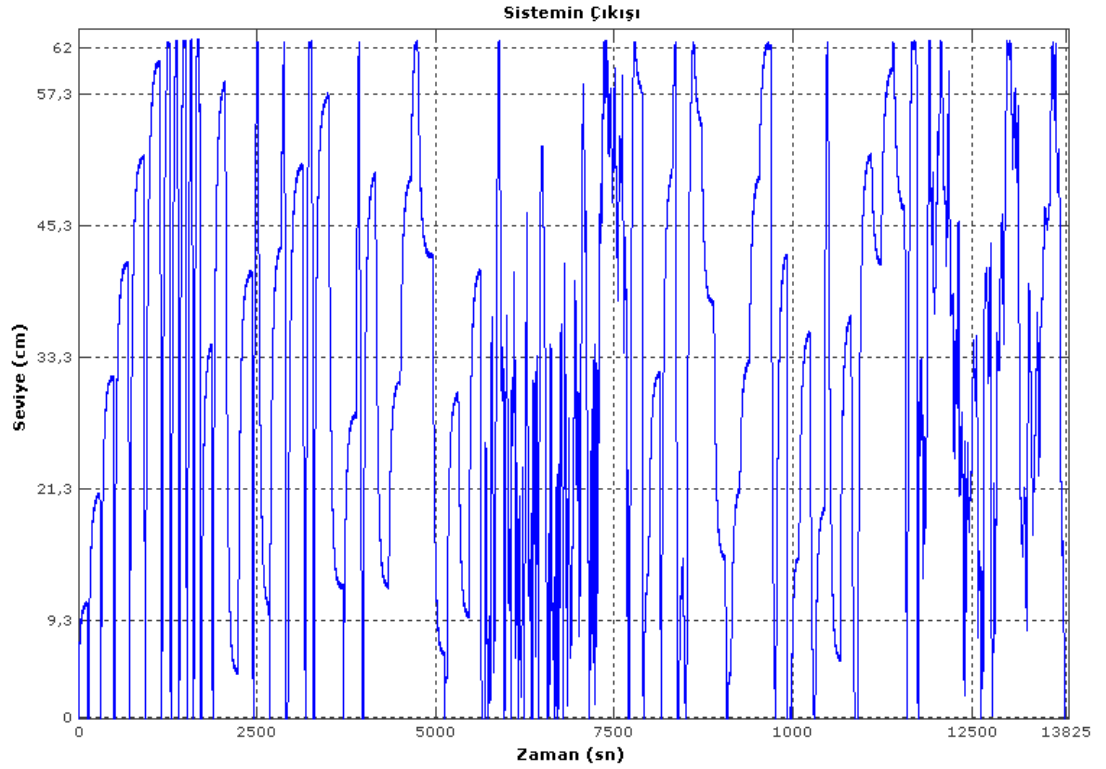
Şekil IV. 1 Gerçekleştirilen ileri tip seri – paralel NARX model blok şeması

IV.2.2. Eğitim Verisinin Toplanması

GuntRT 512 sıvı seviye sisteminin modellenmesi için sistem açık çevrimde çalıştırılmıştır. YSA'nın giriş veri seti için, sisteme Şekil IV.2'de gösterildiği gibi 2-10V arası basamak girişler uygulanmıştır. Sistemin eğitilmesi için yaklaşık 55.000 veri kullanılmıştır. Bu veriler sistemdeki motorun ısınması sonucu ortaya çıkan davranış değişikliği nedeniyle farklı zamanlarda alınmıştır. Gerek sistemin modellenmesinde, gerek parametre optimizasyonunda sisteme gönderilen ve sistemden alınan işaretler gerilim olarak kullanılmıştır. Sisteme verilen girişler neticesinde sistemden alınan seviye bilgisi (V), 1. derece bir fonksiyon ile yüksekliğe (cm) çevrilmiştir. Böylece kullanıcı arayüzünde seviye bilgisi cm cinsinden ifade edilmiştir. Şekil IV.3'de verilen giriş işaretlerine sistemin gösterdiği davranış görülmektedir. YSA eğitiminde, bu iki veri setinin birleştirilmesi ile elde edilen giriş çıkış bilgileri kullanılmış ve sistem modeli ortaya çıkartılmıştır.



Şekil IV. 2 YSA eğitimi için uygulanan girişler



Şekil IV. 3 YSA eğitimi için elde edilen çıkışlar

IV.2.3. 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, her bir gizli katmanda 5 ile 50 arasında değişen nöron sayısı, eğitim sırasında farklı hata parametreleri seçilerek oluşturulan YSA modellerinin hata değerleri ve regresyon değerleri gösterilmiştir. Elde edilen YSA modelleri, Regresyon değerleri, Karesel hatların toplamı (SSE-Sum Square Error) ve Ortalama karesel hata (MSE-Mean Square Error) olmak üzere 3 farklı performans fonksiyonu kullanılarak değerlendirilmiştir.

Tablo IV. 1 YSA mimarisi için yapılan deney sonuçları

Gizli Katman Sayısı	Nöron Sayısı	Eğitim Fonksiyonu	SSE / MSE (Hata)	Filtre	Regresyon Değeri
Tek	5	BR	0.443778 (sse)	Filtresiz	0.98754
	10	BR	0.426776 (sse)	Filtresiz	0.99031
	20	BR	0.410682 (sse)	Filtresiz	0.99862
	30	BR	0.367938 (sse)	Filtresiz	0.99972
	30	BR	0.025381 (sse)	Filtreli	0.99731
	50	BR	0.393405 (sse)	Filtresiz	0.99313
	30	LM	5.85e-006 (mse)	Filtresiz	0.97892
Çift	5	BR	0.376990 (sse)	Filtresiz	0.99024
	10	BR	0.401741 (sse)	Filtresiz	0.99341
	20	BR	0.384437 (sse)	Filtresiz	0.99749
	22	BR	0.355059 (sse)	Filtresiz	0.99168
	20	LM	4.79e-006 (mse)	Filtresiz	0.97381

Tablo IV.1’deki veriler paralelinde elde edilen modellerin, verilen referans seviyeye ulaşabilmek için izledikleri yörüngeler ve model performansları farklılık göstermektedir. Modelleme sırasında seviye bilgisini gösteren sistem çıkışına alçak geçiren 1. derece butterworth bir analog filtre eklenmiştir. Modelleme sonucunda oldukça iyi performans gösteren filtreli model, gerçek zamanlı çalışmadaki davranıştan oldukça farklı bir davranış göstermiştir. Filtreli model kullanılarak elde edilen denetleyici parametreleri gerçek zamanlı olarak sisteme uygulandığında da referans seviyeye oturamamıştır. Dolayısıyla elde edilecek modelin performansı, gerçek zamanlı kapalı çevrim çalışmada da gözlenmelidir. Aksi takdirde modelden kaynaklanan problemler denetleyici veya optimizasyon algoritmasında aranabilir. Ancak modellerin gerçek zamanlı çalışmadaki performansları kapalı çevrim çalışmada da önemlidir. Bu amaçla elde edilen farklı modeller ve kapalı çevrim

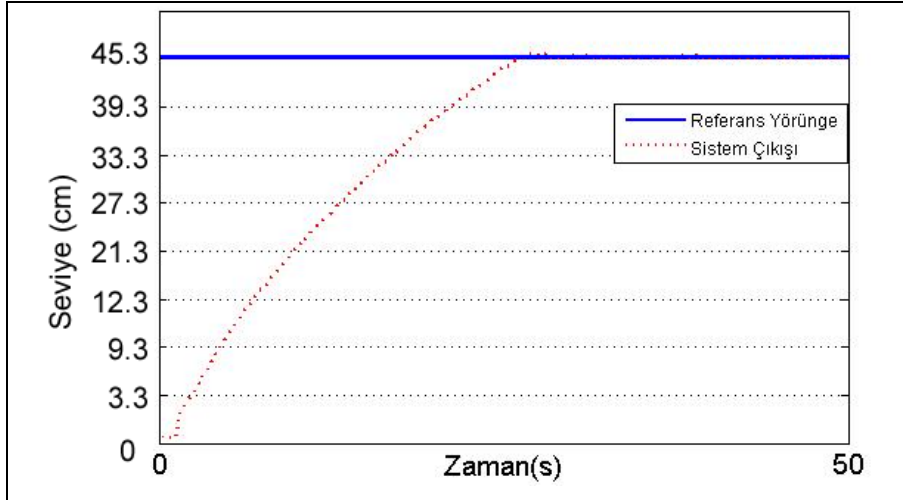
davranışları Şekil IV.4 - Şekil IV.7’de verilmiştir. Şekil IV.4’de verilen BR ile eğitilmiş tek katmanlı yapay sinir ağı modeli, SSE hatası, regresyon değeri, gerçek zamanlı çalışmadaki performansı ile diğer modellerden ayrılmaktadır. Dolayısıyla gerçek sistemi temsil edecek olan model 30 nöronlu, tek giriş katmanlı, filtresiz ve ileri yönlü besleme modeline dayalı olan bu modeldir.

Modelleme sırasında sistemden seviye bilgisi olarak gerilim okunmakta, kontrol işareti olarak gerilim bilgisi gönderilmekte ve modelleme ile ilgili tüm giriş çıkış bilgileri gerilim olarak kullanılmaktadır. Ancak gönderilen kontrol işareti sistem üzerinde bulunan gerilim / akım dönüştürücüden geçerek oransal valfe akım bilgisi gelmektedir. Diğer taraftan sistemden okunan seviye bilgisi gerilim olduğu için kullanıcı bu bilgiyi gerek referans seviyeyi izleyebilmesi gerekse seviye bilgisini anlamlandırabilmesi için Denklem IV.1’de verilen birinci derece bir fonksiyon ile gerilim bilgisi cm cinsinden seviye bilgisine dönüştürülmektedir. Bu fonksiyona göre sistemden okunan gerilim bilgisi ile yükseklik bilgisi arasındaki ilişkiyi gösteren tablo da Tablo IV.2’de gösterilmiştir.

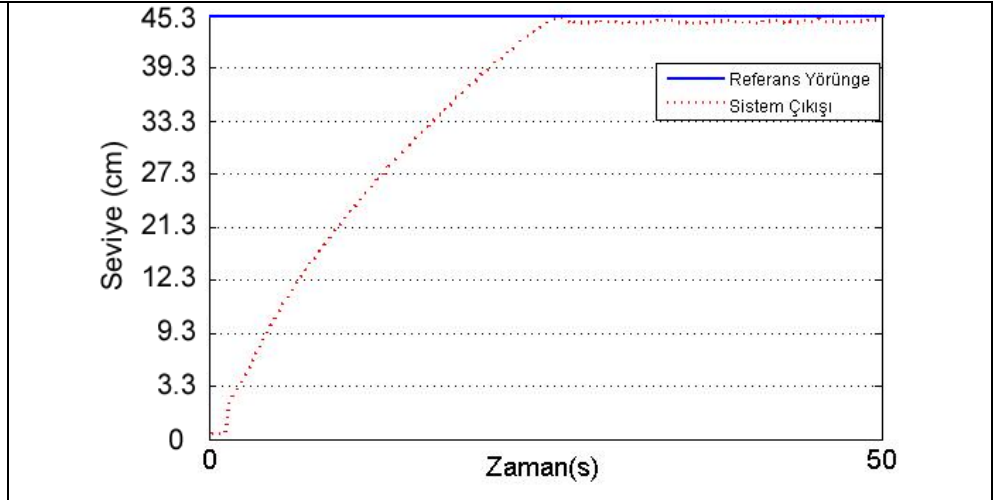
$$y = 60x - 14.7 \quad (IV.1)$$

Tablo IV. 2 Sistem çıkışından alınan gerilim(V) / yükseklik(cm) dönüşüm tablosu

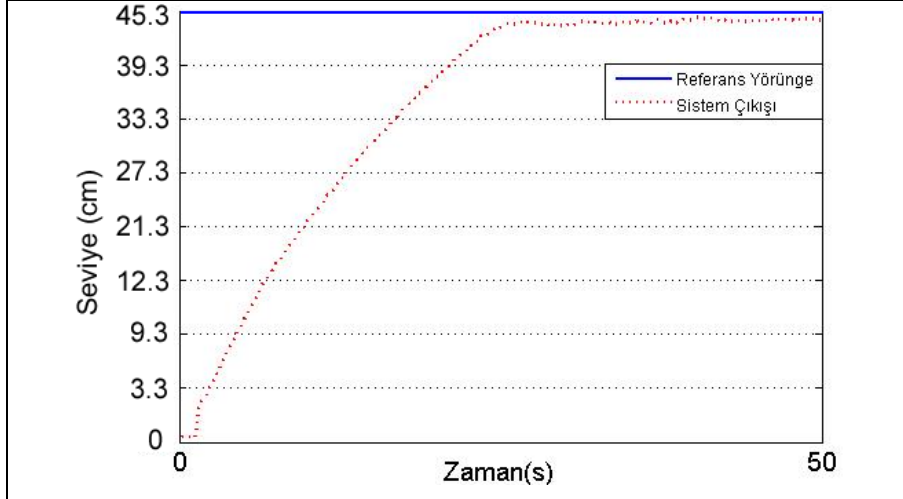
Gerilim (V)	Yükseklik (cm)
0.254	0
0.3	3.3
0.4	9.3
0.5	15.3
0.6	21.3
0.7	27.3
0.8	33.3
0.9	39.3
1.0	45.3
1.1	51.3
1.2	57.3
1.275	61.8



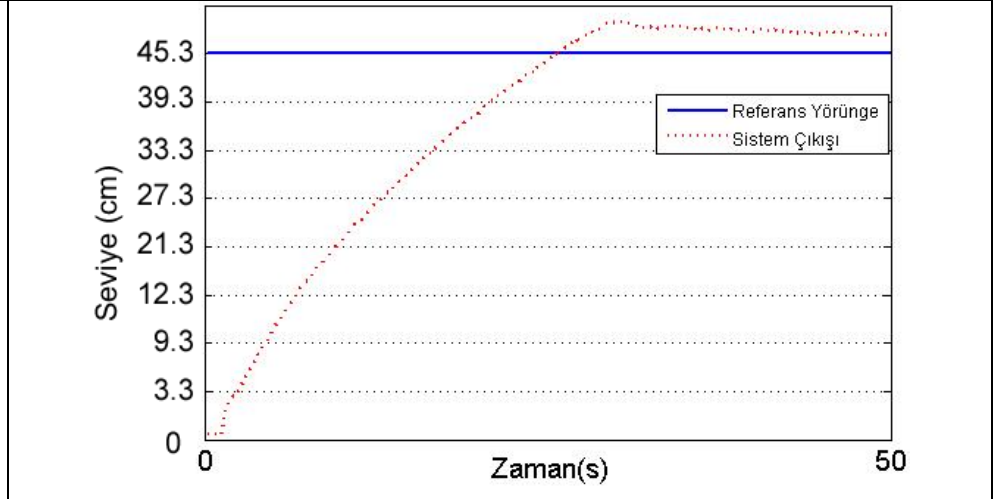
Şekil IV. 4 BR İle Eğitilmiş Tek Giriş Katmanlı Model Davranışı



Şekil IV. 5 BR İle Eğitilmiş Çift Giriş Katmanlı Model Davranışı

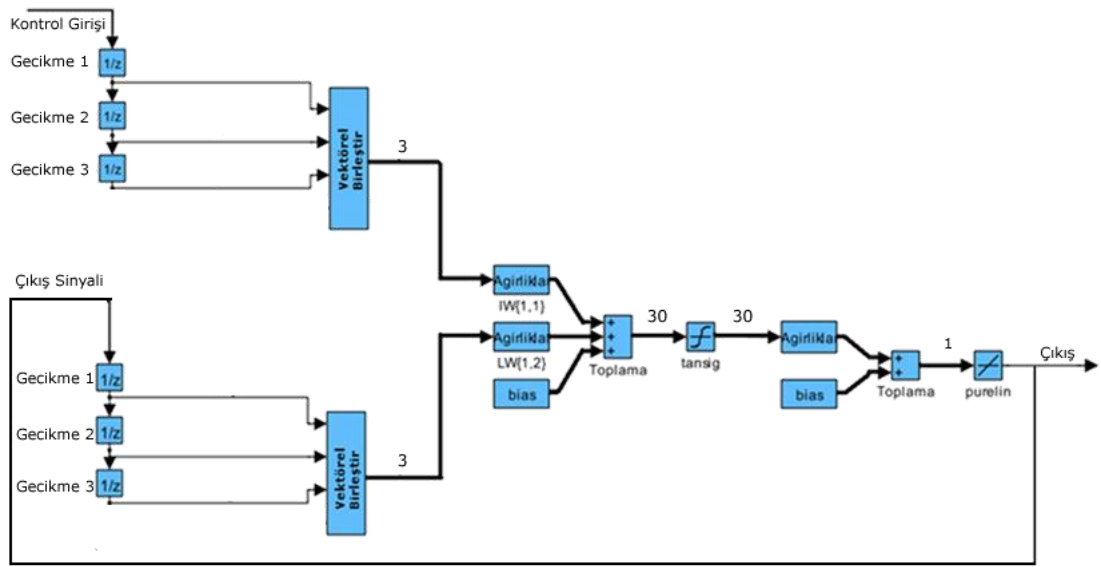


Şekil IV. 6 LM İle Eğitilmiş Tek Giriş Katmanlı Model Davranışı



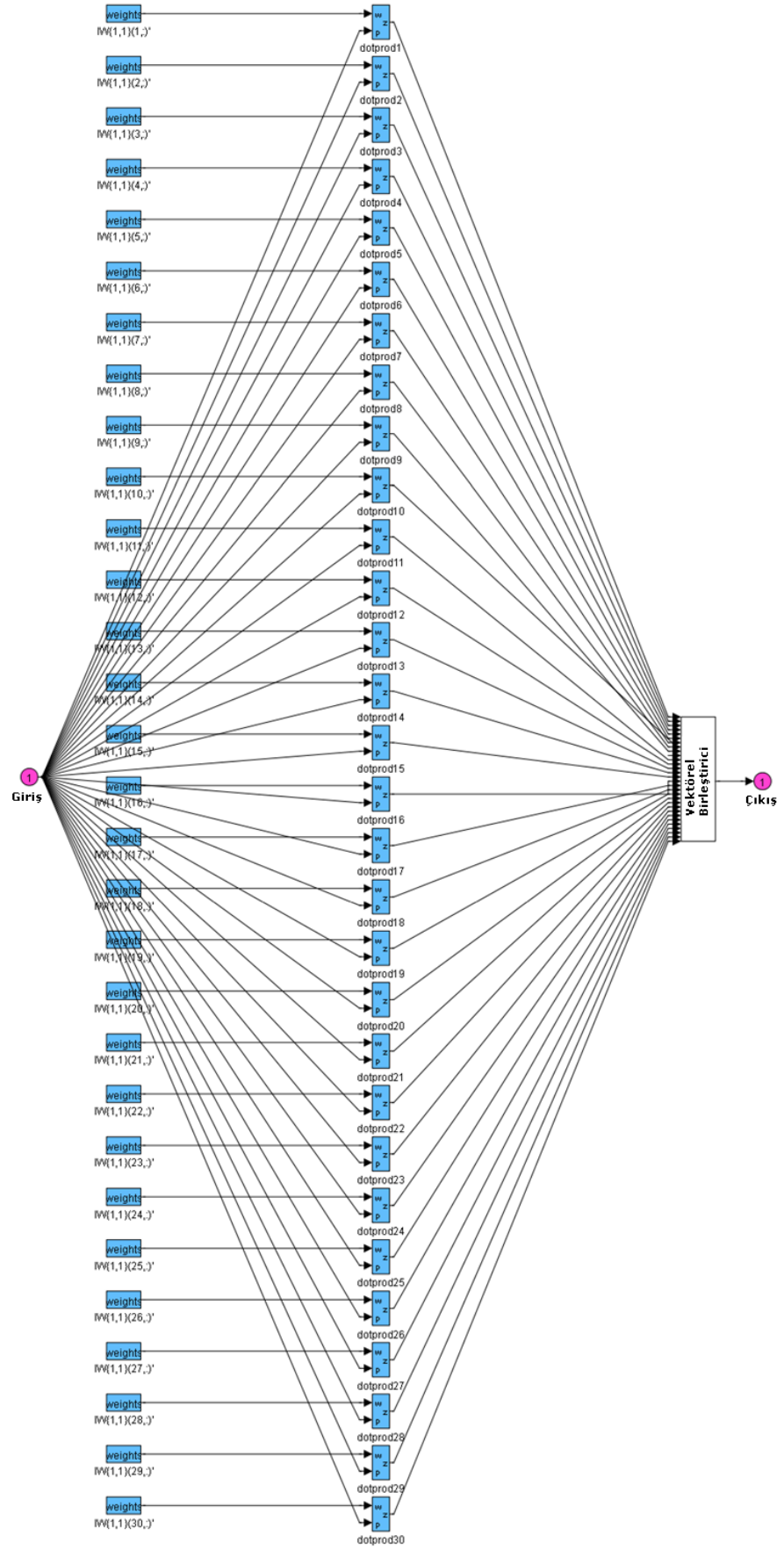
Şekil IV. 7 LM İle Eğitilmiş Çift Giriş Katmanlı Model Davranışı

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 (tansig), çıkış katmanında ise ağırlık net girdisini doğrudan çıkış olarak veren doğrusal aktivasyon fonksiyonu (purelin) seçilmiştir. Her katmanın çıkış değeri bir sonraki katmanın giriş değeridir. Bu şekilde giriş değerlerinin ağırlık girişinden çıkışına doğru ilerlemesi ile oluşturulan ileri beslemeli yapının blok diyagramı Şekil IV.8'de gösterilmektedir. YSA'nın girişinde bulunan 3 adet zaman gecikmesi ile $(1/z)$ uygulanan kontrol sinyalinin ve sistem çıkışının önceki 3 adet bilgisi ağa girişi olarak uygulanmaktadır.



Şekil IV. 8 Kullanılan YSA modelin blok diyagramı

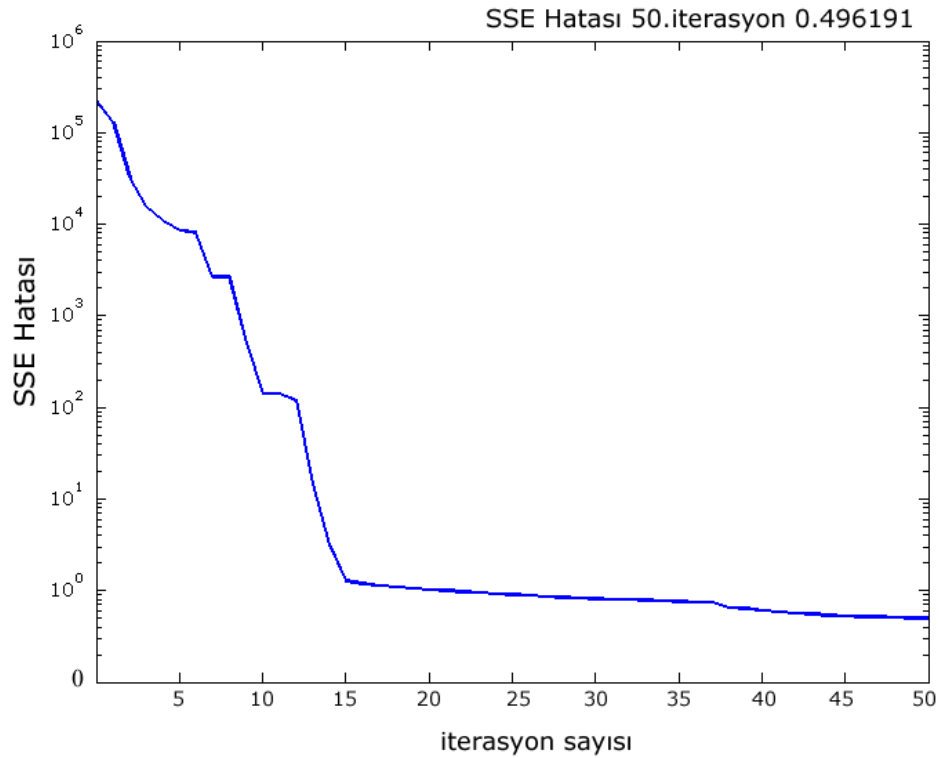
Burada IW ile gösterilen kontrol sinyalinden gelen girişlere uygulanan katsayılar ve LW çıkış sinyalinden gelen girişlere uygulanan katsayılardır. Gizli katmandaki IW olarak adlandırılan katsayıların nöron bağlantısı, Şekil IV.9'da gösterilmektedir. Girişe uygulanan kontrol ve çıkış sinyalleri sırasıyla IW ve LW katsayıları ile çarpılıp, giriş katmanı sonunda toplanarak çıkış katmanına uygulanmaktadır.



Şekil IV. 9 Gizli katmanındaki nöronların bağlantısı

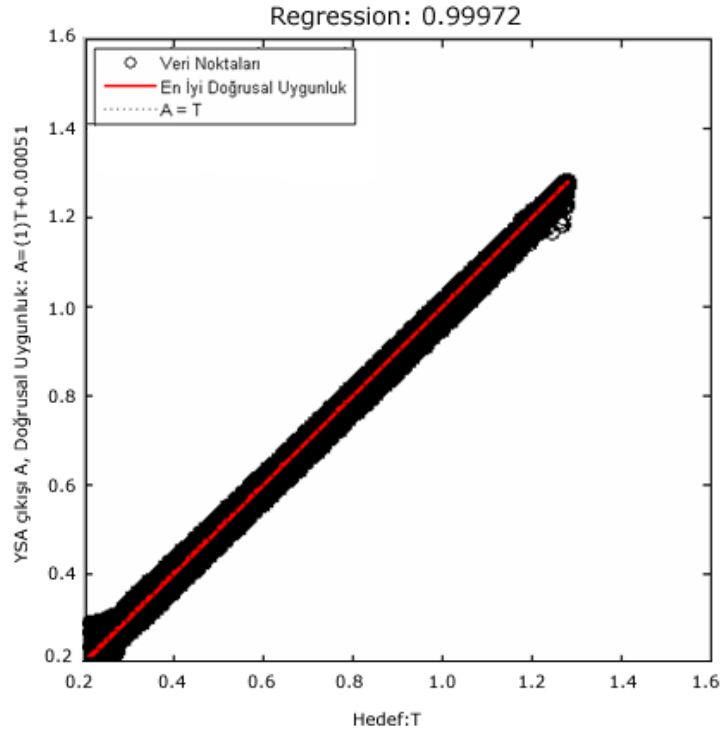
IV.2.4. YSA'nın Eğitimi

Oluşturulan YSA modeli eğitilirken, Bayesian Regularization (BR) eğitim algoritması kullanılmıştır. Eğitim adımları boyunca eğitim fonksiyonu için SSE hatası azalarak minimum değerine ulaşmaya çalışmaktadır. 50 iterasyon için eğitim sırasında SSE'deki değişim Şekil IV.10'da görülmektedir. Eğitim için kullanılan iterasyon sayısı ise daha hassas bir model elde etmek amacıyla 300 seçilmiş ve SSE değeri olarak da 0.367938 değeri elde edilmiştir.

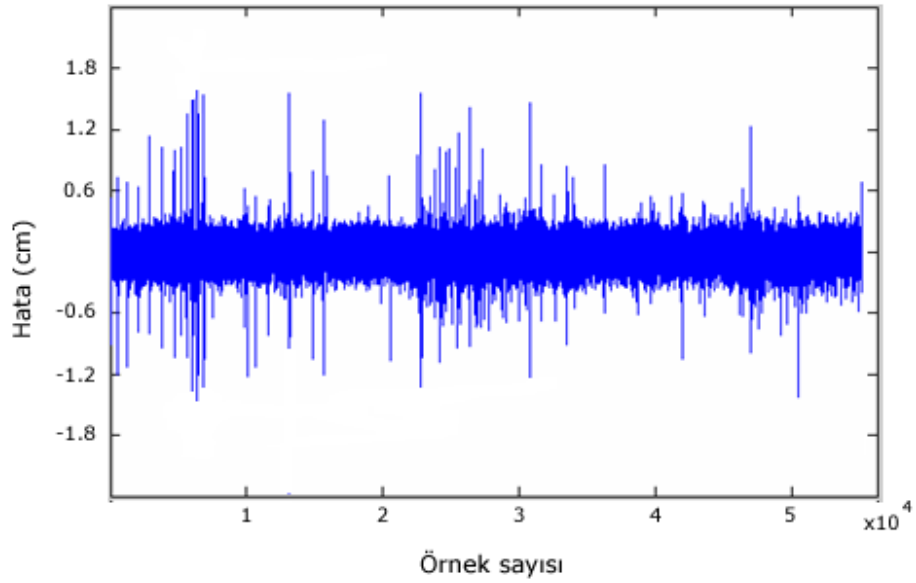


Şekil IV. 10 50 iterasyonlu eğitim süresince SSE hatasının değişim eğrisi

Eğitim sonunda sistem ve YSA model çıkışının regresyon eğrisi Şekil IV.11'de gösterilmektedir. Buradan anlaşılacağı gibi eğitim sonunda YSA model çıkışının istenilen hedef çıkışa yaklaştığı görülmektedir. 0.99972 regresyon değeri model çıkışı ile sistemden alınan test dataları arasındaki yüksek ilişkiyi göstermektedir. Modelin eğitimi sırasında eğitim verisi ile model çıkış arasındaki hata miktarı da modelin performansını gösterir. Şekil IV.12'de modelleme sırasında model ile eğitim verisindeki hatayı örnek sayısı ile ilişkilendirerek gösteren grafik görülmektedir. Model ile sistem arasındaki hata modelleme sırasında ağırlıkla $\pm 0.3\text{cm}$ aralığında değişmektedir.

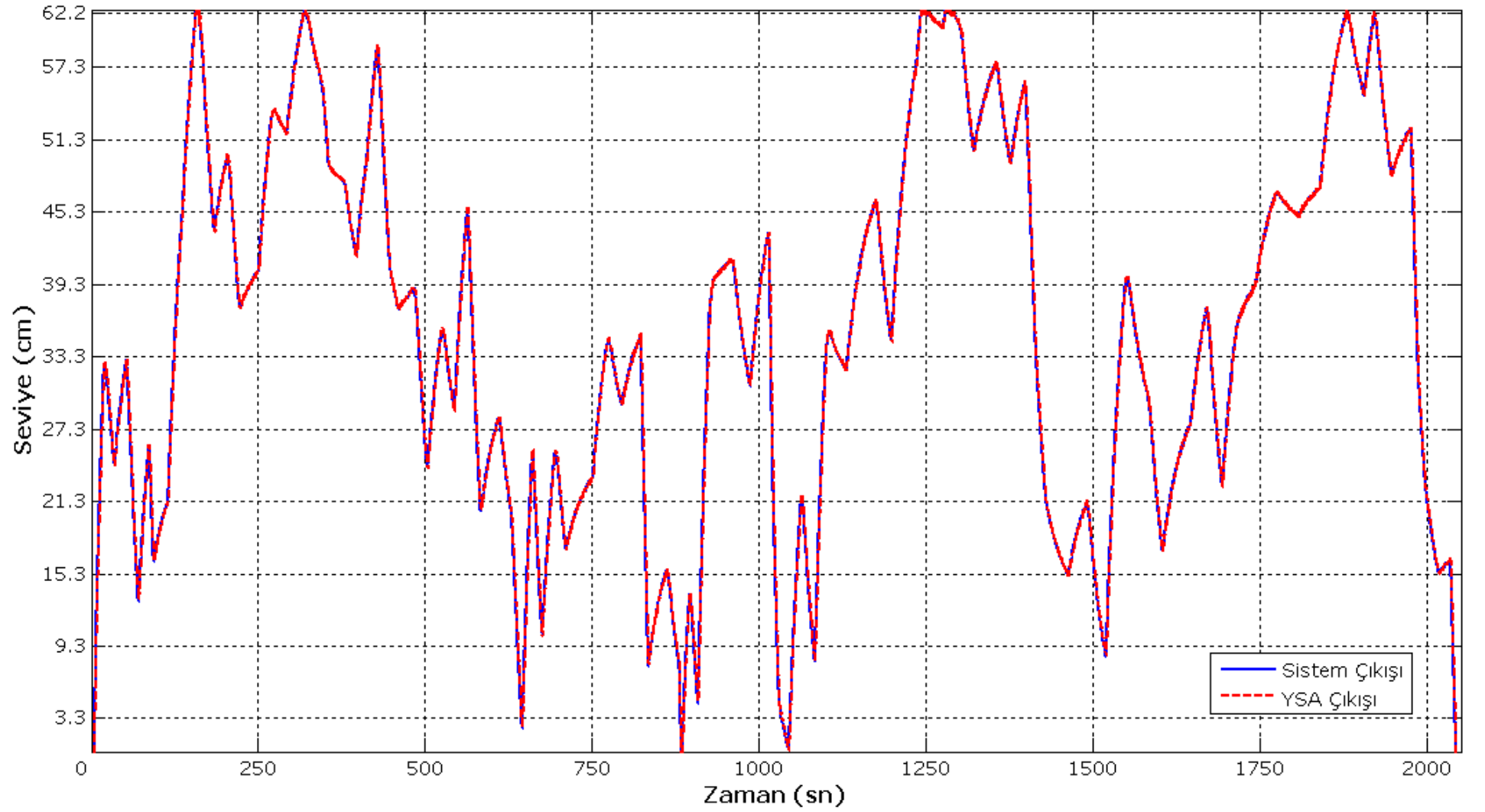


Şekil IV. 11 Hedef veriler ile YSA Model çıkışının regresyon eğrisi



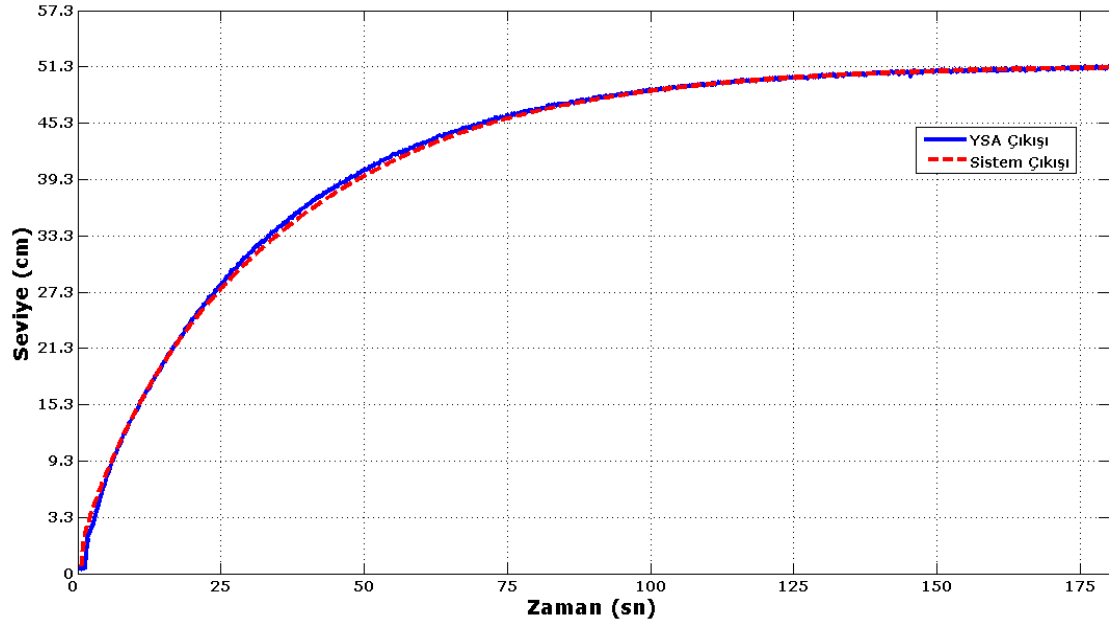
Şekil IV. 12 Hedef veriler ile YSA Model çıkışının hata eğrisi

Şekil IV.13’de ise su seviye sistemine uygulanan 2 ve 10V kontrol sinyali altında su seviye tüpündeki seviye değişimi ile eğitimin sonucu YSA modelin çıkış grafikleri verilmiştir.



Şekil IV. 13 Sistem ile YSA Modeli çıkış grafiği

Şekil IV.14’de, sisteme ve YSA modeline açık çevrimde 8.8V 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-bulanık denetleyici ve KKO-bulanık denetleyici tasarımındaki simülasyon çalışmasında, su seviye kontrol sisteminin yerine kullanılacaktır.



Şekil IV. 14 Açık çevrimde sistem ile YSA çıkışının karşılaştırılması

BÖLÜM V

SIVI SEVİYE DENETİMİ İÇİN BULANIK DENETLEYİCİ PARAMETRELERİNİN KARINCA KOLONİ VE GENETİK ALGORİTMA İLE OPTİMİZASYONU

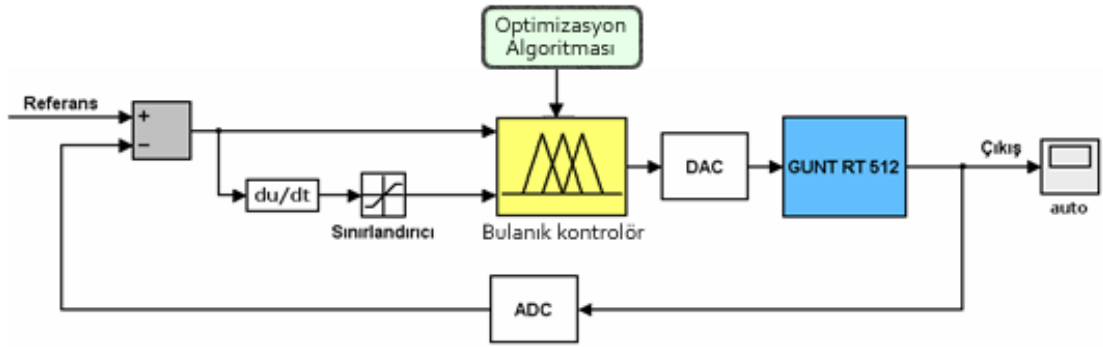
V.1. GİRİŞ

Bu bölümde Yapay Sinir Ağı ile modellenen sistem için tasarlanan bulanık denetleyici yapısı incelenmiştir. Daha sonra bulanık denetleyicinin parametreleri Genetik algoritma ile optimize edilecek ve algoritma performansı simülasyon olarak ortaya konacaktır. İkinci aşamada bulanık denetleyiciye ilişkin parametreler KKO algoritması ile de optimize edilecek ve sistemde farklı referans seviyeleri için simülasyon çalışmaları yapılacaktır.

V.2. BULANIK DENETLEYİCİ TASARIMI

1965 yılında Zadeh tarafından kullanılan bulanık mantık temel bir sistemin girişi ile çıkışı arasındaki ilişkinin bulanık kümeler ile tanımlanmasını sağlar ve günümüzde karmaşık kontrol sistemlerinde bulanık mantık yaygın olarak kullanılmaktadır [94]. Bulanık denetleyicilerin giriş çıkış değişkenlerinin, üyelik fonksiyonlarının ve aksiyon değerlerinin elde edilmesi çok zaman alan ve uzman bilgisi gerektiren ancak her zaman en iyi performans vermeyen deneme yanılma yöntemiyle yapılmaktadır [95]. Bu çalışmada, bulanık kontrol üyelik fonksiyonları sınır değerleri ve aksiyon değerleri optimizasyon algoritmaları ile elde edilmiştir. GuntRT 512 sıvı seviye sistemi için kullanılan bulanık kontrol tasarımında denetleyiciye ait iki farklı giriş ve bir çıkış bulunmaktadır.

Şekil V.1’de tasarlanan sistemin blok bağlantısı verilmiştir. Giriş değişkenleri sistemin referans sıvı seviye değeri ile mevcut seviye bilgisi arasındaki fark, farkın zamanla değişimiyle elde edilen hatadaki değişim bilgileridir.



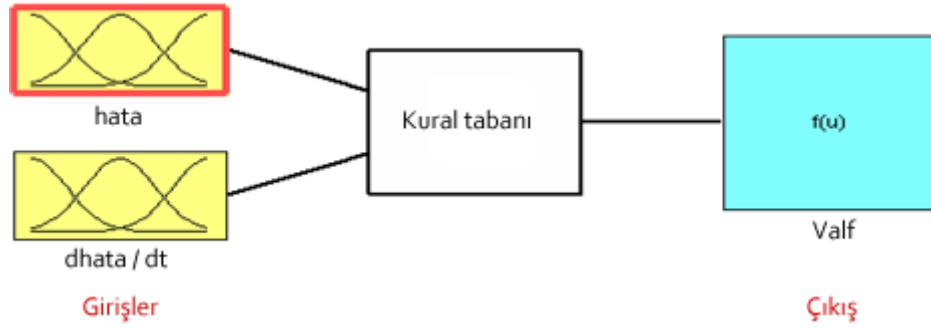
Şekil V.1 Tasarlanan denetleyici bağlantı şeması

Bulanık denetleyici tasaramında Mamdani ve Sugeno (TS) olmak üzere iki tür denetleyici karşımıza çıkmaktadır. TS denetleyicilerin Mamdani denetleyicilere göre temel olarak 2 üstünlüğü bulunmaktadır. TS denetleyicilerde çıkış doğrusal fonksiyonlar oldukları için berraklaştırma işlemi Mamdani'ye göre daha hızlı ve gerçek zamanlı sistemler için daha kullanışlıdır. Optimize işlemlerinde de çıkış değişkenine ait değerlerin optimizasyonu üyelik fonksiyonları olmadığı için TS denetleyiciler için daha kısa sürmektedir. [2] Dolayısıyla GuntRT 512 sıvı seviye sisteminin bulanık kontrolünde de 15 kuraldan oluşan TS türü bulanık denetleyici kullanılmıştır. Bulanık denetleyici yapısı içerisinde kullanılan farklı adımlarda farklı metodlar kullanılmaktadır. Sistemin giriş değişkenleri üyelik fonksiyonları ile oluşturulan kuralların arasında kullanılan and operatörü için prod ve or operatörü için probor metodları kullanılmış, çıkarım işlemi için max, yığın işlemi için min, berraklaştırma işlemi için ise wtaver (weighted average) metodları kullanılmıştır.

V.2.1. Üyelik Fonksiyonları ve Kurallar

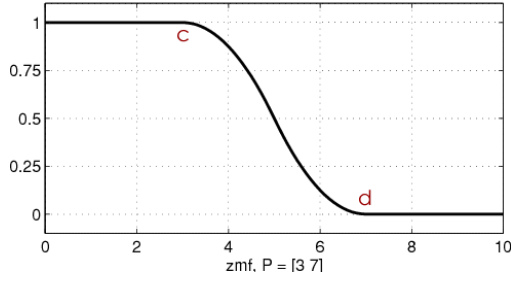
GuntRT 512 su seviye kontrol sistemi verilen referans seviyeye oturacak şekilde bulanık denetleyici ile kontrol edilecektir. TS tipi bulanık denetleyicide, girişler için Şekil V.2'de gösterildiği gibi hata ve hatadaki değişim kullanılmıştır. Bulanık denetleyici tasarımında giriş değişkenlerinin üyelik fonksiyonları adedi ve şekli önemlidir. Gerek üyelik fonksiyonlarının aralıklarının optimize edilmesinde, gerek kural geçişlerinde esneklik sağlaması sebebiyle sistemde Π -şekilli üyelik fonksiyonları kullanılmıştır. Bulanık denetleyici girişlerinden hata için NB (negatif büyük), NK (negatif küçük), S (sıfır), PK (pozitif küçük) ve PB (pozitif büyük), olmak üzere 5, hatadaki değişim için ise negatif (N), sıfır (S) ve pozitif (P) olmak

üzere 3 üyelik fonksiyonu kullanılmıştır. Bu sayede olası tüm hata ve hatadaki değişim değerleri üyelik fonksiyonlarında ilgili aralıklarda işlem görmektedir.

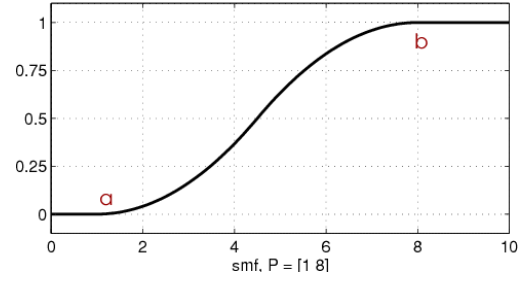


Şekil V.2 Oluşturulan bulanık denetleyici yapısı

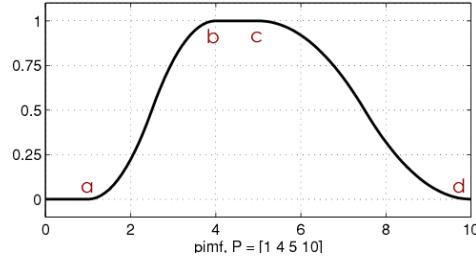
Hata ve hatadaki değişim girişlerindeki üyelik fonksiyonları için 15 kural, üyelik fonksiyonu sınır değerleri için de 24 değer ile birlikte optimize edilecek parametre sayısı 39 tanedir. Bulanık denetleyicinin üyelik fonksiyonu sınır değerleri ve kurallara ait aksiyon değerleri GA ve KKO ile optimize edilmiştir. Dolayısıyla bulanık denetleyiciye ait üyelik fonksiyonu sınır değerleri için olası çalışma aralığı belirlenmeli, optimizasyon algoritmalarının bu aralıkta arama yapması sağlanmalıdır. Bu aralık içerisinde sınır değerleri ile ilgili arama yapılırken kullanılacak kurallara ait aksiyon değerleri de optimize edilecektir. Aynı anda hem sınır değerleri hem de kurallara ait aksiyon değerleri optimize edilerek sistemin minimum hata ile kontrol edilmesini sağlayacak bulanık denetleyici parametreleri bulunacaktır. Hata ve hatadaki değişim için kullanılan üyelik fonksiyonları NB ve PB değerleri için farklılık göstermektedir. Giriş değişkenlerinin minimum ve maksimum değerlerinin dışındaki işaretlerin değerlendirilmesinde negatif sınır için zmf, pozitif sınır için ise smf tipi üyelik fonksiyonları kullanılabilir. Kullandığımız denetleyici yapısında da hata değeri seviye bilgisi olarak +1.275V (62cm)'den fazla veya -1.275V (-62cm) değerinden düşük olamaz. Kullanılan denetleyicinin hata ve hatadaki değişim girişleri için kullanılan NB (negatif büyük) üyelik fonksiyonu Şekil V.3'de verilen zmf tipindeki üyelik fonksiyonu olarak, PB (pozitif büyük) üyelik fonksiyonu ise Şekil V.4'de verilen smf tipindeki üyelik fonksiyonu olarak seçilmişlerdir. Diğer tüm üyelik fonksiyonları için ise Şekil V.5'de verilen Π -şekilli (pimf) üyelik fonksiyonu kullanılmıştır.



Şekil V.3 Z-şekilli üyelik fonksiyonu



Şekil V.4 S-şekilli üyelik fonksiyonu



Şekil V.5 []-şekilli üyelik fonksiyonu

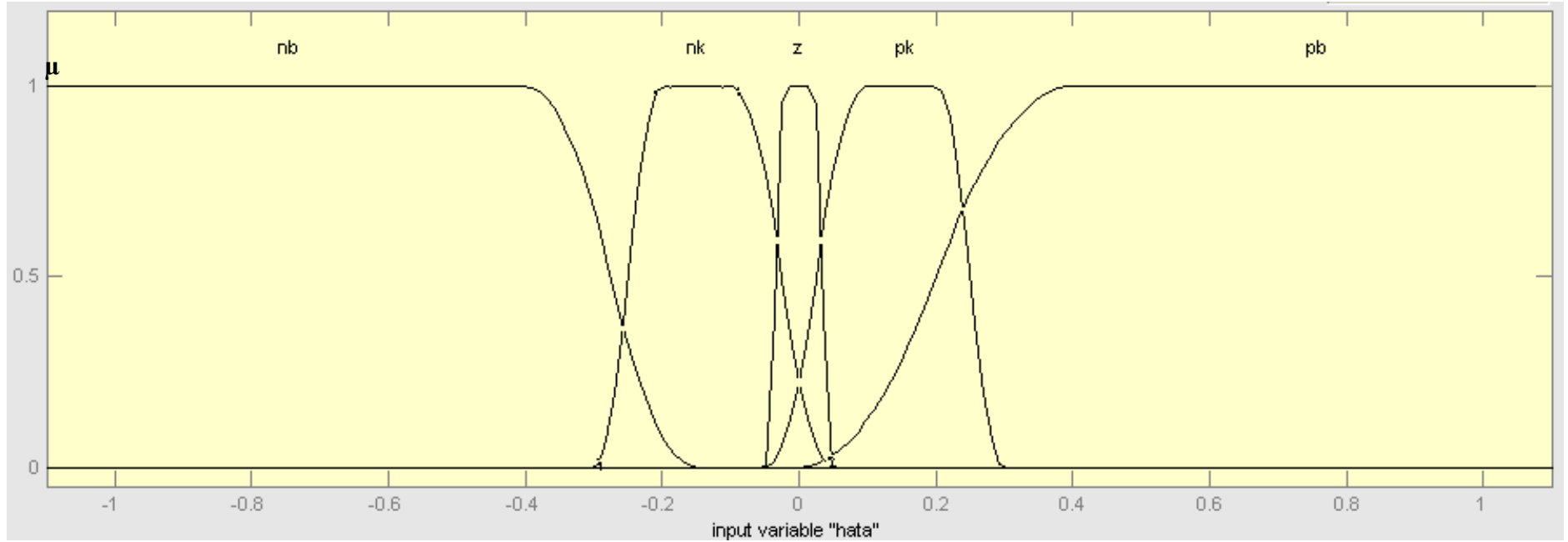
Sınır değerleri tablosundaki a,b,c,d değerleri ilgili üyelik fonksiyonlarına ait köşe noktalarını göstermektedir. Kural tabanını oluşturan 15 kural ise Tablo-V.1'de Seçilen üyelik fonksiyonları ile birlikte optimize edilecek olan sınır aralıkları maksimum ve minimum değerleri ile birlikte Tablo-V.2 ve Tablo-V.3'de, verilmiştir. Kural tablosunda da bulanık denetleyiciye aksiyon değerleri yerine çıkış değişkenlerinin isimleri yazılmıştır. Bu değişkenler GA ve KKO tarafından optimize edilecek aksiyon değerleridir. Şekil-V.6'da hata, Şekil-V.7'de ise hatadaki değişim girişine ait üyelik fonksiyonları verilmiştir.

Tablo V. 1 Bulanık denetleyici kural tablosu

Δe	e	PB	PK	S	NK	NB
N		u(1)	u(4)	u(7)	u(10)	u(13)
S		u(2)	u(5)	u(8)	u(11)	u(14)
P		u(3)	u(6)	u(9)	u(12)	u(15)

Tablo V. 2 Oluşturulan bulanık denetleyicinin *hata* girişine ait üyelik fonksiyonu sınır değerleri

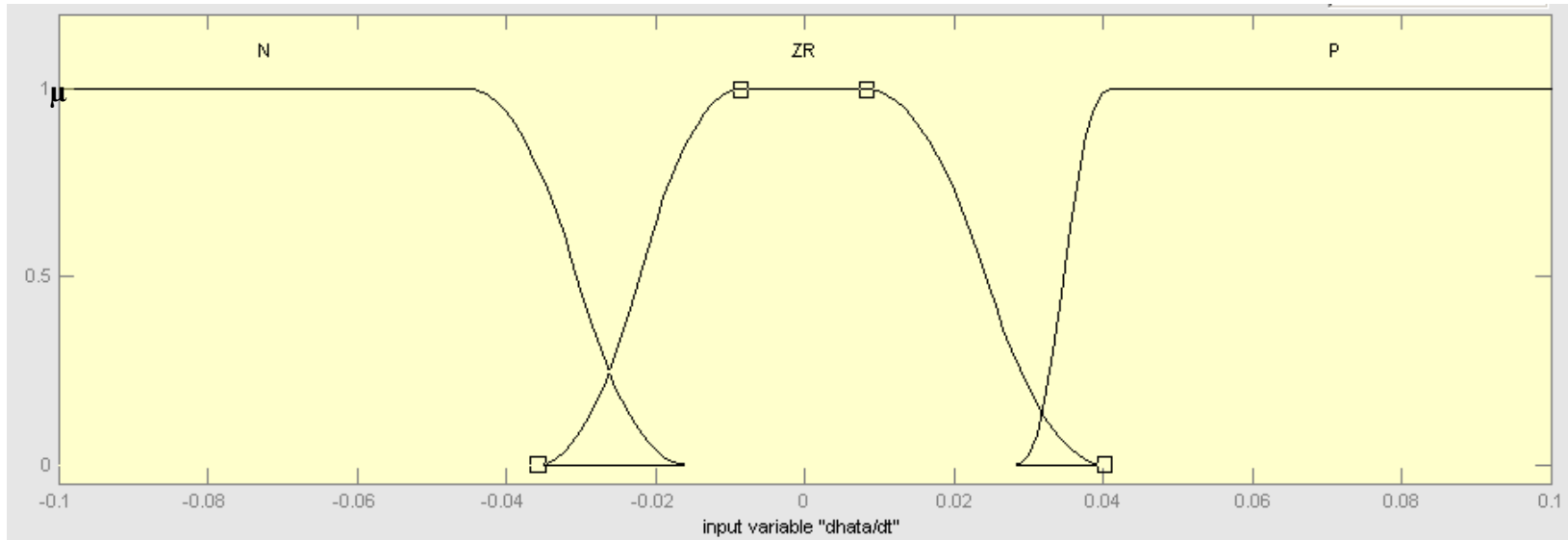
Negatif Büyük				Negatif Küçük				Sıfır				Pozitif Küçük				Pozitif Büyük			
a	b	c	d	a	b	c	d	A	b	c	D	a	b	c	d	A	b	c	d
min	max	min	max	min	max	Min	max	min	Max	min	max	min	max	min	max	min	Max	min	max
-	-	-	-	-4.2	-2.4	-3	-1.8	-1.2	-0.6	-0.24	-0.06	-1.2	-0.6	-0.48	-0.24	0.24	0.48	0.6	1.2



Şekil V.6 Oluşturulan bulanık denetleyicinin *hata* girişine ait üyelik fonksiyonları

Tablo V. 3 Oluşturulan bulanık denetleyicinin *hatadaki değişim* girişine ait üyelik fonksiyonu sınır değerleri

Negatif								Sıfır								Pozitif							
A	B	c	D					a	b	c	d					a	b	c	d				
min	Max	Min	Max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max
-	-	-	-	-3	-1.5	-1.8	-0.6	-3	-1.2	-1.2	0	0	1.2	1.2	3	0.6	1.8	1.5	3	-	-	-	-



Şekil V.7 Bulanık denetleyicinin hatadaki değişim girişine ait üyelik fonksiyonları

V.3. GENETİK ALGORİTMA İLE BULANIK DENETLEYİCİ TASARIMI

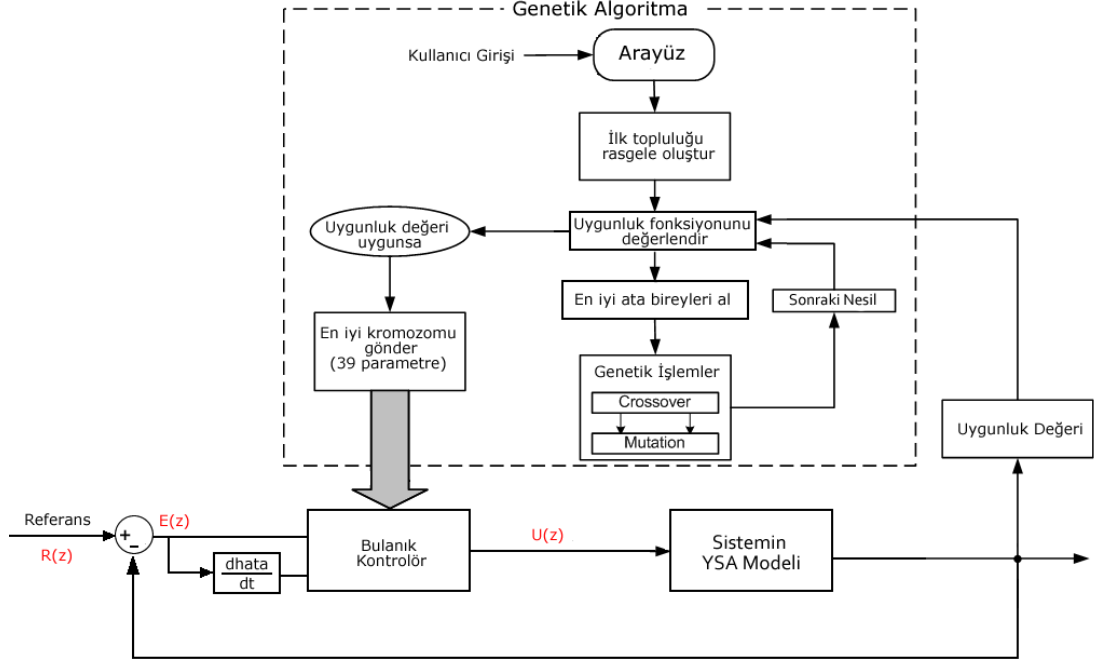
Bulanık denetleyici performansı, kural sayısı, üyelik fonksiyonları, sınır değerleri ve aksiyon değerlerinin uyuma bağlıdır. Ancak sadece kurallara ait aksiyon değerlerinin optimize edilmesi optimizasyon algoritmalarının performansını yansıtmayacak ve arama uzayında esneklik sağlamayacaktır. Dolayısıyla giriş üyelik fonksiyonları olan hata ve hatadaki değişime ait üyelik fonksiyonlarının sınır değerleri de optimize edilmelidir. Bu parametrelere uygun değerlerin atanması, sistemin istenilen referans yörüngeye oturmasını sağlayacaktır.

Bulanık denetleyicinin parametrelerinin 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. Dolayısıyla su seviye sisteminin YSA ile bir modeli çıkartılmış ve üyelik fonksiyonları ve kuralları belirlenen bulanık denetleyicinin üyelik fonksiyonu sınır değerleri ve aksiyon değerleri verilen referans yörüngeyi izleyebilecek şekilde genetik algoritma tarafından optimize edilmiştir. Bu bölümde genetik algoritma ile bulanık denetleyicinin parametrelerin bulunması süreci ele alınmıştır.

Şekil V.1’de gösterilen sistem bağlantı şemasında iki giriş ve bir çıkıştan oluşan negatif geri beslemeli sistemde, denetleyici referans ile sistemden gelen çıkışın farkını alarak oluşan hata sinyaline göre kontrol sinyali üretmektedir. Referans yörüngeyi izleyebilecek denetleyici parametreleri optimizasyonu için tasarlanan Genetik-Bulanık denetleyicinin temel blok şeması Şekil V.8’de gösterilmektedir.

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

1. Bulanık denetleyici parametrelerin ne şekilde genetik olarak kodlanması,
2. Genetik algoritmanın Bulanık denetleyici parametrelerini öğrenme metodu,
3. Sistem çıkış performansının genetik algoritma uygunluk değerine dönüştürülmesi [96].



Şekil V.8 GA ile Bulanık denetleyici parametre optimizasyonu

V.3.1. Bulanık Denetleyici Parametrelerin Genetik Olarak Kodlanması

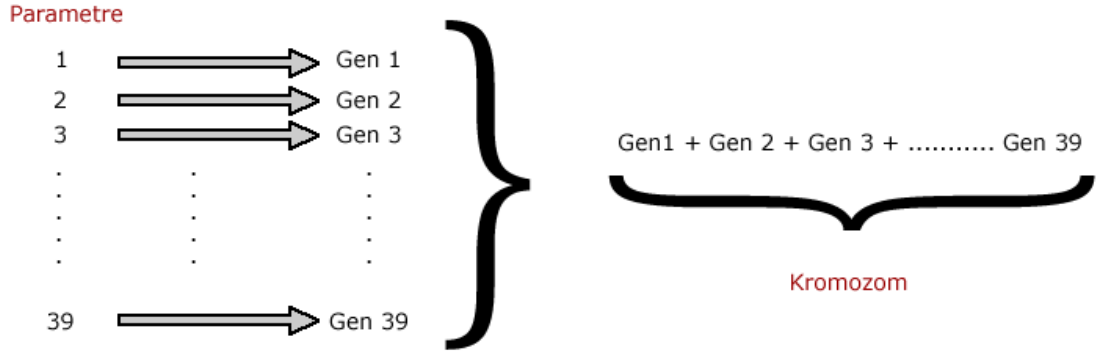
Çevrim dışı çalışma biçiminde, bulanık denetleyici parametrelerinin genetik algoritma ile bulunmasından önce, bulanık denetleyicinin parametrelerinin minimum değerlerinin belirlenmesi ve bulanık denetleyici çıkışının sınırlandırılması gereklidir. Tasarlanan sistemde de, bulanık denetleyiciye ait parametrelerin sınır değerleri Tablo V.2 ve Tablo V.3’de verilmiştir.

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 (Real Coding) olmak üzere iki tür kodlama çeşidi vardır. [17] Bu çalışmada bulanık denetleyici parametrelerinin değerleri tek bir kromozom oluşturacak şekilde gerçel sayı kodlaması ile kodlanmıştır. Bu kodlama biçiminde, üyelik fonksiyonu sınır değerlerinin optimizasyonunda hata için 16, hatadaki değişim için 8 ve kural tablosundaki kurallar için 15, toplam 39 parametre kullanılacaktır. Burada her bir parametre bir gen olarak düşünülürse, 39 adet gene ihtiyacımızın olduğunu görürüz.

V.3.2. Bulanık Denetleyici Parametrelerin GA Tarafından Öğrenilmesi

Genetik algoritma değişken parametreleri de algoritma performansı için önemlidir. Bu değişkenler genetik algoritmanın popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, kuşak aralığı, seçim stratejisi ve fonksiyon ölçeklemesi olarak sayılabilir [97]. Genetik algoritmaya ait bu değişkenleri optimal değerlerini bulmak için birçok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılabilecek parametreler bulunamamıştır [98].

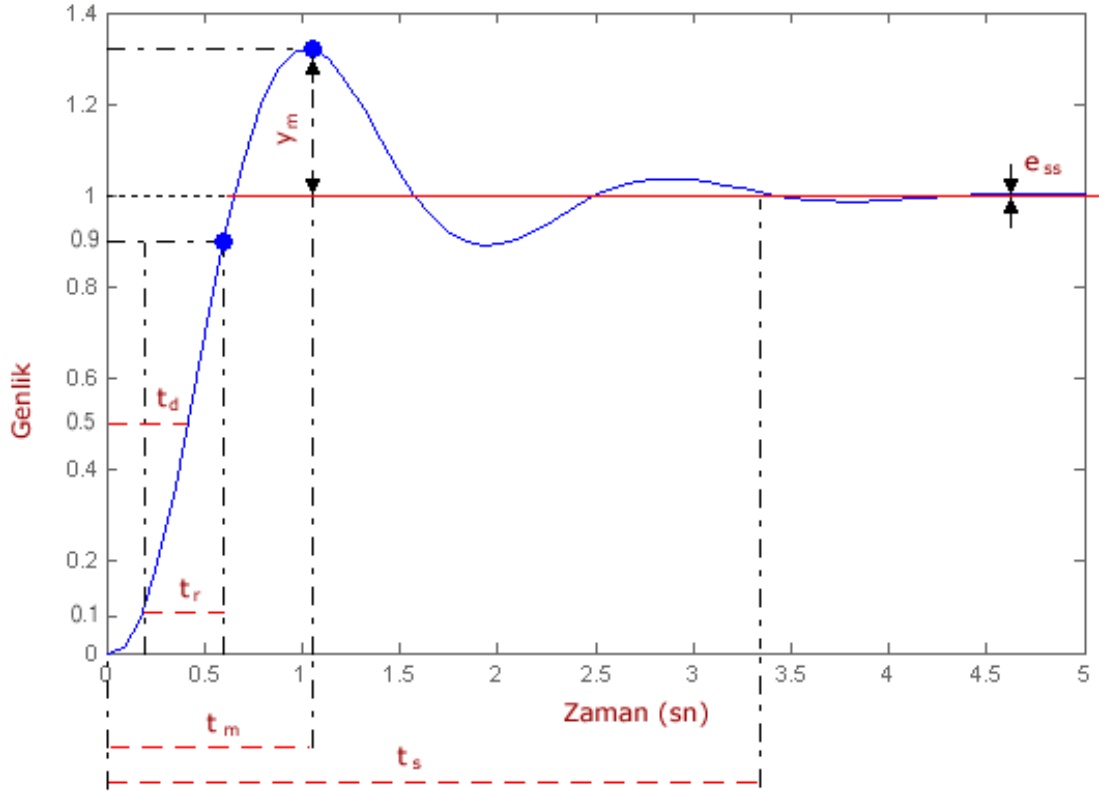
Gerçekleştirilen çalışmada kullanılan öğrenme yaklaşımı, literatürde genetik bulanık sistemlerde çok etkin çalıştığı gösterilen Pittsburgh yaklaşımıdır [66]. Bu yaklaşımı kullanmak için, bulanık denetleyicinin parametre değerleri tek bir kromozom içinde kodlanmıştır. Bu şekile göre oluşturulan Kromozom yapısı Şekil V.9’da gösterilmektedir.



Şekil V.9 Bulanık Denetleyici için geliştirilen kromozom yapısı

V.3.3. Sistem Çıkışının GA Uygunluk Değerine Dönüştürülmesi

Bir kontrol sisteminde sistem performansı, zaman uzayında incelenirken sistemin birim basamak girişine olan cevabı ortaya çıkarılır [66]. Bu incelemede kullanılan temel performans kriterleri Şekil V.10’da gösterilmektedir. Bunlar, Maksimum aşım miktarı (ym), gecikme zamanı (td), yükselme zamanı (tr), yerleşme zamanı (ts), kalıcı hal hatası (ess) parametreleridir. Bu sayede elde edilen değerler ile GA’nın performansı üzerinde ilgili parametreler üzerinden değerlendirme yapılabilmektedir. Ancak yapılan çalışmada GA’nın uygunluk değeri olarak RMS hatası kullanılmıştır.



Şekil V.10 Tipik bir birim basamak giriş için sistem cevabı

V.3.3.1. Maksimum Aşım miktarı (y_m)

Maksimum aşım, sistemin kalıcı hal durumu (y_{ss})’den ilk taşma miktarını gösterir. Buna göre maksimum aşım miktarı $y_m = y_{max} - y_{ss}$ şeklinde ifade edilir. Burada y_{max} ulaşılan en yüksek genlik değerini, y_{ss} ise kalıcı hal seviyesini göstermektedir. Maksimum aşım miktarı sistemin bağlı kararlılığını ölçmek için kullanılır. Büyük aşım miktarları pek çok durumda tercih edilmez.

V.3.3.2. Gecikme Zamanı (t_d)

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

V.3.3.3. Yükselme Zamanı (t_r)

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

V.3.3.4. Yerleşme Zamanı (ts)

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

V.3.3.5. 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 incelemeye oldukça önemli oldukları görülür. Yapılan çalışmada, sistemin çıkışının verilen bir referans seviyeye oturması ya da bir yörünge boyunca izlemesi istendiği için, referansa oturuncaya kadar oluşacak hatayı minimize edecek bir uygunluk fonksiyonu oluşturulmuştur. Bu uygunluk fonksiyonu denklem V.1’de gösterilmiştir.

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

Burada Φ , uygunluk değeri k , çevrim dışı çalışmada örnekleme sayısını 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 hata değeri kullanılarak denetleyicinin performansı, GA için uygunluk değerine çevrilmiştir. Denklem V.1’deki formülden elde edilen uygunluk değerine göre sistemin çıkışı ile referans arasındaki RMS hatasının minimum olduğu bulanık denetleyici parametreleri GA ile bulunmaktadır. Böylece verilen referansı bulanık denetleyici sistem, minimum hata ile takip etmektedir.

V.3.4 Kullanılan Genetik Operatörler

GA’da temel olarak kodlama, populasyon büyüklüğü, çaprazlama, mutasyon ve seçim gibi genetik operatörler kullanılır. Kullanılan genetik operatörlerin değerlerinin seçiminde deneme yanılma yöntemini kullanmak gerek zaman gerek performans değerlendirme için etkin sonuç vermeyebilir. Bu yüzden sistemin alternatif parametrelerle farklı kereler çalıştırılarak elde edilen RMS hatasının ortalaması alınmış ve en düşük RMS değeri veren parametreler genetik algoritmanın operatör parametreleri olarak kullanılmış ve bu değerler Tablo V.4’de verilmiştir.

Gerçekleştirdiğimiz sistemde kullanmış olduğumuz genetik operatörler, değerleri şöyledir;

Kodlama: Bölüm III.2’de GA’nın sistemin parametreleriyle değil kodlanmış biçimiyle çalıştığı belirtilmişti. Yapılan çalışmada kodlama çeşidi olarak gerçel sayı kodlama seçilmiştir ve bulanık denetleyici parametre 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 $N = 40$ seçilmiştir.

Çaprazlama: Dağıtılmış (scattered) 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 elit model kullanılmıştır. Literatürde elit model, popülasyon içinde en iyi uygunluk değerine sahip olan dört bireyin, bir sonraki jenerasyona değişikliğe uğramadan aktarılması olarak tanımlanır. Bu şekilde, en uygun bireyin kromozomlarının seçim etkisi ile sonraki jenerasyonlara aktarılmadan ölmesinin önüne geçilir. Elit model ile GA’nın performansı artmıştır.

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

Durdurma kriterleri: Genetik algoritma ile parametre optimizasyonu için kullanılan bazı durdurma kriterleri de kullanılmıştır. Bu kriterler genellikle istenilen uygunluk değerine ulaşıldığı zaman veya ulaşması mümkün görünmediği zaman devreye giren parametrelerdir. Kullanılan genetik algorithmada 150 jenerasyon veya $1e-006$ hatadaki değişim değerine ulaşıldığı zaman optimizasyon sona erdirilmektedir.

Tablo V. 4 Farklı deęerler ile alıřtırılan genetik operatrler ve RMS deęerleri

	Pop_size	Elit_Count	Cros_Frac	Cros_Fcn	Selection	RMS
1	40	4	0.4	Scattered	Tournament_4	0.0166
2	40	4	0.4	Scattered	Tournament_8	0.0169
3	40	4	0.4	Scattered	Roulette	0.017
4	40	4	0.4	Arithmetic	Tournament_4	0.023
5	40	4	0.4	Arithmetic	Tournament_8	0.0204
6	40	4	0.4	Arithmetic	Roulette	0.0209
7	40	4	0.8	Scattered	Tournament_4	0.0153
8	40	4	0.8	Scattered	Tournament_8	0.0162
9	40	4	0.8	Scattered	Roulette	0.0158
10	40	4	0.8	Arithmetic	Tournament_4	0.019
11	40	4	0.8	Arithmetic	Tournament_8	0.0226
12	40	4	0.8	Arithmetic	Roulette	0.0231
13	40	8	0.4	Scattered	Tournament_4	0.0215
14	40	8	0.4	Scattered	Tournament_8	0.0171
15	40	8	0.4	Scattered	Roulette	0.0182
16	40	8	0.4	Arithmetic	Tournament_4	0.0218
17	40	8	0.4	Arithmetic	Tournament_8	0.0196
18	40	8	0.4	Arithmetic	Roulette	0.0213
19	40	8	0.8	Scattered	Tournament_4	0.0163
20	40	8	0.8	Scattered	Tournament_8	0.0158
21	40	8	0.8	Scattered	Roulette	0.0167
22	40	8	0.8	Arithmetic	Tournament_4	0.0192
23	40	8	0.8	Arithmetic	Tournament_8	0.0208
24	40	8	0.8	Arithmetic	Roulette	0.0203
25	80	4	0.4	Scattered	Tournament_4	0.018
26	80	4	0.4	Scattered	Tournament_8	0.0187
27	80	4	0.4	Scattered	Roulette	0.0192
28	80	4	0.4	Arithmetic	Tournament_4	0.0188
29	80	4	0.4	Arithmetic	Tournament_8	0.0201
30	80	4	0.4	Arithmetic	Roulette	0.0215
31	80	4	0.8	Scattered	Tournament_4	0.0178
32	80	4	0.8	Scattered	Tournament_8	0.0188
33	80	4	0.8	Scattered	Roulette	0.0178
34	80	4	0.8	Arithmetic	Tournament_4	0.0188
35	80	4	0.8	Arithmetic	Tournament_8	0.0204
36	80	4	0.8	Arithmetic	Roulette	0.0208
37	80	8	0.4	Scattered	Tournament_4	0.0188
38	80	8	0.4	Scattered	Tournament_8	0.0181
39	80	8	0.4	Scattered	Roulette	0.0183
40	80	8	0.4	Arithmetic	Tournament_4	0.021
41	80	8	0.4	Arithmetic	Tournament_8	0.0196
42	80	8	0.4	Arithmetic	Roulette	0.0206
43	80	8	0.8	Scattered	Tournament_4	0.0188
44	80	8	0.8	Scattered	Tournament_8	0.0169
45	80	8	0.8	Scattered	Roulette	0.0193
46	80	8	0.8	Arithmetic	Tournament_4	0.019
47	80	8	0.8	Arithmetic	Tournament_8	0.0197
48	80	8	0.8	Arithmetic	Roulette	0.0206

Tablo V.4’de gösterildiği gibi seçilen farklı operatörlerin performansları RMS hataları ile ortaya konmuş ve 7. satırda verilen değerler kullanılan genetik algoritmanın parametreleri olarak kullanılmıştır. Bu değerler sistem arka arkaya 5 kere çalıştırılarak elde edilen ortalama değerler olduğu için parametreleri doğruluk ve tutarlılığı da sağlanmıştır.

Elde edilen en iyi Genetik algoritma sonuçları için kullanılan genetik algoritma parametreleri ile optimize edilen bulanık denetleyiciye ait üyelik fonksiyonu sınır değerleri ve aksiyon değerleri Tablo V.5, Tablo V.6 ve Tablo V.7’de verilmiştir. Burada elde edilen tüm değerler algoritmanın her çalıştırılmasında birbirinden farklı sonuçlar üretecektir.

Tablo V.5 Genetik Algoritma ile Optimize Edilen *hata* Girişi Sınır Değerleri

NB		NK				Z				PK				PB	
-2.5296	-1.9218	-4.2306	-1.9992	-0.7296	-0.1038	-0.6528	-0.306	0.4206	0.762	0.0684	0.6024	2.7738	4.6746	1.9824	3.1104

Tablo V.6 Genetik Algoritma ile Optimize Edilen *dhata/dt* Girişi İçin Sınır Değerleri

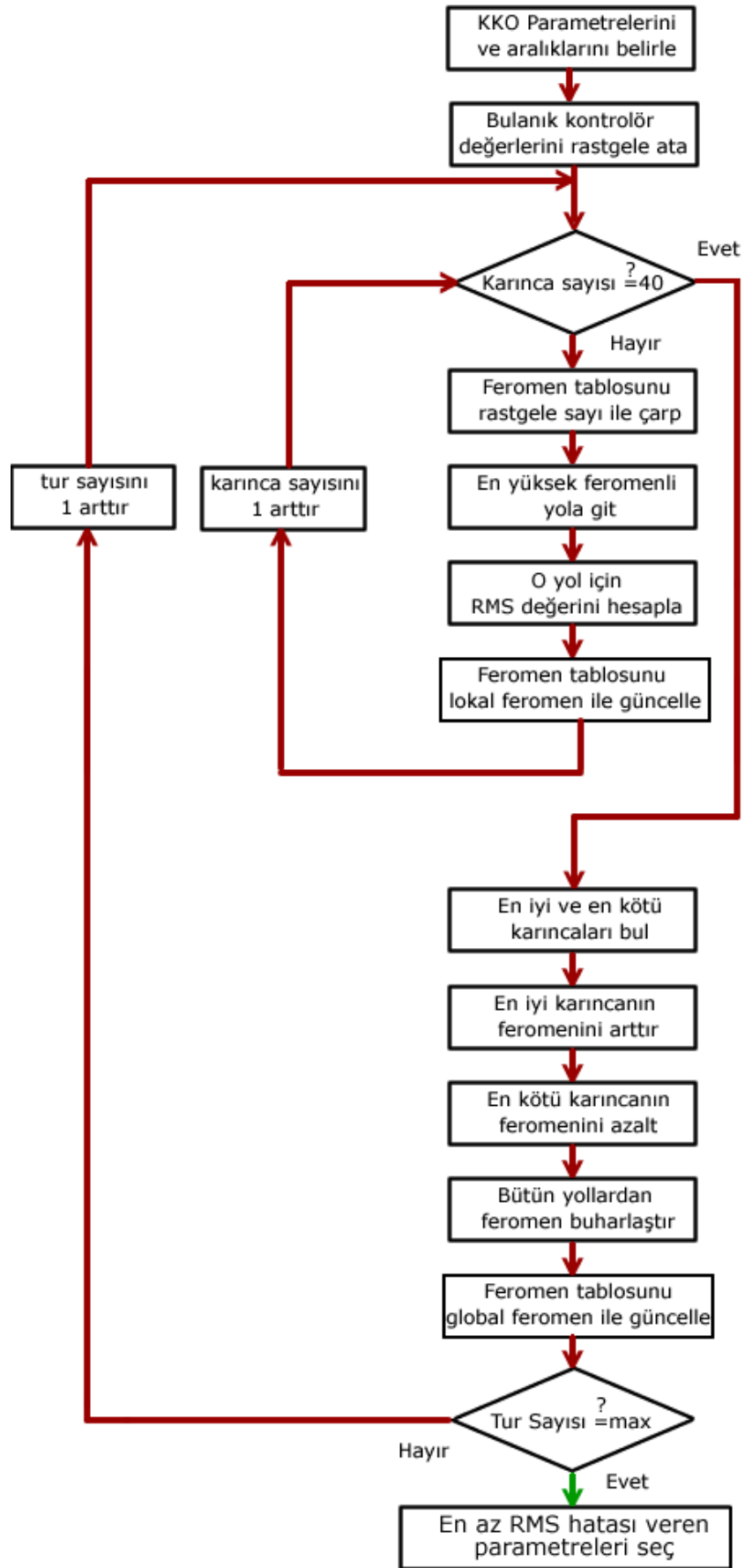
N		Z				P	
-2.574	-1.77	-2.79	-0.036	0.108	2.682	1.29	1,722

Tablo V.7 Genetik Algoritma ile Optimize Edilen Aksiyon Değerleri

Δe	PB	PK	Z	NK	NB
N	9.878	8.238	7.499	5.595	2.043
Z	9.951	9.951	8.434	5.342	2.014
P	9.902	8.061	8.517	5.987	2.107

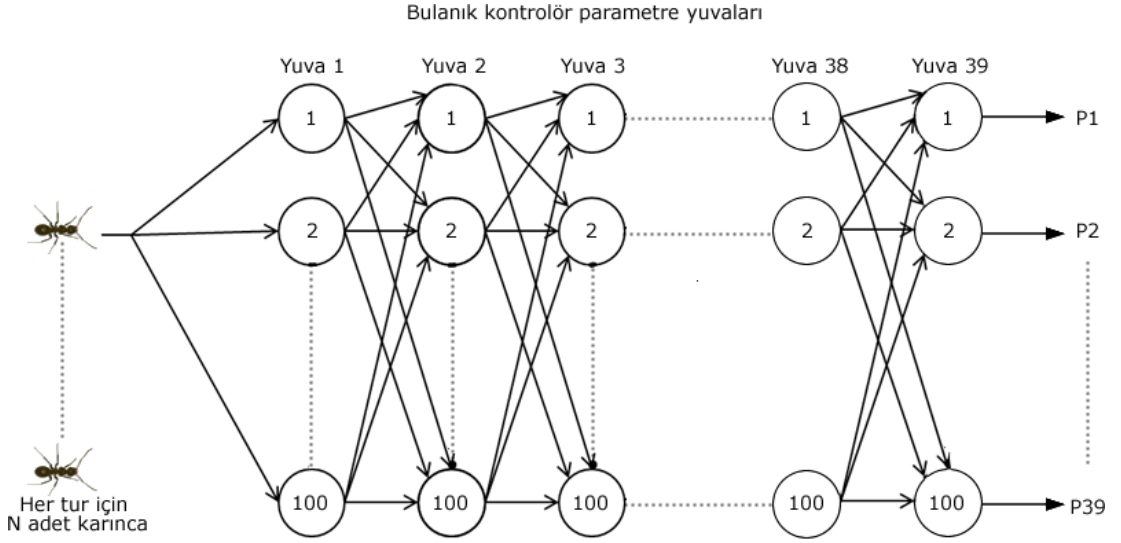
V.4. KARINCA KOLONİ OPTİMİZASYONU ALGORİTMASI İLE BULANIK DENETLEYİCİ TASARIMI

Karıncaların en kısa yolu bulma mantığından yola çıkılarak elde edilen KKO algoritması ile herhangi bir denetleyiciye ait parametre optimizasyonu sağlanabilmektedir. Bulanık kontrollü bir sistemin parametrelerini KKO algoritması ile optimize etmek mümkün olmaktadır. Şekil V.11’de verilen referansı izlemek üzere gerekli bulanık denetleyici parametrelerinin bulunması için KKO algoritması ile tasarlanmış optimizasyonun işaret akış diyagramı görülmektedir. KKO algoritmasında her yapay karıncanın seçmiş olduğu yollar farklı bulanık denetleyici parametrelerine karşılık gelmektedir. Optimizasyon işlemi başladığında KKO, denetleyici için bulanık denetleyici parametrelerini atar ve bu parametreler ile sistem çevrimdışı çalıştırılır. Sisteme referans giriş olarak bir referans seviye veya yörünge verilmektedir. Böylece KKO’nun bulacağı parametre değerleriyle çalıştırılan simülasyondan elde edilen çıkış ile referans karşılaştırılacaktır. Birbirinden farklı bu çıkış eğrileri aslında her bir karıncanın izlediği yol olarak düşünülebilir. Dolayısıyla referansa en yakın çıkışı verecek olan parametre değerleri 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 arasında az hata oluşturan yani yapay karıncalar tarafından fazla tercih edilerek feromeni yüksek olan yollara ait değerler, optimizasyon sonucu elde edilen parametre değerleridir. 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. KKO algoritmasının tasarımında dikkat edilmesi gereken bazı özellikler vardır.



Şekil V.11 KKO Algoritması ile Bulanık Denetleyici Parametre Optimizasyonu [54]

Öncelikle optimize edilecek değişkenlerin arama uzayının çok büyük olmasını engellemek için maksimum ve minimum değerleri seçilir. Bu sayede KKO, sınırlandırılan çalışma uzayı içerisinde arama yapabilecektir. Şekil V.12’de gösterildiği gibi bulanık denetleyici için her bir parametre değeri bir yuva gibi düşünülürse bu yuvalara giren bir karınca bir sonraki yuvayı seçerken varsa feromeni fazla olan yolu, yoksa (ilk karınca) rastgele bir yuva seçerek diğer yuvaya girmektedir.



Şekil V.12 Bulanık denetleyici parametreleri için karıncanın izleyebileceği yollar

Her yuvadan çıkışında önceki yuvalardan aldığı değerleri de tutmaktadır. 39. yuvadan çıktığı zaman elde ettiği değerler karıncanın izlediği yol ile elde edilecek çıkış eğrisini verecektir. Dolayısıyla elde edilecek çıkış, referans karşılaştırılarak karıncanın izlediği yollara “ödül” veya “ceza” feromenleri verilecektir. Gerçekleştirilen çalışmada, aynı tur içerisinde “yerel feromen güncelleme”, her bir tur sonunda ise “global feromen güncelleme” olmak üzere iki şekilde güncelleme yapılmaktadır. Yerel feromen güncelleme işleminde her bir karınca izlediği yolu bitirdikten sonra bırakılan feromeni her bir tur içinde Denklem V.2’de verilen eşitlik ile güncellemektedir [99].

$$\tau_{ij}(t) = \frac{0.2a}{c(i)} \quad (V.2)$$

Burada τ_{ij} , feromen matrisinin satır ve sütun indisindeki feromen miktarını, a yerel feromen güncelleme katsayısı, $c(i)$ ise karınca tarafından o tur içerisinde elde

edilen referans ile elde edilen çıkış eğrisi arasındaki RMS hatası yani yolun maliyetidir. 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 V.3 ve Denklem V.4’de verilen eşitlikler ile tekrar değiştirilmektedir [99].

$$\tau_{iyi_{ij}}(t) = \tau_{iyi_{ij}}(t-1) + \frac{a}{c_{iyi}} \quad (V.3)$$

$$\tau_{kötü_{ij}}(t) = \tau_{kötü_{ij}}(t-1) - \frac{0.6a}{c_{kötü}} \quad (V.4)$$

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 V.5 ve Denklem V.6’da verilmiştir [99].

$$\tau_{ij}(t) = \tau_{ij}(t-1)^\lambda + \Delta \quad (V.5)$$

$$\Delta = \tau_{iyi} + \tau_{kötü} \quad (V.6)$$

Burada;

λ : Buharlaşma katsayısı

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

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

Δ : Her tur sonunda global feromen güncelleme matrisi

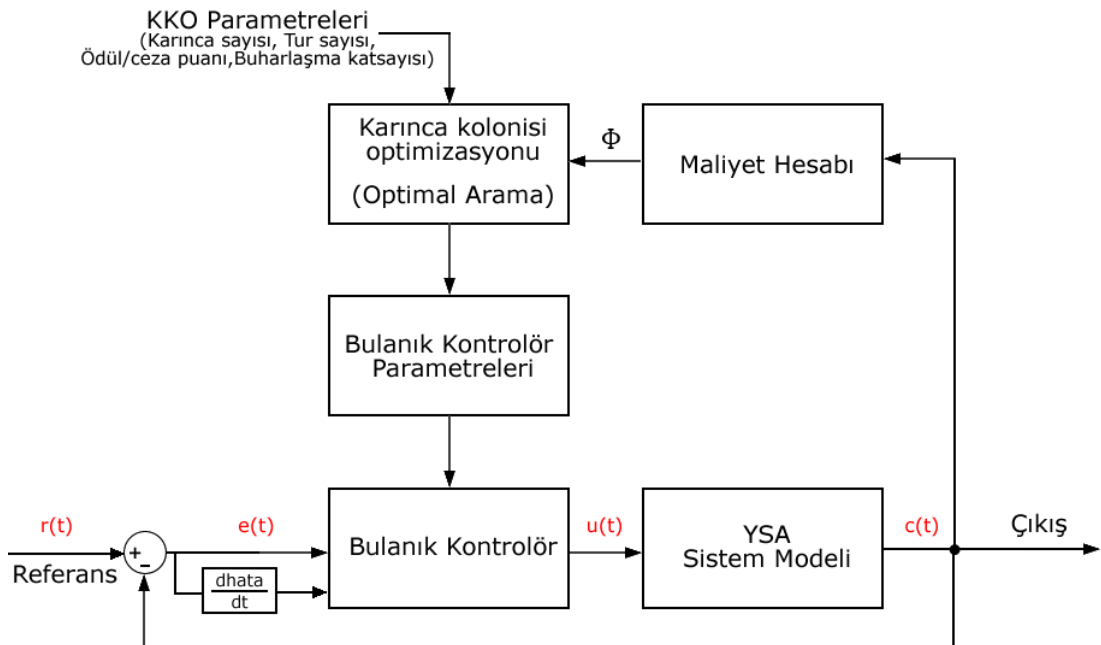
Burada KKO değişkenlerinin (λ, a) değerleri deneme yanılma ile verilmelidir. En kısa zamanda referansa yakın sonuç veren değerler o sistem için seçilen en uygun parametre değerleridir. Benzer şekilde çıkış eğrisi de referansa oturmuştur.

V.4.1. KKO Algoritması Parametrelerinin Bulunması

KKO algoritması ile bulanık denetleyicinin parametreleri elde edilirken seçilen KKO algoritmasının parametreleri deneme yanılma metodu ile bulunmuştur. Yapay karıncaların geçecekleri yuva, bulanık denetleyici parametreleri olarak ayarlanmıştır. Parametrelerin alt ve üst sınır değerleri çalışma aralığı göz önüne alınarak sisteme verilmiştir. Her bir parametre alt limit ile üst limit arasında 100'er parçaya bölünerek, optimizasyon çalıştırıldığında yapay karıncaların geçebilecekleri yuvalar yani her turda bir karıncanın seçebileceği parametre değerleri olarak ayarlanmıştır. KKO ile bulanık denetleyici parametreleri bulunurken izlenen adımlar Şekil V.13'de verilmiştir. İlk olarak optimizasyon için KKO parametreleri belirlenmiştir. Bu parametrelerin belirlenmesi süreci;

- bir turdaki karınca sayısı,
- tur sayısı
- her bir karıncanın tur içerisindeki performansına göre verilen ödül veya ceza,
- her bir karıncanın bir yolu izledikten sonra o yola bıraktığı feromon miktarının buharlaşmasına ilişkin buharlaşma katsayısı

olmak üzere 4 değişken, algoritma farklı kereler çalıştırılarak RMS değerleri ile birlikte elde edilmiştir.



Şekil V.13 KKO-Bulanık denetleyici tasarımı ile kontrol edilen sistem şeması

Elde edilen bu deęerler bir tablo üzerine yerleřtirilmiř ve karınca koloni optimizasyonu algoritmasının sistem iin en uygun parametreleri Tablo V.8’de grlen RMS deęerlerine gre seilmiřtir. Bu tablo sonularından da grleceęi zere her turda 40 karınca kullanıldıęı ve feromen sabitinin $a = 0.05$ olarak alındıęı rneklerde buharlařma sabitinin her deęeri iin RMS hatasının minimum deęerlerde olmaktadır. Dolayısıyla sistemin parametre optimizasyonunda bu deęereler ile bulanık denetleyicinin 39 parametresi optimize edilerek gerek sistemde kullanılacak yelik fonksiyonlarına ait sınır deęerleri ve aksiyon deęerleri elde edilmiřtir.

KKO algoritması parametreleri seildikten sonra yapay karıncaların gidebilecekleri yolların hepsine eřit sayıda feromen deęeri verilmiř ve bu deęer feromen matrisinde tutulmuřtur. İlk yapay karınca yuvadan ıktıęında gidebileceęi btn yollarda eřit miktarda feromen olduęu iin rastgele bir yol seerek toplam 39 parametrenin her biri iin 39 farklı yuvaya uęrayarak yolunu tamamlamıřtır. Yolun sonunda seilen bulanık denetleyici parametresi ile simlasyon alıřtırılmıř ve referans ile simlasyon ıkıřı arasındaki RMS hatası hesaplanarak bu yapay karıncanın gemiř olduęu yolların feromen matrisindeki deęeri arttırılmıřtır. Arkasından gelen yapay karıncanın da tur iinde aynı yoldan gitmemesi iin feromen tablosu rastgele sayı ile arpılarak gncellenmiřtir. Bylece her turda karıncalar farklı parametre deęerlerini alarak verilen referansa oturmaya alıřmıřlardır.

40 yapay karınca da turunu tamamladıęında bu turdaki en dřk RMS hatasına sahip yapay karıncanın yolundaki feromen miktarı arttırılmıř ve en fazla RMS hatasına sahip yapay karıncanın yolundaki feromen miktarı iyi olan karıncanın arttırılan miktarından daha fazla oranda azaltılmıřtır. Ayrıca her tur sonunda btn yapay karıncaların getikleri yollardaki feromen miktarı, buharlařma sabiti(λ) ile buharlařtırılmıřtır. Eęer seilen bulanık denetleyici parametreleri ile kk RMS hatası elde edilmiř ise bir sonraki karıncalarda bu katsayılar etrafında dolanarak toplam turlarını bitirmeye alıřacaktır. Maksimum tur tamamlandıęında feromen tablosundaki en yksek deęere sahip yollar, yani seilmiř olan parametreler optimizasyon sonucu olarak kaydedilir.

Tablo V.8 Farklı deęerleri ile alıřtırılan KKO operatrleri ve RMS deęerleri

	Karınca sayısı	Puan sabiti (a)	Buharlařma katsayısı	RMS
1	10	0.005	0.001	0.0171
2	10	0.005	0.01	0.0165
3	10	0.005	0.1	0.0165
4	10	0.005	0.5	0.0159
5	10	0.05	0.001	0.015
6	10	0.05	0.01	0.0152
7	10	0.05	0.1	0.015
8	10	0.05	0.5	0.015
9	10	0.5	0.001	0.0153
10	10	0.5	0.01	0.0149
11	10	0.5	0.1	0.0167
12	10	0.5	0.5	0.0158
13	10	1	0.001	0.0184
14	10	1	0.01	0.0184
15	10	1	0.1	0.0163
16	10	1	0.5	0.0165
17	20	0.005	0.001	0.0164
18	20	0.005	0.01	0.0162
19	20	0.005	0.1	0.0163
20	20	0.005	0.5	0.0157
21	20	0.05	0.001	0.0146
22	20	0.05	0.01	0.0146
23	20	0.05	0.1	0.0145
24	20	0.05	0.5	0.0147
25	20	0.5	0.001	0.0149
26	20	0.5	0.01	0.0158
27	20	0.5	0.1	0.0154
28	20	0.5	0.5	0.0153
29	20	1	0.001	0.0172
30	20	1	0.01	0.0163
31	20	1	0.1	0.0158
32	20	1	0.5	0.016
33	40	0.005	0.001	0.0162
34	40	0.005	0.01	0.0159
35	40	0.005	0.1	0.0157
36	40	0.005	0.5	0.015
37	40	0.05	0.001	0.0145
38	40	0.05	0.01	0.0145
39	40	0.05	0.1	0.0146
40	40	0.05	0.5	0.0147
41	40	0.5	0.001	0.016
42	40	0.5	0.01	0.0149
43	40	0.5	0.1	0.0154
44	40	0.5	0.5	0.0158
45	40	1	0.001	0.0154
46	40	1	0.01	0.0177
47	40	1	0.1	0.0164
48	40	1	0.5	0.0163

Elde edilen en iyi KKO algoritması sonuları iin kullanılan KKO parametreleri ile optimize edilen bulanık denetleyiciye ait yelik fonksiyonu sınır deęerleri ve aksiyon deęerleri Tablo V.9, Tablo V.10 ve Tablo V.11’de verilmiřtir.

Tablo V.9 KKO Algoritması ile Optimize Edilen *hata* Girişi Sınır Değerleri

NB		NK				Z				PK				PB	
-3.228	-1.848	-3.648	-2.964	-0.696	-0.06	-0.81	-0.45	0.385	0.834	0.078	0.66	2.352	4.008	1.908	2.652

Tablo V.10 KKO Algoritması ile Optimize Edilen *dhata/dt* Girişi Sınır Değerleri

N		Z				P	
-2.502	-1.668	-2.208	-1.008	0.972	2.352	0.864	2.58

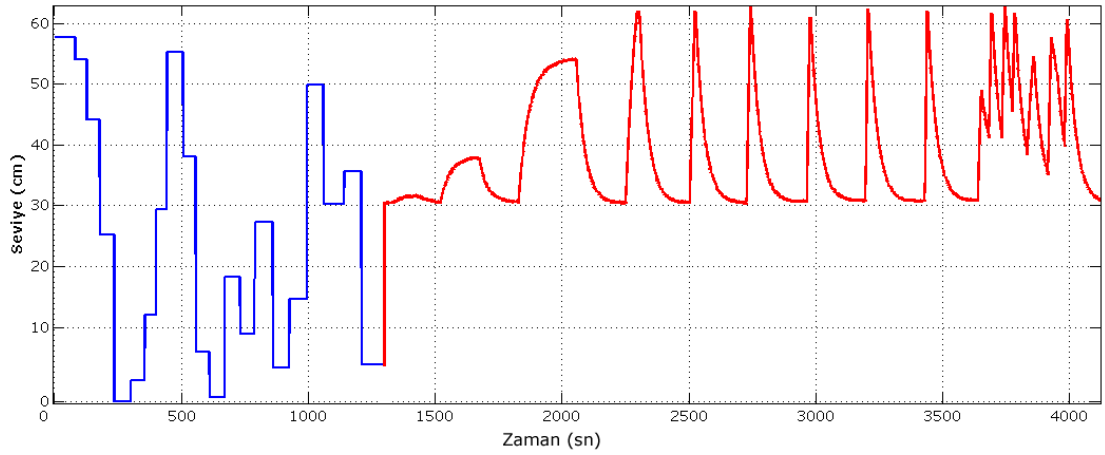
Tablo V.11 KKO Algoritması ile Optimize Edilen Aksiyon Değerleri

Δe	E	PB	PK	Z	NK	NB
N		9.96	8.4	8.14	5.03	2.01
Z		9.99	10	8.4	5.03	2.01
P		9.92	8.8	8.5	7.28	2.04

Burada elde edilen tüm değerler algoritmanın her çalıştırılmasında birbirinden farklı olacaktır.

V.5. RASTGELE GİRİŞ İŞARETLERİ İLE PARAMETRE OPTİMİZASYONU

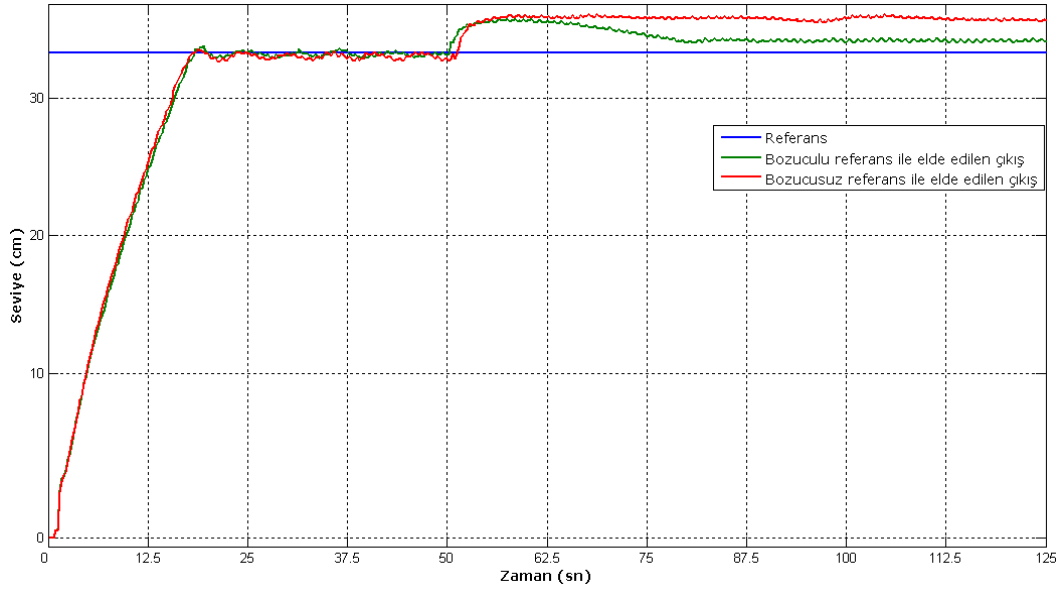
YSA ile modellenen sistemler verilen bozuculardan bağımsız olarak modellenmektedirler. Sistem bozucunun herhangi bir kademesindeki değeri için modellenir ve elde edilen model üzerinden parametre optimizasyonu yapılır. Kullandığımız GuntRT 512 sıvı seviye sisteminde modelleme işlemi tahliye vanası % 100 açıklıkta iken yapılmış ve parametre optimizasyonu da bu model üzerinden gerçekleştirilmiştir. Parametre optimizasyon işlemlerinde sisteme gerçek zamanlı alınan bozucu işaretler verildiğinde optimizasyon sonucu elde edilen parametrelerle sistemin gerçek zamanlı çalışmada daha başarılı sonuçlar verdiği görülmüştür. Bozuculu sistemlerde, verilecek referansın bozucular ile gerçek sistem davranışına uygun hale getirilmesiyle elde edilen parametreler gerçek zamanlı çalışmada daha etkin sonuçlar vermiştir.



Şekil V.14 Parametre optimizasyonunda bozucu ile kullanılan referans eğrisi

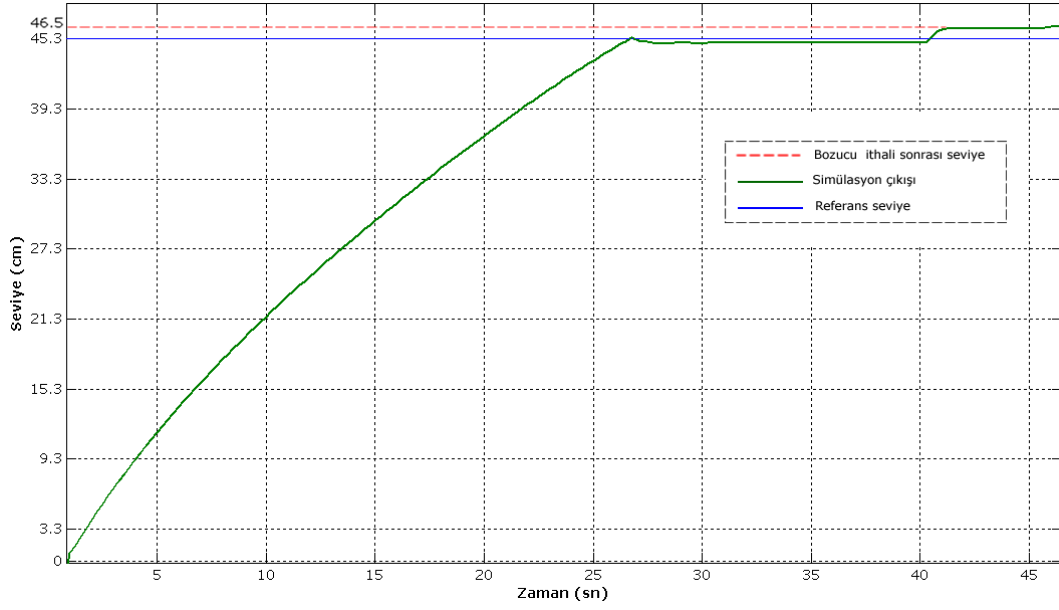
Şekil V.14'de parametre optimizasyonunda kullanılan referans eğrisi görülmektedir. Burada, 1350. saniyeye kadar simülasyon referansı, 1350. saniyeden sonra ise gerçek sistem bozuculu çıkışı verilmiştir. Parametre optimizasyonunda bozucu ithali kullanılarak elde edilen değerler ile sistem gerçek zamanlı olarak çalıştırılmıştır. Sisteme bozucu ithal edilmeden simülasyonda verilen referans seviye takibinde elde edilen değerler ile de sistem gerçek zamanlı olarak çalıştırılmış ve her iki çıkış karşılaştırılmıştır.

Şekil V.15’de görüldüğü gibi bozucu ithal edilerek elde edilen optimizasyon parametreleri çıkışı, bozucusuz referans ile elde edilen parametre çıkışına göre daha iyi sonuç vermiştir.. Her iki çıkış eğrisinden de görüldüğü gibi sisteme bozucu ithal edildiğinde sistemde kalıcı hal hatası görülmektedir. Bu sistemin tahliye vanasının %100 açık modellenmesinden kaynaklanmaktadır. Sistem %50 seviyesine kadar bozucu etkisini minimize etmekte fakat bu noktadan sonra kalıcı hal hatası görülmektedir.



Şekil V.15 Farklı referans girişler ile optimize edilen parametrelerin sistem çıkışı

Bozucu ithali sonrası sistemde görülen kalıcı hal hatasının sistemden mi, elde edilen parametrelerden mi kaynaklandığını tespit etmek için sistem %70 bozucu ithali ile tekrar modellenmiş ve elde edilen yeni model, bozucusuz model ile elde edilen denetleyici parametreleri ile çalıştırılmıştır. Şekil V.16’da görüldüğü gibi bozucu ile modellenen sistem bozucusuz modellenen sistem ile optimize edilerek elde edilen parametreler ile çalıştırılmış ve sistemde yine kalıcı hal hatası meydana gelmiştir. Dolayısıyla gerçek zamanlı çalışmada görülen kalıcı hal hatası beklenen bir sistem davranışdır.



Şekil V.16 %70 bozucu ithali sonrası sistem davranışı ve kalıcı hal hatası

Simülasyon üzerinde görülen kalıcı hal hatası gerçek zamanlı çalışmada görülen kalıcı hal hatası ile yaklaşık aynı değerlerdedir. (%70 bozucu ile simülasyonda 1.2cm, gerçek sistemde 1.35cm) Sistem %50'ye kadar olan bozucuları tolere edebilmiş ancak bu değerden sonraki bozucular ilk önce yüksek aşım daha sonra azalarak kalıcı hal hatasına dönüşmektedir.

BÖLÜM VI

GUNT RT 512 SİSTEMİ İLE YAPILAN DENEYLER

VI.1. GİRİŞ

Bu bölümde, GuntRT 512 su seviye kontrol sisteminin YSA ile elde edilen modeli kullanılarak bulanık denetleyiciye ait parametrelerin optimizasyonu sonucu elde edilen değerlerin gerçek zamanlı sistem uygulamaları yapılacaktır. Bunun öncesinde örnekleme zamanındaki değişimin denetleyici davranışı üzerindeki etkisi incelenmiş ve her iki optimizasyon algoritması için de karşılaştırmalı performansları ortaya konmuştur. Daha sonra Genetik ve KKO algoritmaları ile elde edilen bulanık denetleyiciye ait üyelik fonksiyonu sınır değerleri ve aksiyon değerleri farklı referanslar üzerindeki davranışları incelenmiştir. İlk olarak basamak girişler ve basamak geçişler arasındaki davranışı incelenen denetleyicinin daha sonra da yörünge ve bozuculara karşı gösterdiği davranış ele alınmıştır.

VI.2. ÖRNEKLEME ZAMANI DENEYLERİ

Örnekleme zamanı gerek denetleyicinin performansı gerek sistem davranışı için önemlidir. Kullanılan sıvı seviye sisteminde örnekleme zamanının etkisini görmek için farklı sıklıktaki örnekler 45cm'lik referans seviye ile birlikte kullanılmıştır. Bu deneyler sırasıyla 0.01sn, 0.1sn, 0.25sn, 0.5sn, 1sn ve 2sn olmak üzere 6 farklı örnekleme zamanı ile çalıştırılmıştır. Elde edilen çıkış eğrilerinin referans yörünge ile arasındaki fark RMS hatası olarak elde edilmiş, ilgili veriler Tablo VI.1'de verilmiştir. Tablo VI.1'den görüleceği üzere örnekleme sıklığı azaldıkça RMS değeri de, kontrol işaretinin daha uzun aralıklarla sisteme gönderilmesi sebebiyle artmaktadır. Benzer şekilde örnekleme sıklığının artması ile sisteme gönderilen kontrol işareti sıklığı da artmaktadır. Ancak tüm veriler içerisinde RMS değerinin en düşük olduğu örnekleme zamanı değeri 0.25sn'dir. Bu, eğimi 10 olan referans yörünge için de basamak giriş olarak verilen 45 cm seviyesi için elde edilen değerler için de geçerlidir. Burada dikkat edilmesi gereken bir nokta da

kontrol işaretlerinde örnekleme sıklığı ile görülen değişimlerdir. Örnekleme sıklığının azalması sistemin gerekli kontrol işaretini zamanında sisteme göndermesini de geciktirmekte ve referansı izlemesini zorlaştırmaktadır.

Tablo VI.12 Farklı örnekleme zamanları için sistem çıkışı RMS değerleri

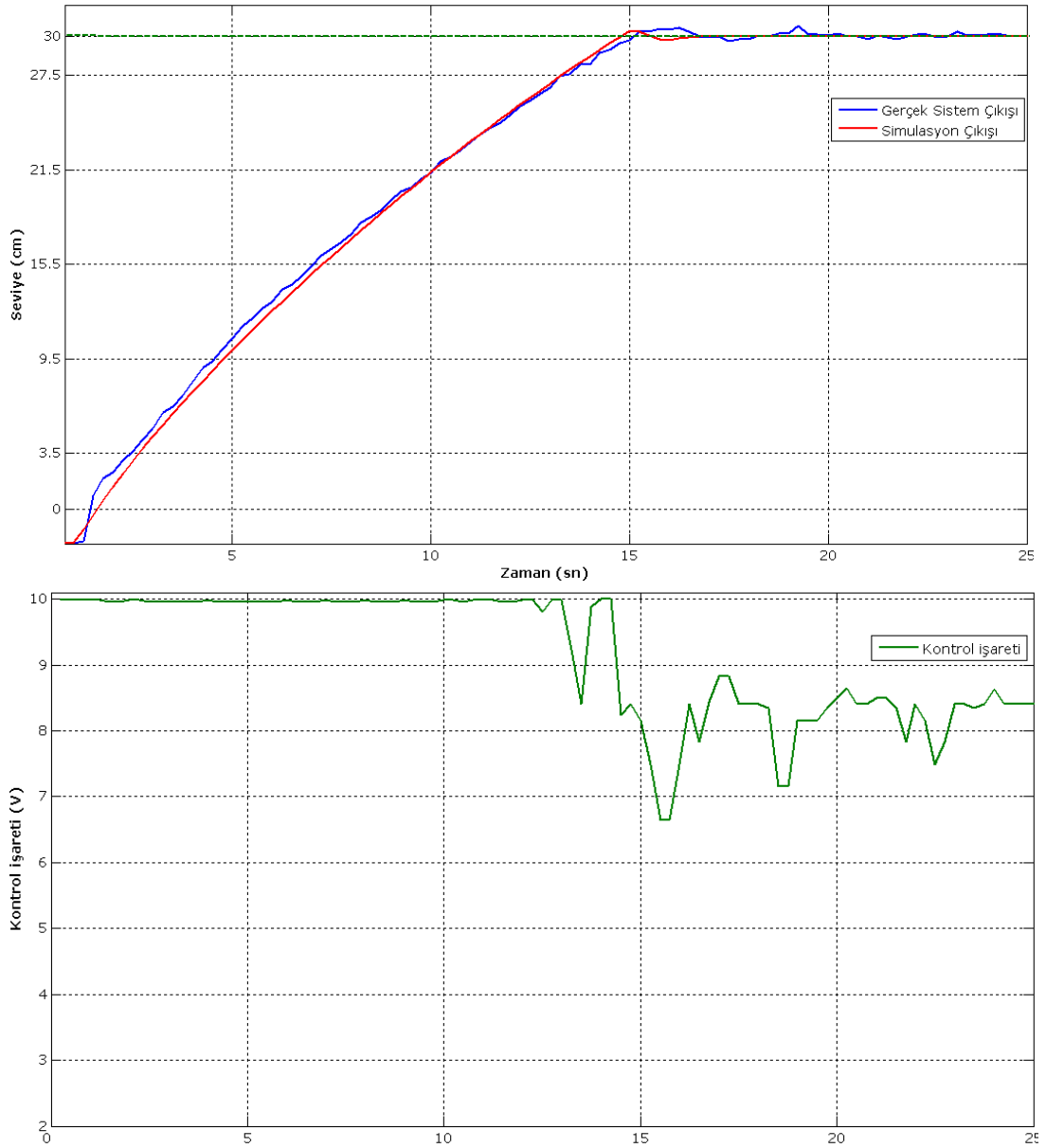
Örnekleme zamanı	Genetik		KKO	
	Yörünge	Basamak Giriş (45cm)	Yörünge	Basamak Giriş (45cm)
0.01	0.0133	0.1757	0.0072	0.1735
0.1	0.0065	0.1755	0.0059	0.1725
0.25	0.0058	0.1746	0.0057	0.1721
0.5	0.0071	0.1776	0.0064	0.1765
1	0.0126	0.1821	0.0118	0.1842
2	0.0231	0.1964	0.0208	0.1949

VI.3. KKO - BULANIK DENETLEYİCİ DENEYLERİ

Bu bölümde KKO algoritması ile optimize edilen bulanık denetleyici parametreleri ile çalıştırılan sisteme gerçek zamanlı olarak sırasıyla basamak girişler, basamak girişler arası geçişler ve farklı eğimlerdeki yörüngeler referans olarak verilmiş sistem davranışı, bozucular ithal edilerek de denetleyici davranışı ortaya konmuştur.

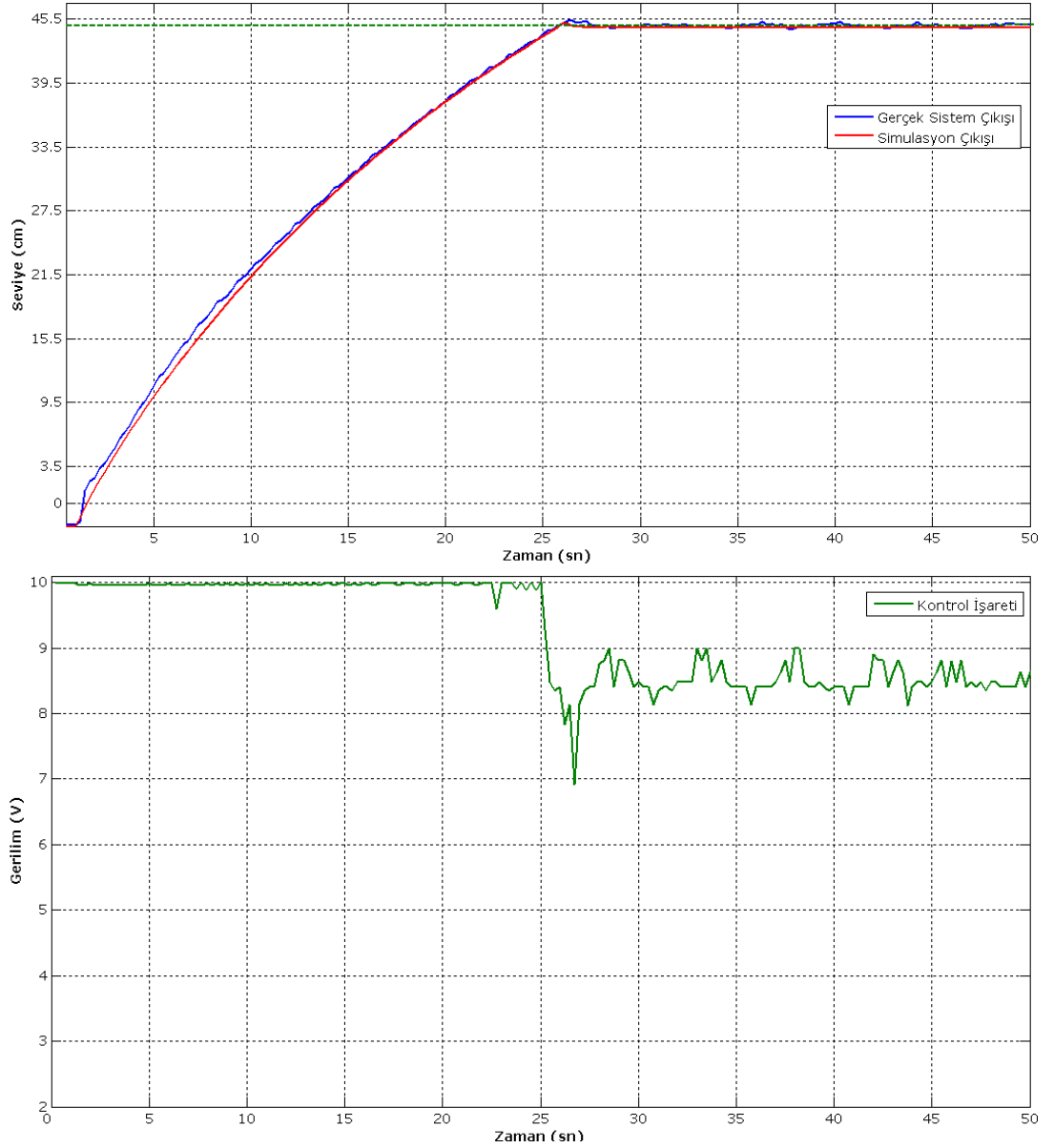
VI.3.1. Sistemin Basamak Giriş Cevapları

Sisteme sırasıyla 30 cm, 45cm ve 60 cm referans seviyeleri, boşaltım vanası tam açıklıkta iken verilmiştir. Bu referanslar için simülasyon ve gerçek sistem çıkışları ile bu çıkışları sağlayan kontrol işaretleri Şekil VI.1, Şekil VI.2 ve Şekil VI.3’de gösterilmiştir. Kontrol işaretleri davranışlarından da anlaşılabacağı gibi sistemin çıkışı referans değerlere yaklaştıkça, denetleyici işareti maksimum seviyeden kararlı durum seviyesine doğru gelir.



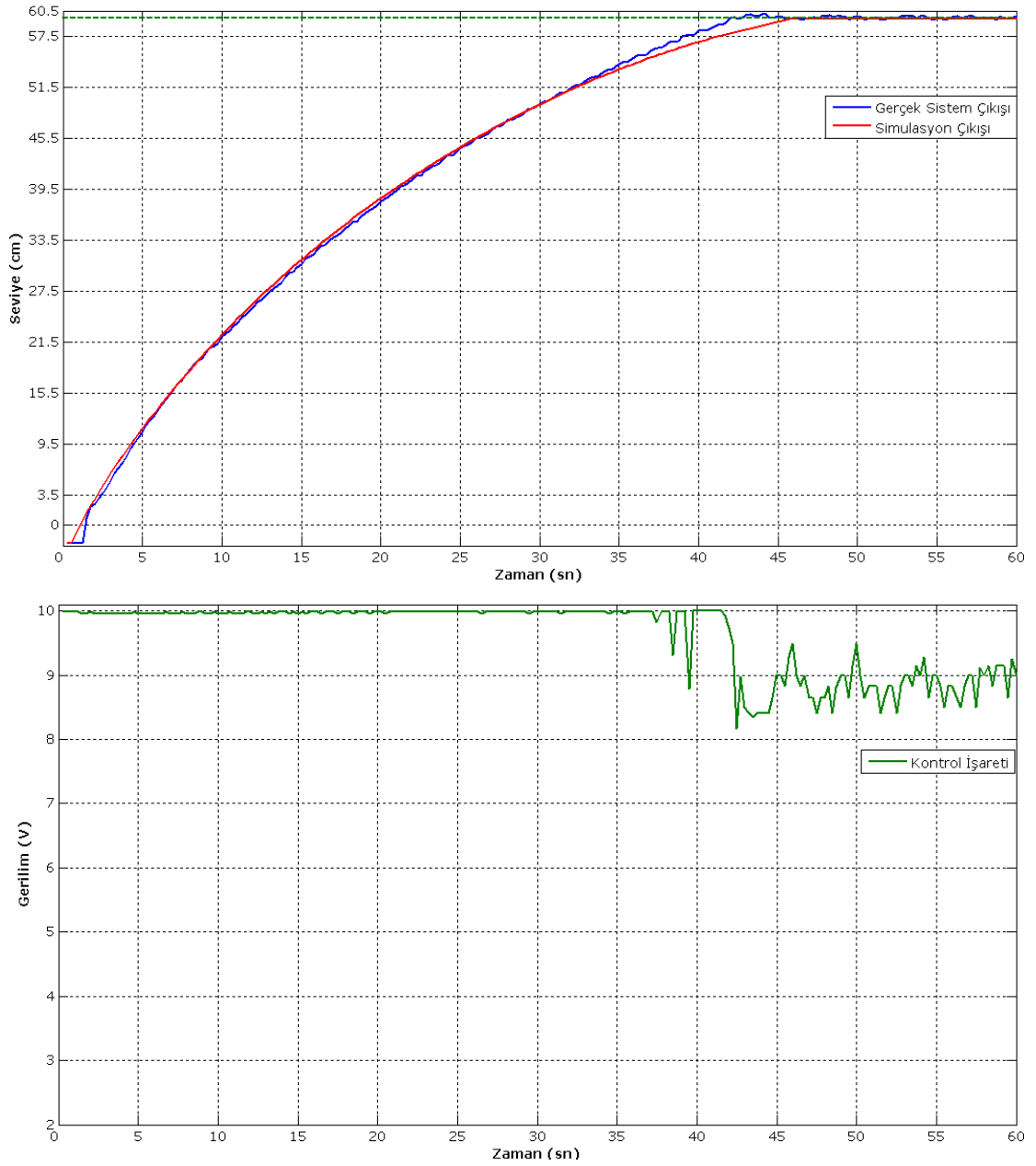
Şekil VI.29 30 cm referans için gerçek sistem ile simülasyon çıkışı ve kontrol işareti

30 cm referans seviyesi için verilen çıkış grafiğinde de görüleceği gibi simülasyon çıkışı ile gerçek sistem çıkışı birbirlerine oldukça yakın davranış göstermektedir. Sıvı seviye tüpünde minimum seviye tüp boş iken -2 cm civarındadır. Bu, simülasyon ve gerçek sistem çıkışı grafiklerinde de görüleceği gibi sıvı seviyesi 0 cm'nin altından başlamaktadır. Denetleyici, referansa gelinceye kadar 10V gönderirken yaklaşık 0.57 cm aşım yapmıştır. Referansa yaklaştıkça denetleyici çıkışı yaklaşık 8.4V ile dengelemeye çalışmış ve seviyeyi 30cm'de tutmuştur. Bu noktadan sonra kalıcı hal hatası ise yaklaşık 0.08 cm civarındadır.



Şekil VI.2 45 cm referans için gerçek sistem ile simülasyon çıkışı ve kontrol işareti

45 cm referans seviyesi için verilen çıkış grafiğinde de görüleceği gibi simülasyon çıkışı ile gerçek sistem çıkışı üst üste oturmaktadır. Kontrol işaretinden de görüleceği üzere referans seviyeye gelinceye kadar denetleyici tam sürme işareti, 10V gönderirken yaklaşık 0.5 cm civarında aşım yapmış, 28. saniyeden itibaren sistemin referansa oturmasını sağlamıştır. Bu noktadan sonra kalıcı hal hatası ise yaklaşık 0.06 cm civarında kalarak oldukça başarılı bir seviye takibi gerçekleştirmiştir. Şekil VI.3’de 60 cm için verilen referansta da sistem simülasyon cevabından yaklaşık 2sn önce referansa ulaşarak 0.4 cm aşım yapmış, 9V’luk kontrol işareti, 0.03 cm’lik kalıcı hal hatası ile 43. saniyeden itibaren referans seviyeye oturmuştur.



Şekil VI.3 60 cm referans için gerçek sistem, simülasyon çıkışı ve kontrol işareti

VI.3.2. Basamak Girişler Arası Geçişlerde Sistem Davranışı

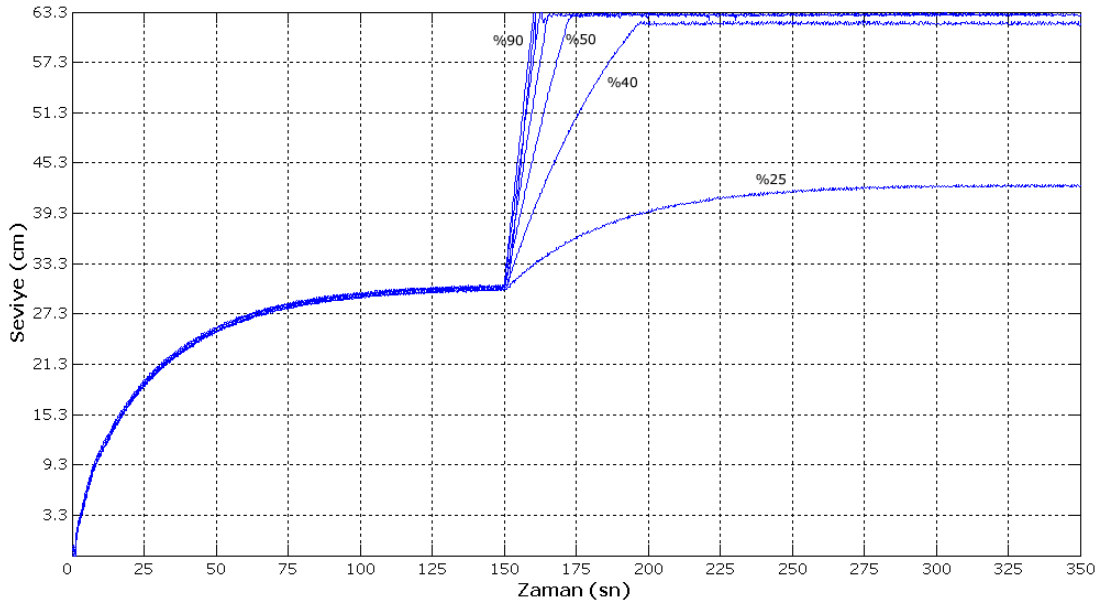
Sisteme çalışma anında farklı referans değerleri verilerek denetleyici cevabı incelenmiştir. Çalışma sırasında sistem, ilk referans değerine oturduktan sonra referans değerini değiştirerek sistemin referanslar arasında geçişi incelenmiştir. Sisteme uygulanan referans giriş Şekil VI.4’de görülmektedir.



Şekil VI.4 Basamak geçişler sırasında sistem davranışı ve kontrol işareti değişimi

VI.3.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı

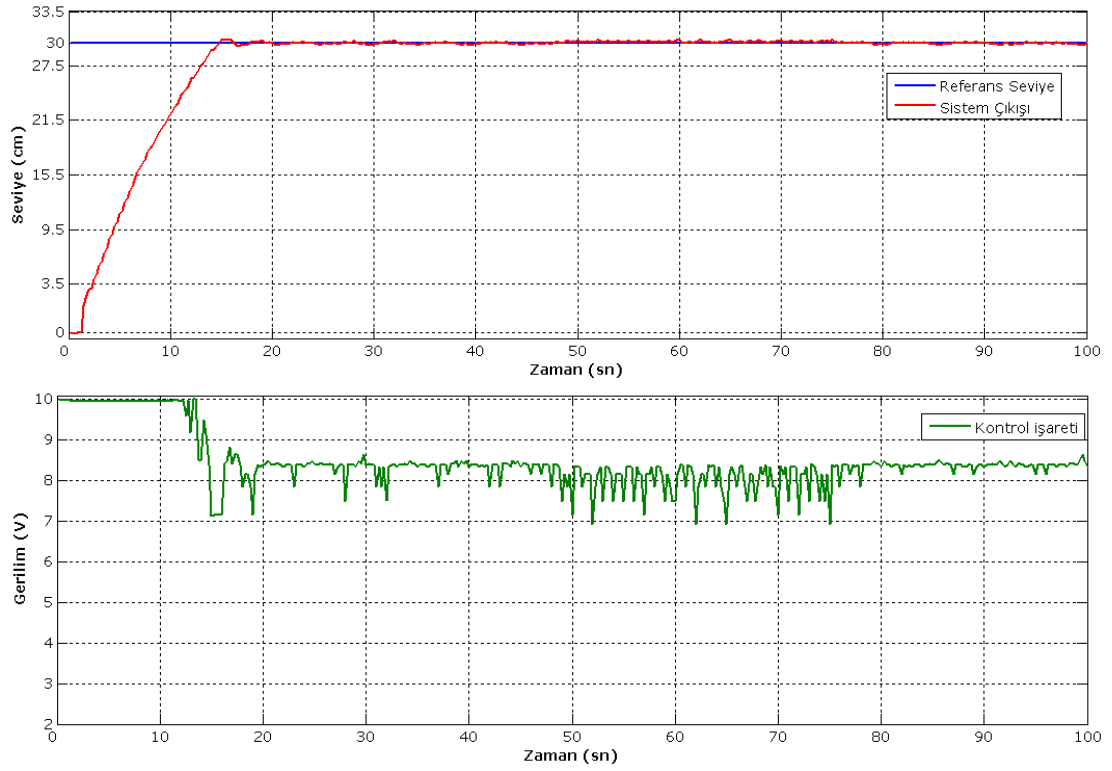
GuntRT 512 su seviye sistemine bozucu ithali tahliye vanası açıklığı değiştirilerek yapılmaktadır. Bu tahliye vanası yüzdelik dilimlere ayrılarak %0 - %90 aralığında kullanılmaktadır. %0, tahliye vanasının tamamen açık, %90 ise tamamen kapalı olduğunu göstermektedir. Dolayısıyla sisteme bozucu ithali, tahliye vanasının açıklık değeri azaltılarak veya artırılarak sağlanmaktadır. Tahliye vanası sistem modellenirken tam açıklıkta (%0) olduğu için bunun dışındaki tüm açıklıklar sisteme bozucu etkisi gösterir. Bozucular, sistem 30 cm seviyesinde oturmuş iken sırasıyla %25, %40, %50, %60, %70, %80 ve %90 olarak verilmiş ve kontrol altındaki sistem davranışı incelenmiştir. Sistemin kapalı çevrim davranışı izlenmeden önce açık çevrim davranışı da incelenmiştir. Sıvı seviye sistemi 30 cm referansa oturmuş iken %25 - %80 arasındaki bozucular denetleyici kullanılmadığı zaman bozucuların sisteme etkisini görmek için verilmiştir.



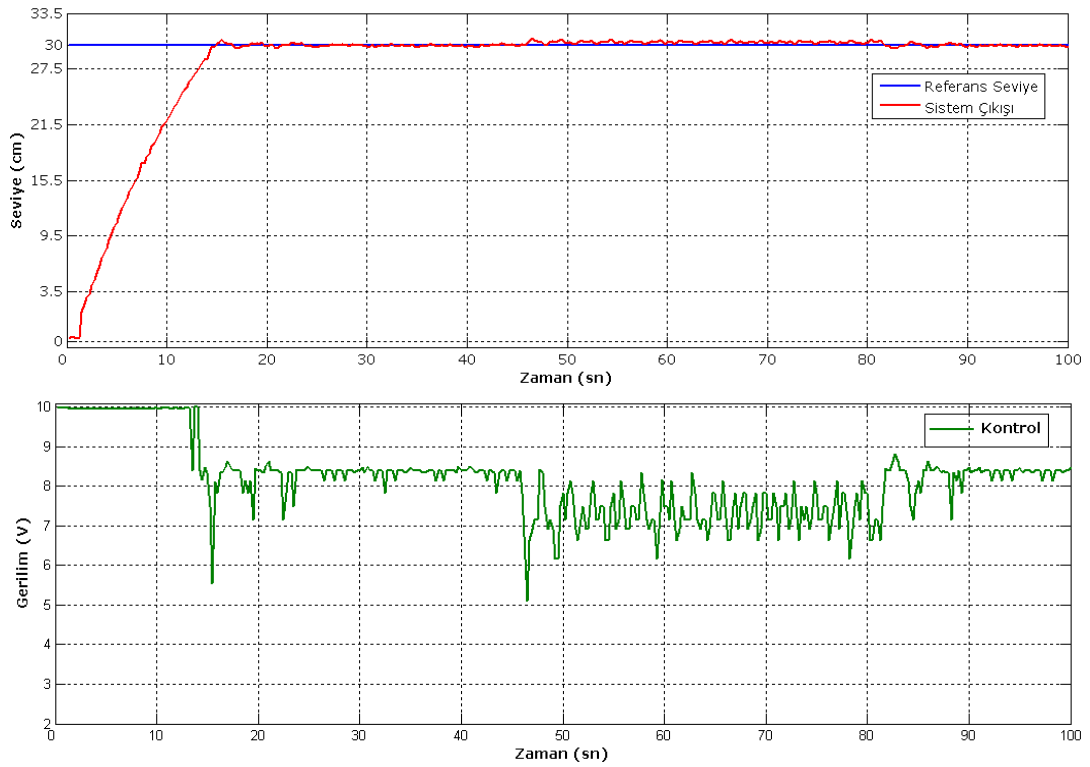
Şekil VI.5 Açık çevrimde sistemin bozuculara cevabı

Şekil VI.6'da sistemin çıkışındaki vana 50. saniyeden itibaren %25 oranında kapatılarak sistemin çıkışındaki sıvı miktarı azaltılmış ve tahliye işleminin daha yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici kontrol işaretini azaltarak sıvı seviyesinin referansta kalmasını sağlamıştır. 75. saniyede tahliye vanası tam açıklığa (%0) getirilerek bozucu kaldırılmıştır. Bu andan itibaren kontrol işareti de artmış ve seviyenin referansa oturmasını sağlamıştır. 25 saniyelik bozucu

süresince seviyede yaklaşık 0.1 cm'lik sapma olmuştur. Ancak bu hata için tanımlanan “Sıfır” üyelik fonksiyonu sınırları içerisindedir.

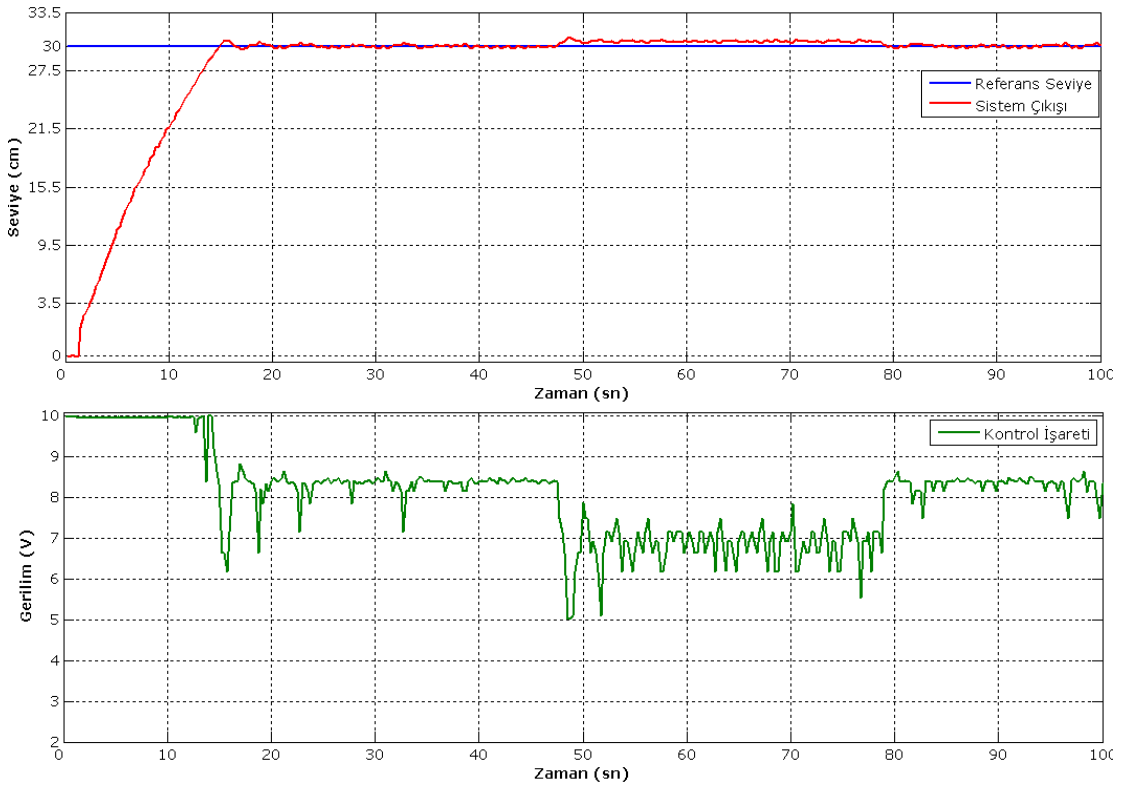


Şekil VI.6 %25 bozucu uygulanan sistemin cevabı ve kontrol işareti



Şekil VI.7 %40 bozucu uygulanan sistemin cevabı ve kontrol işareti

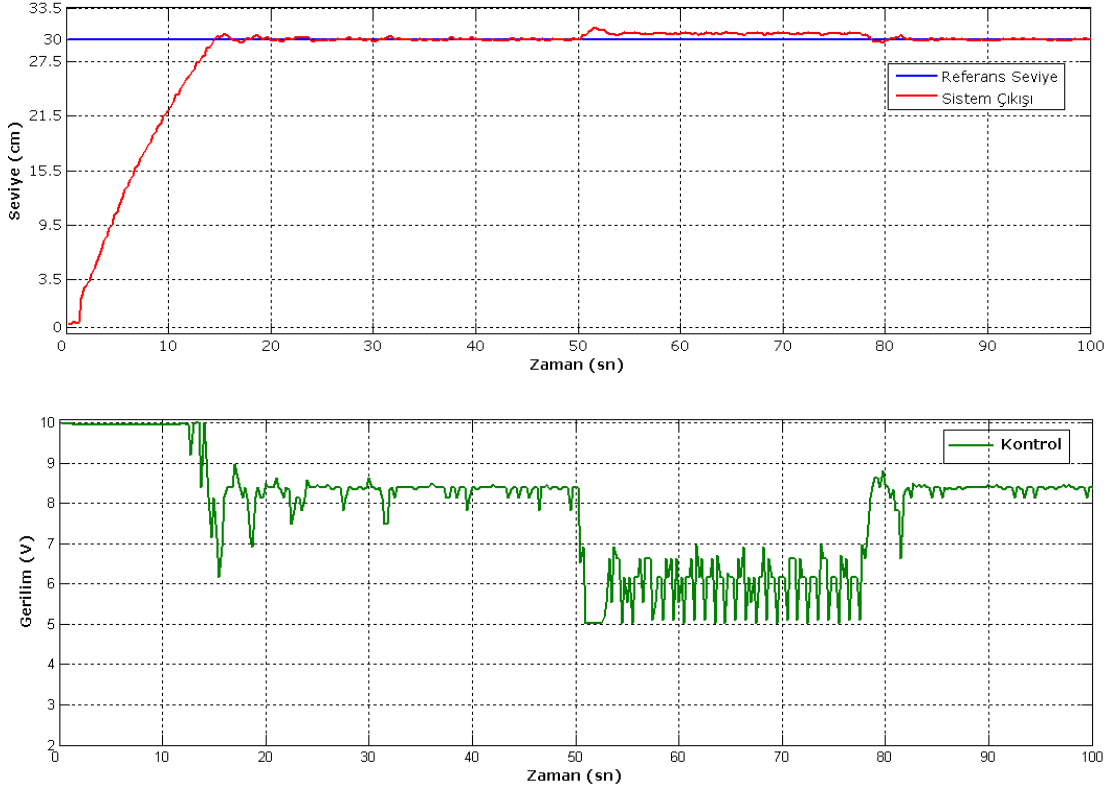
Şekil VI.7’de, tahliye vanası 47. saniyeden itibaren %40 oranında kapatılarak sistemin çıkışındaki sıvı miktarı azaltılmış ve tahliye işleminin daha da yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici, kontrol işaretini azaltarak su seviyesinin referansı takip etmesini sağlamıştır. 81. saniyede kaldırılan bozucu ile tahliye işlemi tam açıklıktaki vana üzerinden yapılmış, kontrol işareti de kademeli olarak artmış ve seviyenin referans ile birlikte hareket etmesini sağlamıştır. 34 saniyelik bozucu süresince seviyede yaklaşık 0.22 cm’lik bir sapma olmuştur. Ancak bu fark da, üyelik fonksiyonlarından “Sıfır” bölgesi sınır değerleri içerisinde olduğu için sistem tarafından hata olarak algılanmamakta, tanımlanmış bölge içerisinde yer almaktadır.



Şekil VI.8 %50 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.8’de, tahliye vanası 48. saniyeden itibaren %50 oranında kapatılarak sistemin çıkışındaki sıvı miktarı azaltılmış ve tahliye işleminin daha da yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici, kontrol işaretini azaltarak su seviyesinin referansı takip etmesini sağlamıştır. 78. saniyede kaldırılan bozucu ile tahliye işlemi tam açıklıktaki vana üzerinden yapılmış, kontrol işareti de artmış ve seviyenin referans ile birlikte hareket etmesini sağlamıştır. 30 saniyelik bozucu süresince seviyede yaklaşık 0.4 cm’lik bir sapma olmuştur. Bu sapma değeri de diğer

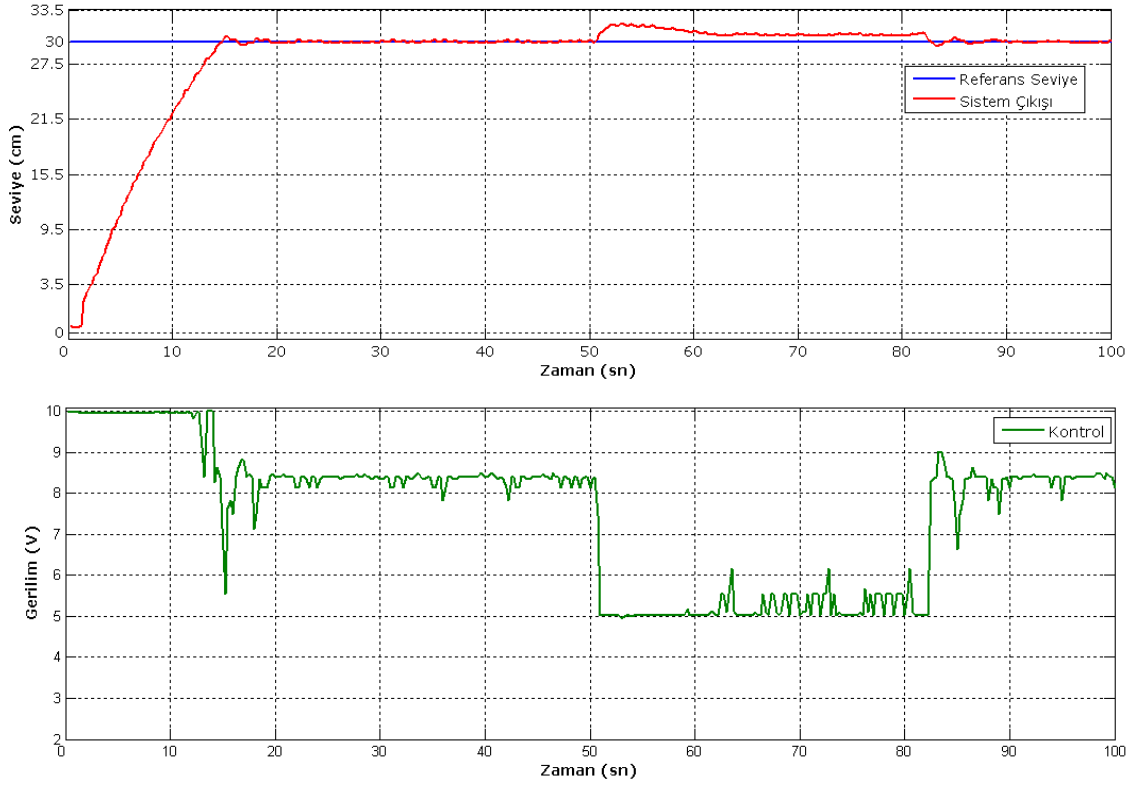
bozucularda olduğu gibi hata'nın üyelik fonksiyonlarından “Sıfır” hata bölgesi sınırları içerisinde kalmıştır.



Şekil VI.9 %60 bozucu uygulanan sistemin cevabı ve kontrol işareti

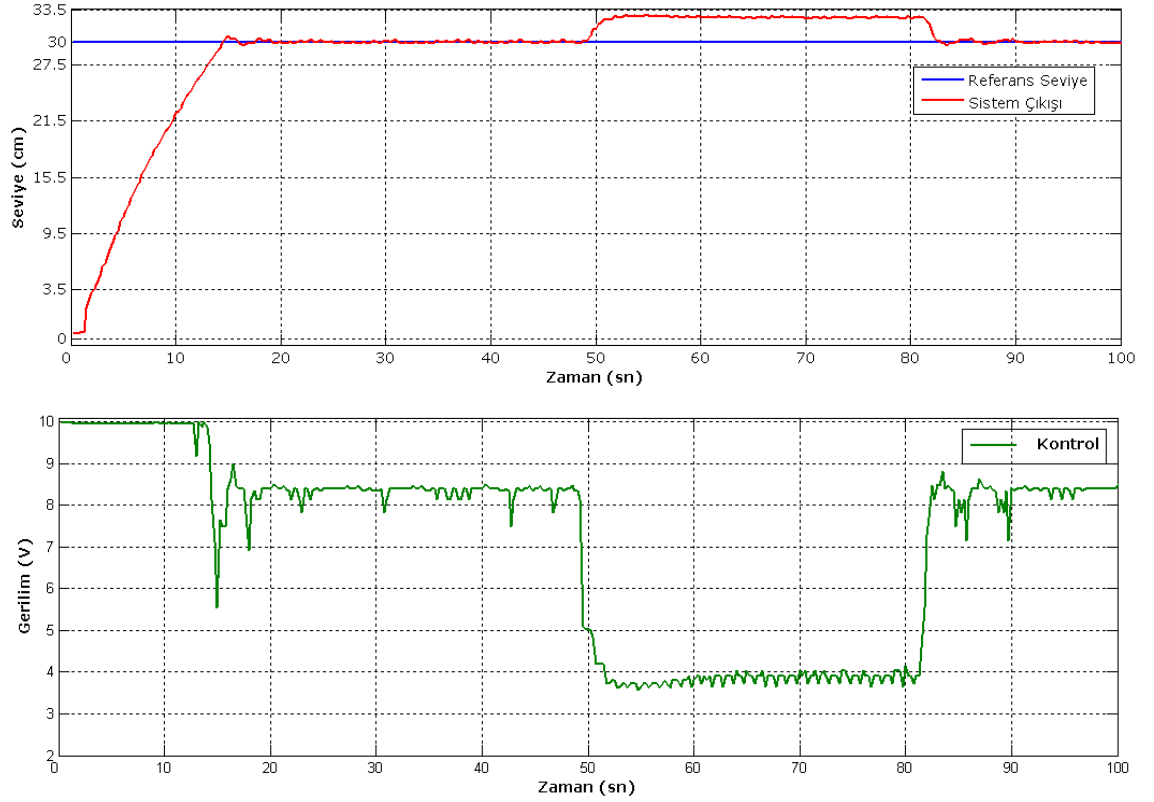
Şekil VI.9’da, tahliye vanası 50. saniyeden itibaren %60 oranında kapatılarak sistemin çıkışındaki sıvı miktarı azaltılmış ve tahliye işleminin daha da yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici, kontrol işaretini azaltarak su seviyesinin referansı takip etmesini sağlamıştır. 77. saniyede kaldırılan bozucu ile tahliye işlemi tam açıklıktaki vana üzerinden yapılmış, kontrol işareti de kademeli olarak artmış ve seviyenin referans ile birlikte hareket etmesini sağlamıştır. 27 saniyelik bozucu süresince seviyede yaklaşık 0.5 cm’lik bir sapma olmuştur. Bu sapma değeri de benzer şekilde giriş değişkenlerinden hata'nın üyelik fonksiyonlarından “Sıfır” hata bölgesi sınırları içerisinde kalmış ve denetleyici tarafından hata kabul edilmeyen sınırlar içerisinde tutulmuştur.

Üyelik fonksiyonlarının sayılarının arttırılması ve aralıklarının azaltılması ile bu hata değerinin daha düşük seviyelerde tutulması da sağlanabilir. Ancak sistem için 0.5 cm’lik sapma kabul edilebilir bir sapma değeridir ve kullanılan üyelik fonksiyonları bu sınırlar için yeterlidir.



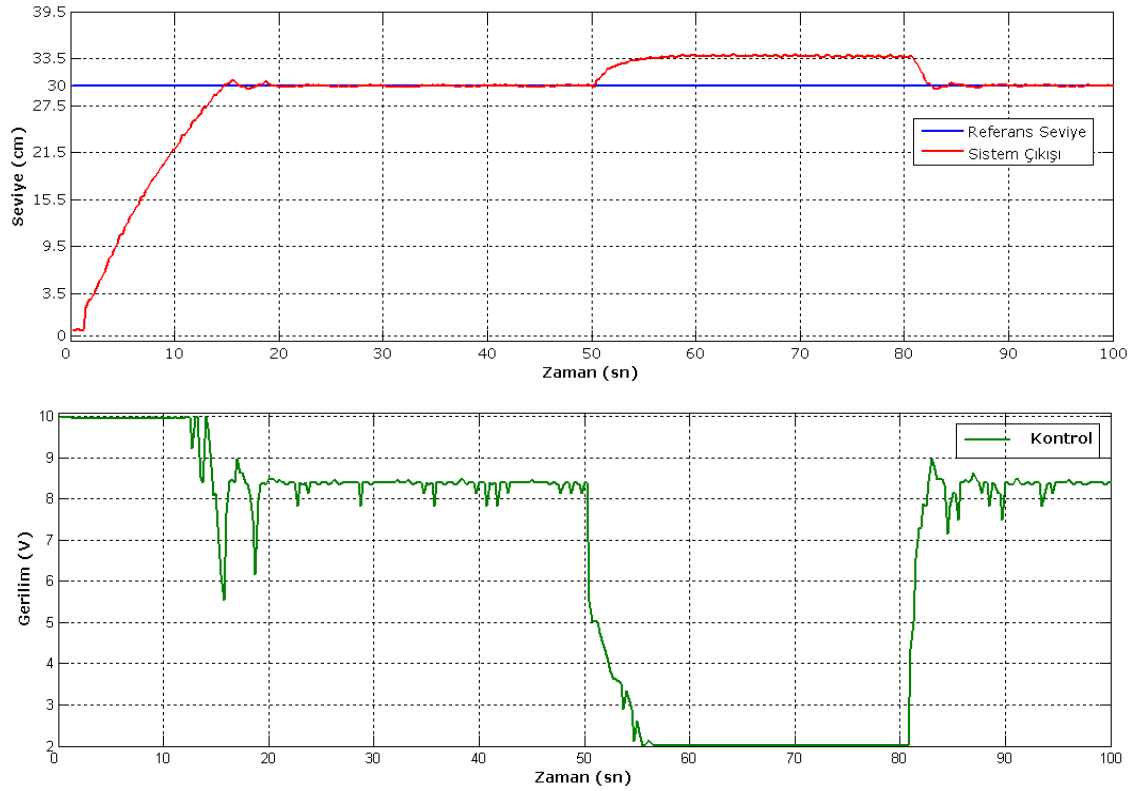
Şekil VI.10 % 70 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.10’da, tahliye vanası 51. saniyeden itibaren %70 oranında kapatılarak sistemin çıkışındaki sıvı miktarı azaltılmış ve tahliye işleminin daha da yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici, kontrol işaretini azaltarak su seviyesinin referansı takip etmesini sağlamıştır. 83. saniyede kaldırılan bozucu ile tahliye işlemi tam açıklıktaki vana üzerinden yapılmış, kontrol işareti de kademeli olarak artmış ve seviyenin referans ile birlikte hareket etmesini sağlamıştır. 32 saniyelik bozucu süresince seviyede yaklaşık 0.85 cm’lik bir sapma olmuştur. Bu sapma değeri de giriş değişkenlerinden hata’nın “Sıfır” ve “pozitif küçük” olarak tanımlanan aralıkta yer almaktadır.



Şekil VI.11 % 80 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.11’de, tahliye vanası 48. saniyeden itibaren %80 oranında kapatılarak sistemin çıkışındaki sıvı miktarı azaltılmış ve tahliye işleminin daha da yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici, kontrol işaretini 3.5V seviyelerine kadar azaltarak su seviyesinin referansı takip etmesini sağlamıştır. 83. saniyede kaldırılan bozucu ile tahliye işlemi tam açıklıktaki vana üzerinden yapılmış, kontrol işareti de kademeli olarak artmış ve seviyenin referans ile birlikte hareket etmesini sağlamıştır. 35 saniyelik bozucu süresince seviyede yaklaşık 2.5 cm’lik bir sapma olmuştur. Bu sapma değeri de giriş değişkenlerinden hata’nın “pozitif küçük” ve “pozitif büyük” olarak tanımlanan aralıkta yer almaktadır.

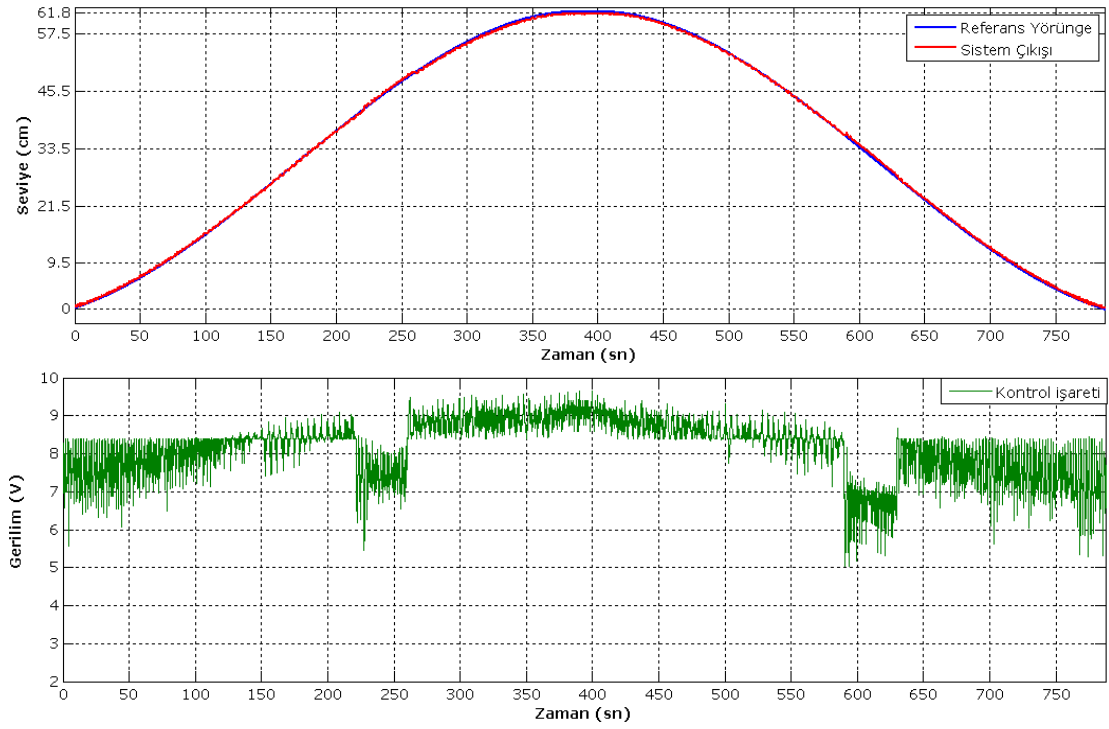


Şekil VI.12 % 90 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.12’de sistemin çıkışındaki vana 50. saniyeden itibaren %90 oranında kapatılarak sistemin çıkışındaki sıvı miktarı daha da azaltılmış ve tahliye işleminin daha yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici kontrol işaretini 2 seviyesine kadar düşürerek seviyenin referans ile birlikte hareket etmesini sağlamaya çalışmış ancak yaklaşık 3.5 cm’lik sapmayı ortadan kaldıramamıştır. Bu sapma değeri de giriş değişkenlerinden hata’nın “pozitif büyük” olarak tanımlanan aralıkta yer aldığı için minimum kontrol işareti ile sapma değeri tanımlanan aralık için minimum değerde tutulmuştur. 81. saniyede kaldırılan bozucu ile tahliye işlemi tam açıklıktaki vanadan yapılmış, kontrol işareti de kademeli olarak artmış, seviyenin referansı izlemesini sağlamıştır.

V.1.4. KKO – Bulanık Denetleyicinin Yörünge Takip Cevapları

Burada, sistem girişine Denklem II.1’de tanımlanan 3. derece bir fonksiyon 2 parçalı referans olarak verilmiştir. Çıkış eğrisi için kullanılan bu fonksiyon daha sonra simetriği alınarak sistemin iniş fonksiyonu olarak kullanılmıştır. Bu sayede su seviyesinin 0 cm’den başlayarak 61.8cm’ye kadar çıkması ve tekrar 0 cm’ye inmesi sağlamıştır.



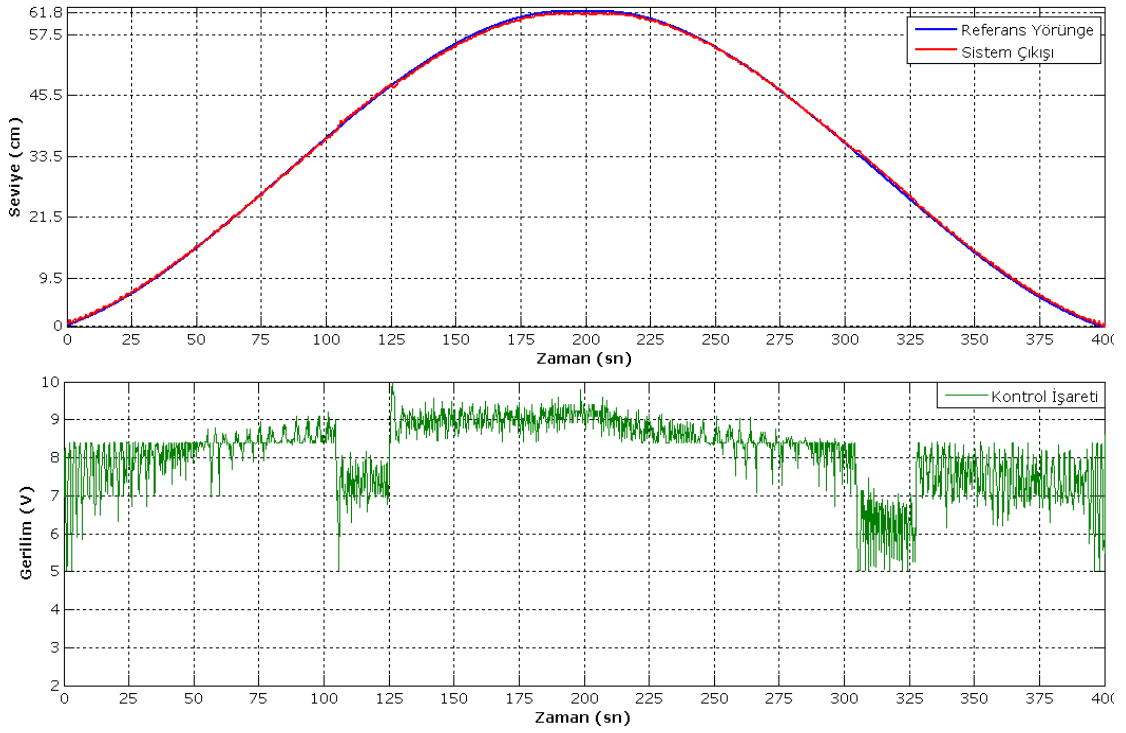
Şekil VI.13 Yörünge katsayısı 20 olan eğri için sistem cevabı ve kontrol işareti

$$V_y = -0.0000267 \cdot t^3 + 0.00173 \cdot t^2 + 0.21; \quad (VI.1)$$

$$V_y = -0.000214 \cdot t^3 + 0.00692 \cdot t^2 + 0.21; \quad (VI.2)$$

$$V_y = -0.026721 \cdot t^3 + 0.17311 \cdot t^2 + 0.21; \quad (VI.3)$$

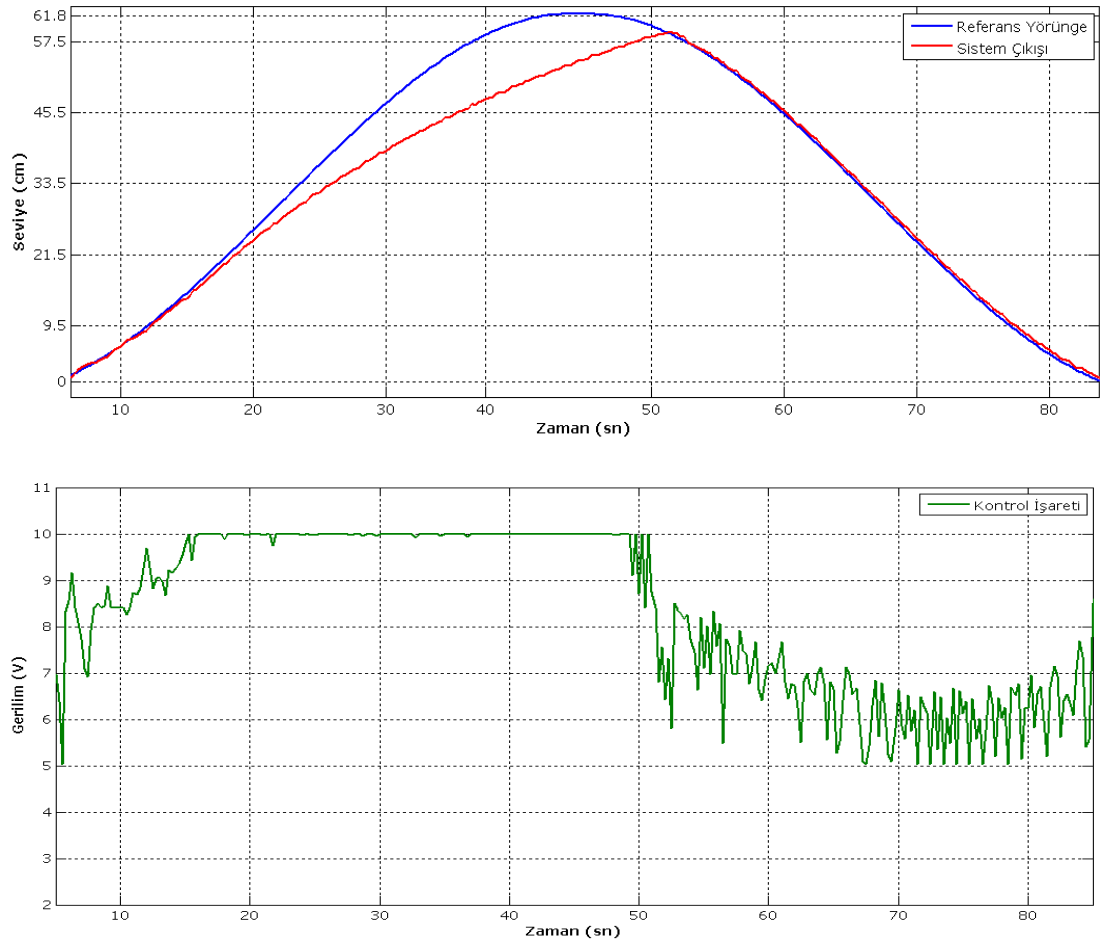
Şekil VI.13’de, denklem VI.1’de verilen ve yörünge katsayısı 20 olarak ayarlanmış eğri sisteme referans olarak verilmiştir. Sistem 61.8cm’ye geldiğinde ise aynı eğimle ilk konumuna gelmesi için gerekli yörünge oluşturularak sistemin boşalması sırasındaki cevabı incelenmiştir. Sistemin dolum ve boşalım sırasında yörüngeyi başarılı bir şekilde takip ettiği görülmüş ancak 10cm’nin altındaki referans seviyeleri için yüksek titreşimden kaynaklanan seviye değişimleri kontrol işaretindeki salınımlara da sebep olmuştur. Ancak, bu referans yörüngeyi takip etmeyi engellememiştir. Sisteme 215 ve 580. saniyelerde 50 saniye süresince %50 bozucular verilmesine rağmen sistem referans yörüngeyi izlemiştir.



Şekil VI.14 Yörünge katsayısı 10 olan eğri için sistem cevabı ve kontrol işareti

Şekil VI.14’de, denklem VI.2’de verilen ve yörünge katsayısı 10 olarak ayarlanmış eğri sisteme referans olarak verilmiştir. Sisteme 110. ve 310. saniyelerde 50 saniye süresince %50 bozucular verilerek denetleyici davranışı incelenmiştir. Sistem, bu referans yörüngeye %50 bozucular verilmesine karşın referansa oturmuştur.

Şekil VI.15’de, denklem VI.3’de verilen ve yörünge katsayısı 2 olarak ayarlanmış eğri sisteme referans olarak verilmiştir. Sistem referans yörüngeyi 15. saniyeye kadar takip etmesine rağmen bu saniyeden sonra takip edememiştir. 15. saniyeden sonra kontrol işareti en yüksek çıkışı olan 10V değerini vermesine karşın çıkışı istenen seviyeye getirememiştir. Bu, denetleyici performansından değil referans yörüngeyi eğiminden kaynaklanmaktadır. 52. saniyeden itibaren sistem boşalma eğrisinde yörüngeyi yakalamış ve inişe kadar referansı takip etmiştir. Burada sisteme verilecek referans yörüngeyi sistemin cevap verebileceği bir yörünge olması gerekliliği de ortaya çıkmaktadır. Aksi takdirde denetleyici, sistemi tam sürse dahi referans seviyeyi takip etmesi mümkün olmayabilir.



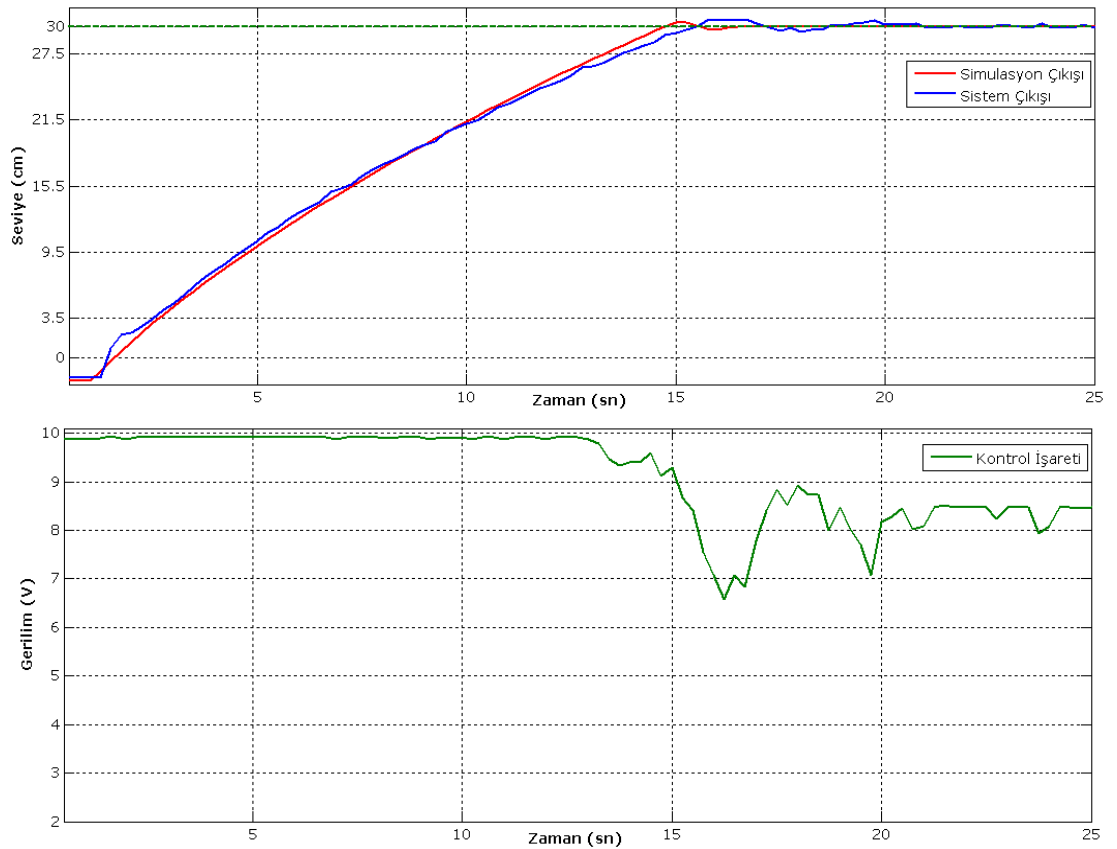
Şekil VI.15 Yörünge katsayısı 2 olan eğri için sistem cevabı ve kontrol işareti

VI.2. GENETİK - BULANIK DENETLEYİCİ DENEYLERİ

Bu bölümde Genetik algoritma ile optimize edilen bulanık denetleyici parametreleri ile GuntRT 512 sıvı seviye sistemi çalıştırılmıştır. Sisteme sırasıyla basamak girişler, basamak girişler arası geçişler ve farklı eğimlerdeki yörüngeler referans olarak verilmiştir. Daha sonra bozucu ithal edilerek sistemin davranışı incelenmiştir. Kapalı çevrim sistem cevabı ile simülasyon sonuçları da aynı referans seviyeleri ile birlikte verilerek karşılaştırılmıştır.

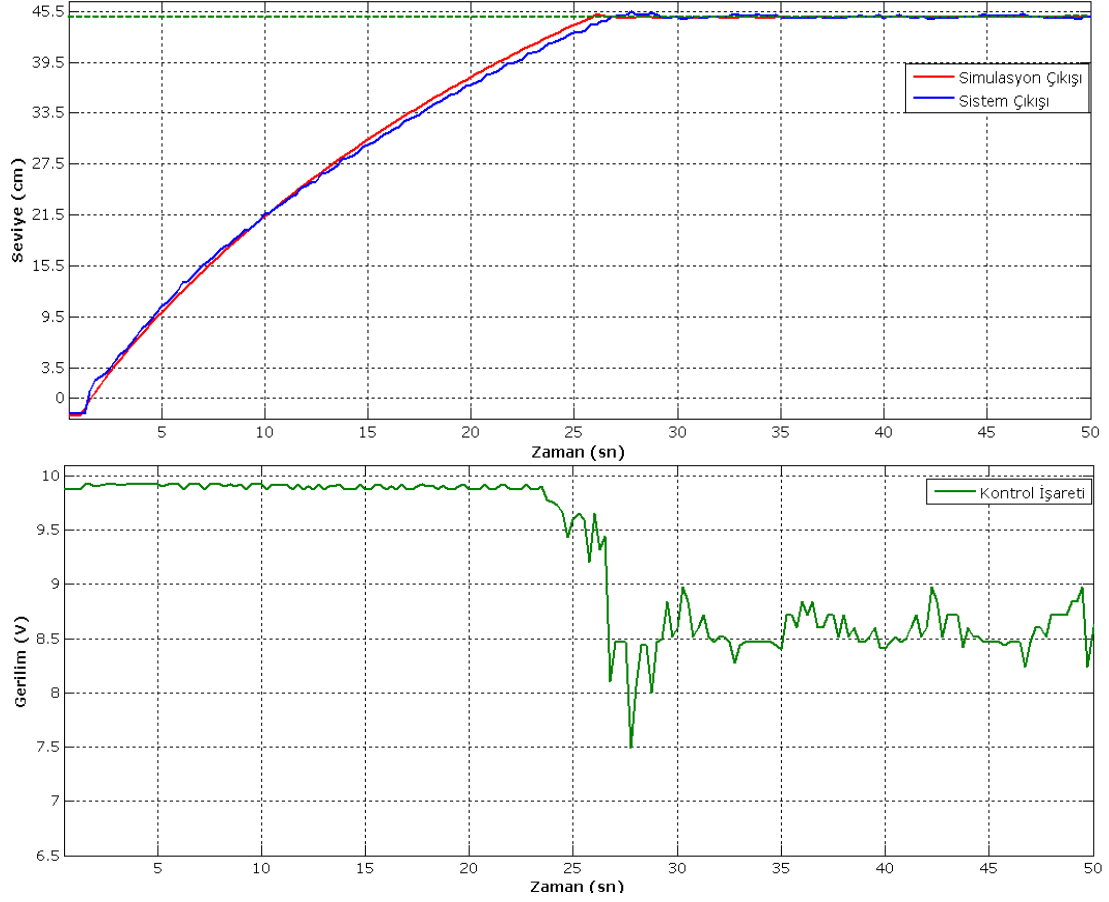
VI.2.1. Basamak Girişlere Sistemin Davranışı

Sıvı seviye sistemi ve simülasyon, 30cm, 45cm ve 60cm referans seviyeleri için %0 bozucu etkisi ile çalıştırılmıştır. Şekil VI.16'da 30 cm referans seviyesi için simülasyon, gerçek sistem çıkışları ve bu çıkışları sağlayan kontrol sinyalleri verilmiştir. Simülasyon çıkışı istenen seviyeye 15sn'de gelmesine karşın, gerçek sistem yaklaşık 15.7sn'de gelmektedir. Simülasyon çıkışında kararlı hal hatası olmamasına karşın gerçek sistem çıkışında 0.1cm'lik bir kararlı hal hatası oluşmuştur.



Şekil VI.16 30 cm referans için sistem, simülasyon çıkışı ve kontrol işareti

Kontrol işaretleri davranışlarından da anlaşılacağı gibi sistemin çıkışı referans değerlere yaklaştıkça, denetleyici işareti maksimum seviyeden kararlı durum seviyesine doğru gelir.

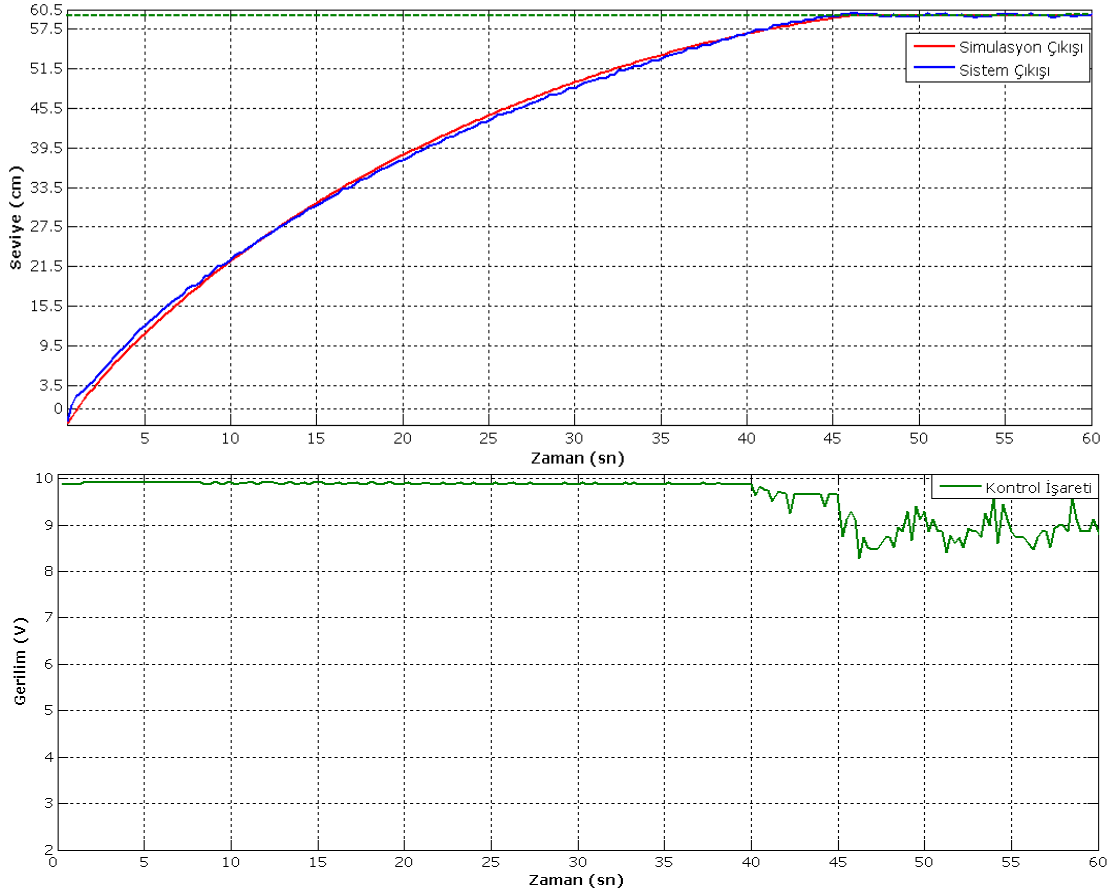


Şekil VI.17 45 cm referans için sistem, simülasyon çıkışı ve kontrol işareti

Şekil VI.17’de 45 cm referans seviyesi için verilen çıkış grafiğinde görüleceği gibi simülasyon çıkışı ile gerçek sistem çıkışı birbirlerine oldukça yakın davranış göstermektedir. Her iki sistem çıkışı da çok küçük bir aşım yaptıktan sonra referans seviyeye oturmakta ve bu seviyede kalmaktadır. Kontrol işaretlerinden de görüleceği üzere referans seviyeye gelinceye kadar denetleyici tam sürme işareti, 10V gönderirken 0.5 cm civarında aşım yapmıştır. Referans seviyeye yaklaştıkça kontrol işareti ortalama 8.6V civarında dengelenmiştir.

Şekil VI.18’deki 60 cm seviyesi için verilen çıkış grafiğinde simülasyon ile gerçek sistem oldukça yakın davranış göstermektedirler. 60 cm’ye gelinceye kadar

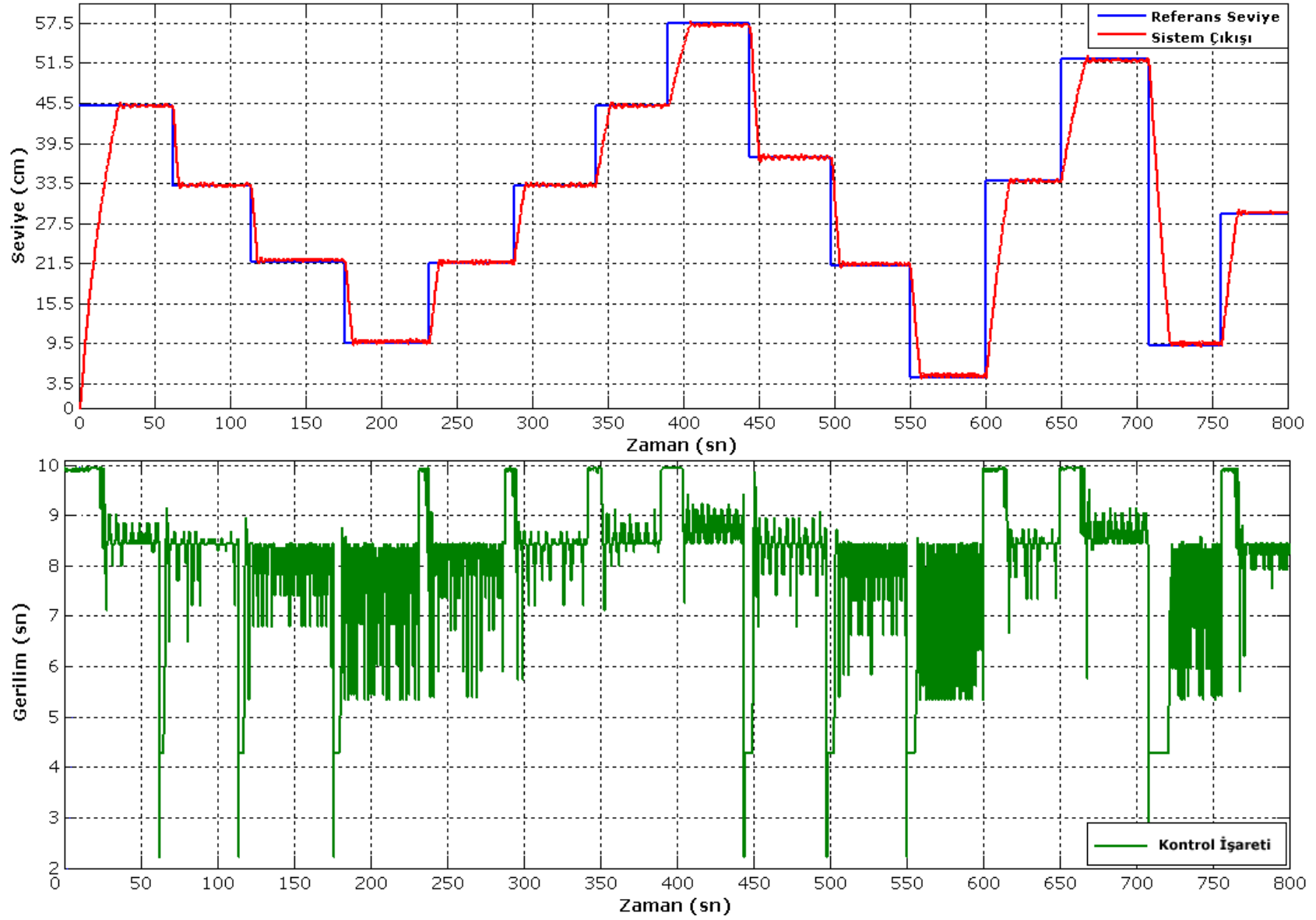
denetleyici tam sürme işareti, 10V gönderirken yaklaşık 0.2 cm civarında aşım yapmış, 40. saniyeden itibaren kontrol işaretini 8.2V'a kadar çekerek 46. saniyede sistemin referans seviyeye oturmasını sağlamıştır. Bu noktadan sonra kalıcı hal hatası ise yaklaşık 0.02 cm civarında kalmıştır.



Şekil VI.18 60 cm referans için sistem, simülasyon çıkışı ve kontrol işareti

VI.2.2. Basamak Girişler Arasında Geçiş Sistemin Davranışı

Sisteme çalışma anında farklı referans değerleri verilerek denetleyici cevabı incelenmiştir. Çalışma sırasında sistem, ilk değerine oturduktan sonra referans değerini değiştirerek sistemin referanslar arasında geçişi incelenmiştir. Şekil VI.19'da bu çalışmaya ait grafikler verilmiştir. Kontrolör istenen seviyeye çıkarken kontrol işaretini tam sürmekte, inerken ise minimum seviye olan 2V seviyesinde tutmaktadır.

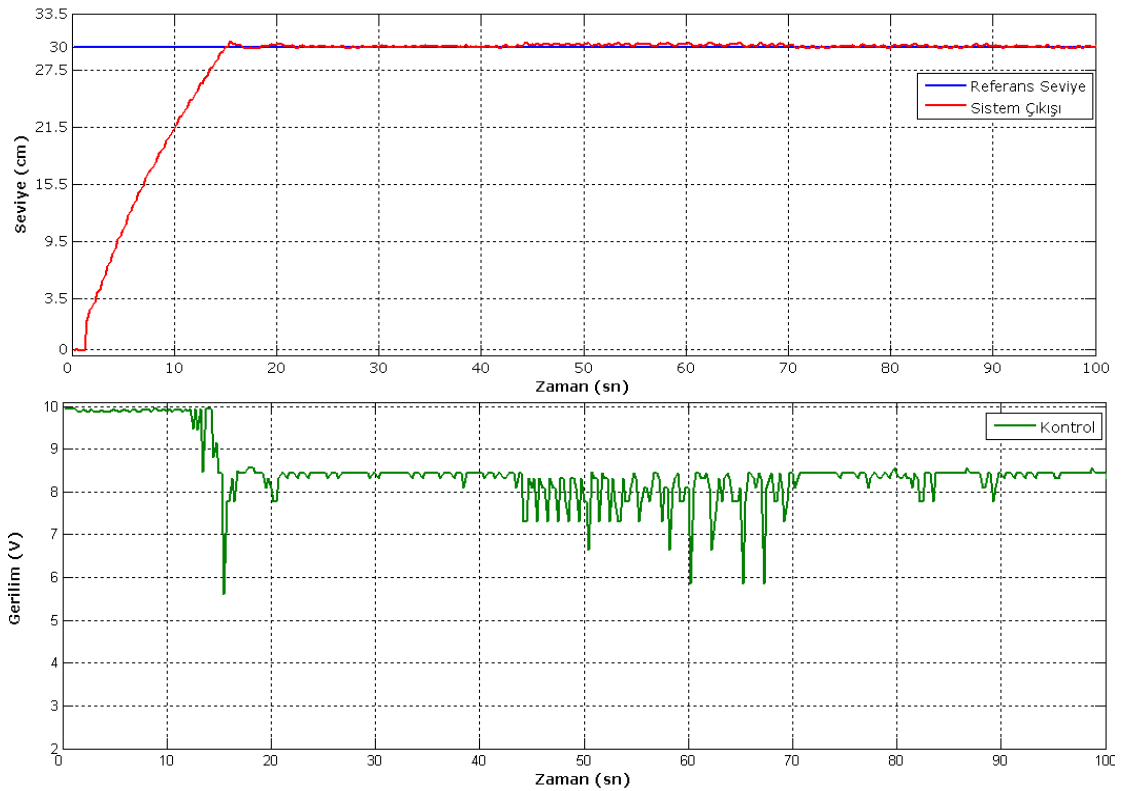


Şekil VI.19 Basamak geçişler sırasında sistem davranışı, kontrol işareti değişimi

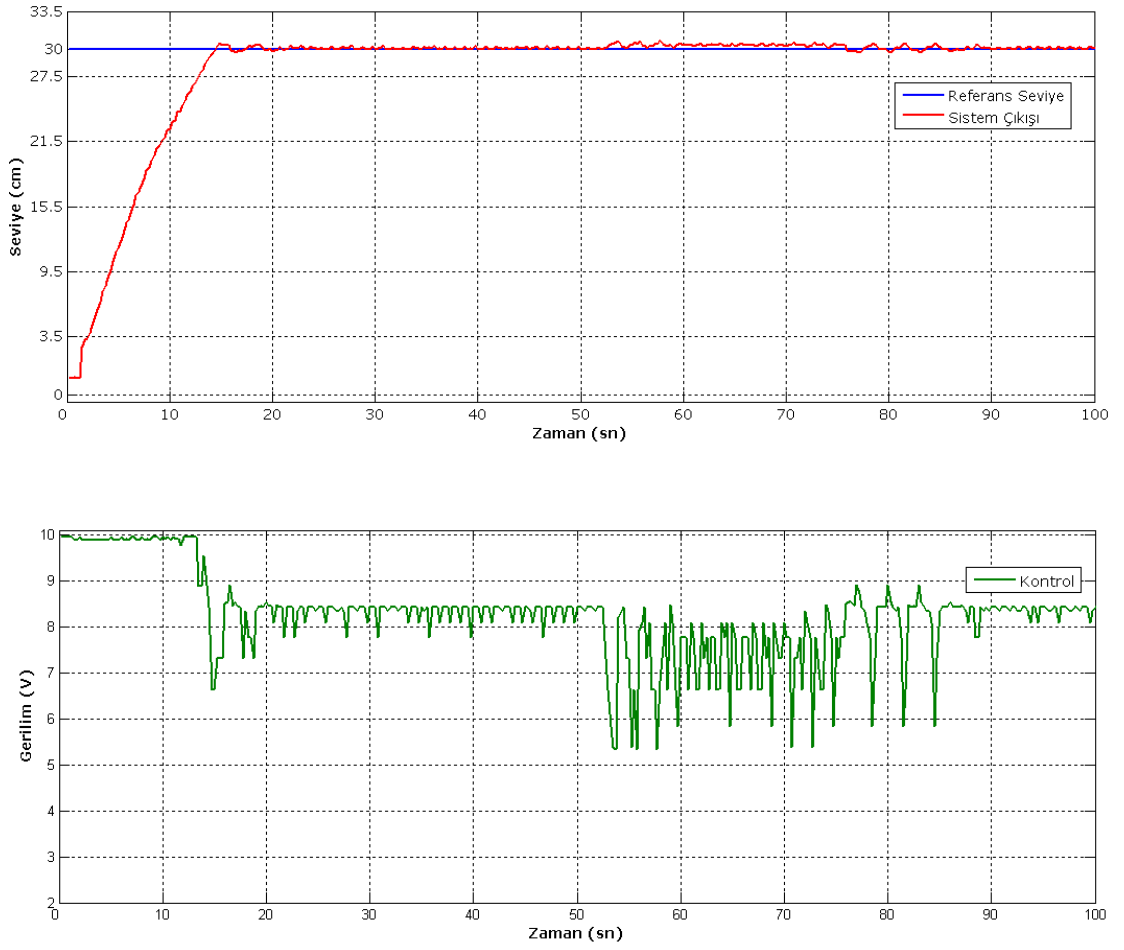
VI.2.3. Sisteme Bozucu İthal Edildiğinde Sistemin Davranışı

Genetik bulanık denetleyici ile gerçekleştirilen sisteme tahliye vanasının yüzdelik değeri ile bozucu ithal edilmiştir. Bozucular, bir önceki sisteme benzer şekilde 30 cm seviyesinde oturmuş iken sırasıyla %25, %40, %50, %60, %70, %80 ve %90 olarak verilmiş ve kontrol işareti ile birlikte sistem davranışı incelenmiştir. Bu sayede KKO – Bulanık ve Genetik – Bulanık denetleyicinin bozucular karşısındaki performanslarını da aynı referans seviyeleri için kıyaslamak mümkün olmuştur.

Şekil VI.20’de sistemin çıkışındaki vana 45. saniyeden itibaren %25 oranında kapatılarak sistemden sıvı çıkışının daha yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici, kontrol işaretini azaltarak seviyenin referans ile birlikte hareket etmesini sağlamıştır. Bozucu uygulandığı zaman içerisinde denetleyici istenen seviyeyi sağlayabilmek için kontrol işaretini 6V’a kadar düşürmüştür. 70. saniyede bozucu tekrar %0’a getirilmiş bu andan itibaren denetleyici 8.5V kontrol işareti göndererek referans seviyeyi korumuştur. 25 saniyelik bozucu süresince seviyede KKO denetleyicili tasarımda olduğu gibi yaklaşık 0.1cm’lik bir sapma olmaktadır.

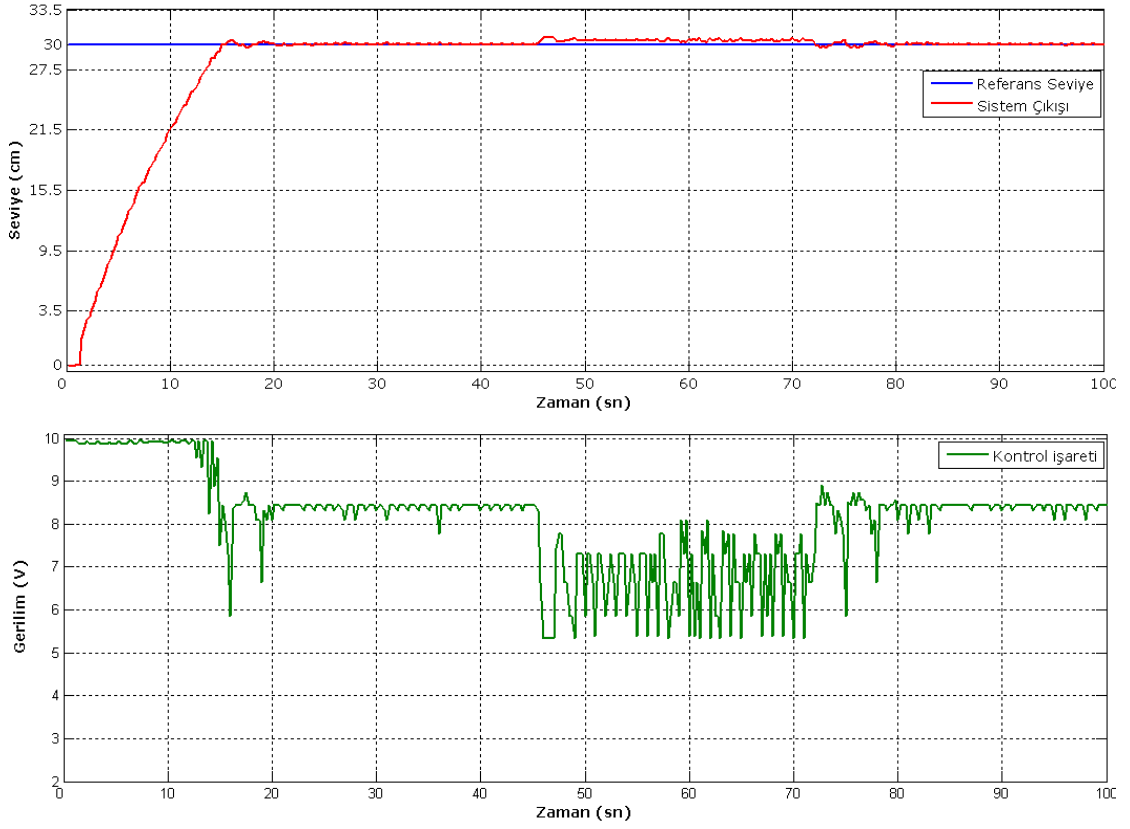


Şekil VI.20 %25 bozucu uygulanan sistemin cevabı ve kontrol işareti



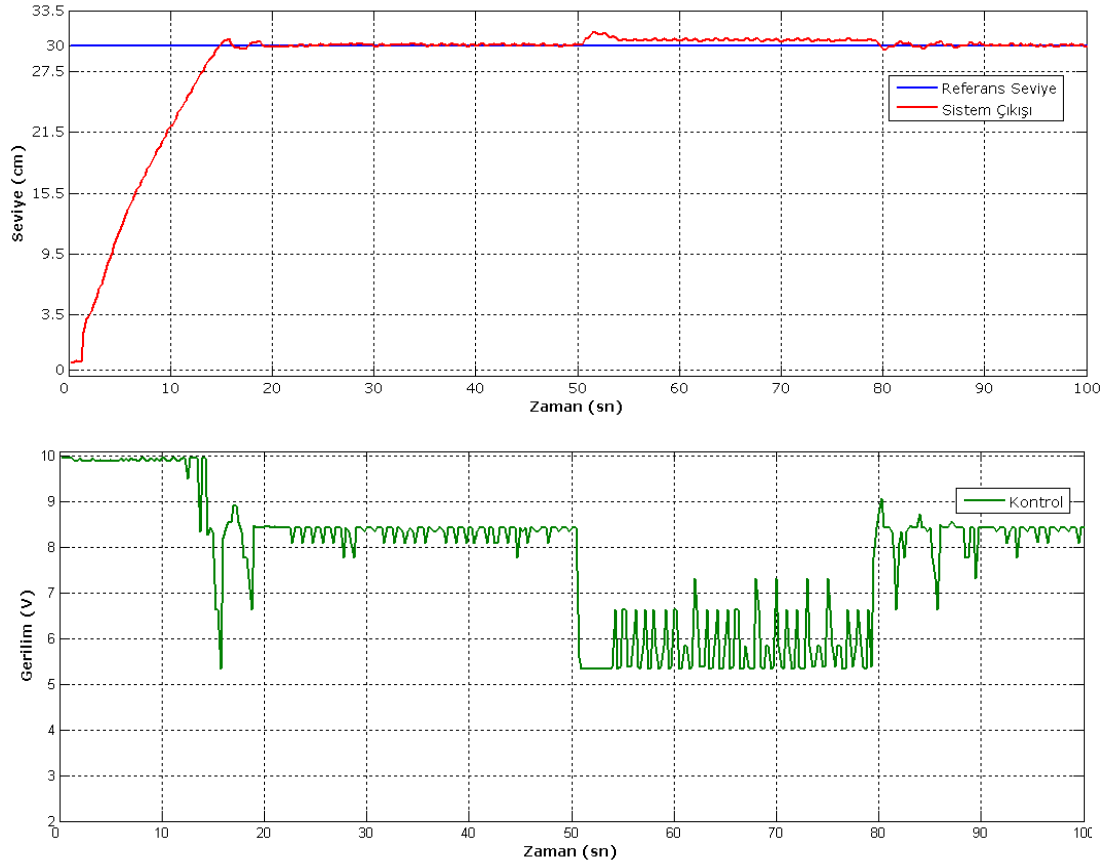
Şekil VI.21 %40 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.21’de sistemin çıkışındaki vana 53. saniyeden itibaren %40 oranında kapatılarak sistemden sıvı çıkışının daha yavaş yapılması sağlanmıştır. Bu durumda bulanık denetleyici, kontrol işaretini azaltarak seviyenin referans ile birlikte hareket etmesini sağlamıştır. Bozucu uygulandığı zaman içerisinde denetleyici istenen seviyeyi sağlayabilmek için kontrol işaretini 5.5V’a kadar düşürmüştür. 78. saniyede bozucu tekrar %0’a getirilmiş bu andan itibaren denetleyici 8.5V kontrol işareti göndererek referans seviyeyi korumuştur. 15 saniyelik bozucu süresince seviyede KKO denetleyicili tasarımda olduğu gibi yaklaşık 0.25cm’lik bir sapma oluşmuştur.



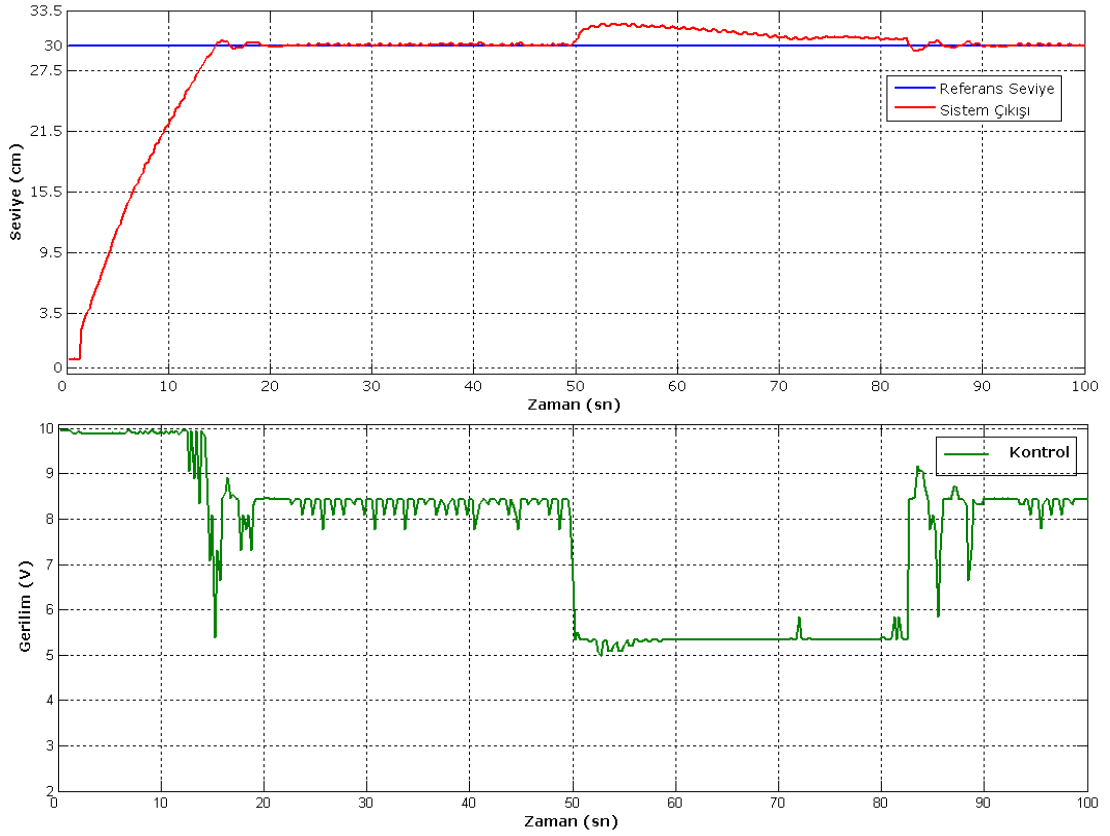
Şekil VI.22 %50 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.22’de sistemin çıkışındaki vana 46. saniyeden itibaren %50 oranında kapatılmıştır. Dolayısıyla sistem çıkışındaki tahliye işleminin daha yavaş yapılması ile bozucu uygulanmıştır. Bu andan itibaren su seviyesini istenen referansta tutabilmek için bulanık denetleyici, kontrol işaretini 5.3V’a kadar düşürmüştür ancak kalıcı hal hatasını engelleyememiştir. 72. saniyede bozucu kaldırılarak kontrol işareti kademeli olarak artmış, denetleyici su seviyesinin istenen seviyeye oturmasını sağlamıştır. 26 saniyelik bozucu süresince yaklaşık 0.4 cm’lik bir sapma oluşmuştur.



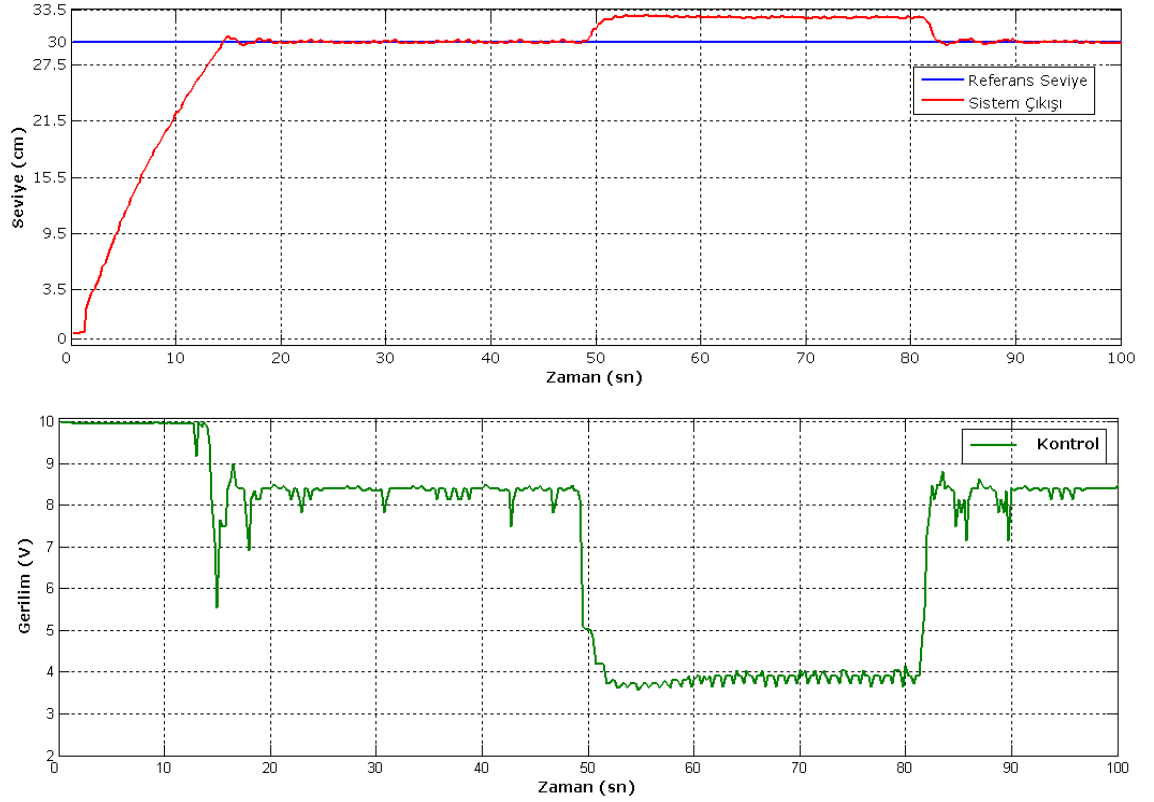
Şekil VI.23 %60 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.23’de sistemin çıkışındaki vana 51. saniyeden itibaren %60 oranında kapatılmıştır. Dolayısıyla sistem çıkışındaki tahliye işleminin daha yavaş yapılması ile bozucu uygulanmıştır. Bu andan itibaren su seviyesini istenen referansta tutabilmek için bulanık denetleyici, kontrol işaretini 5.3V’a kadar düşürmüştür ancak kalıcı hal hatasını engelleyememiştir. 79. saniyede bozucu kaldırılarak kontrol işareti kademeli olarak artmış, denetleyici su seviyesinin istenen seviyeye oturmasını sağlamıştır. 28 saniyelik bozucu süresince yaklaşık 0.85 cm’lik bir sapma oluşmuştur.



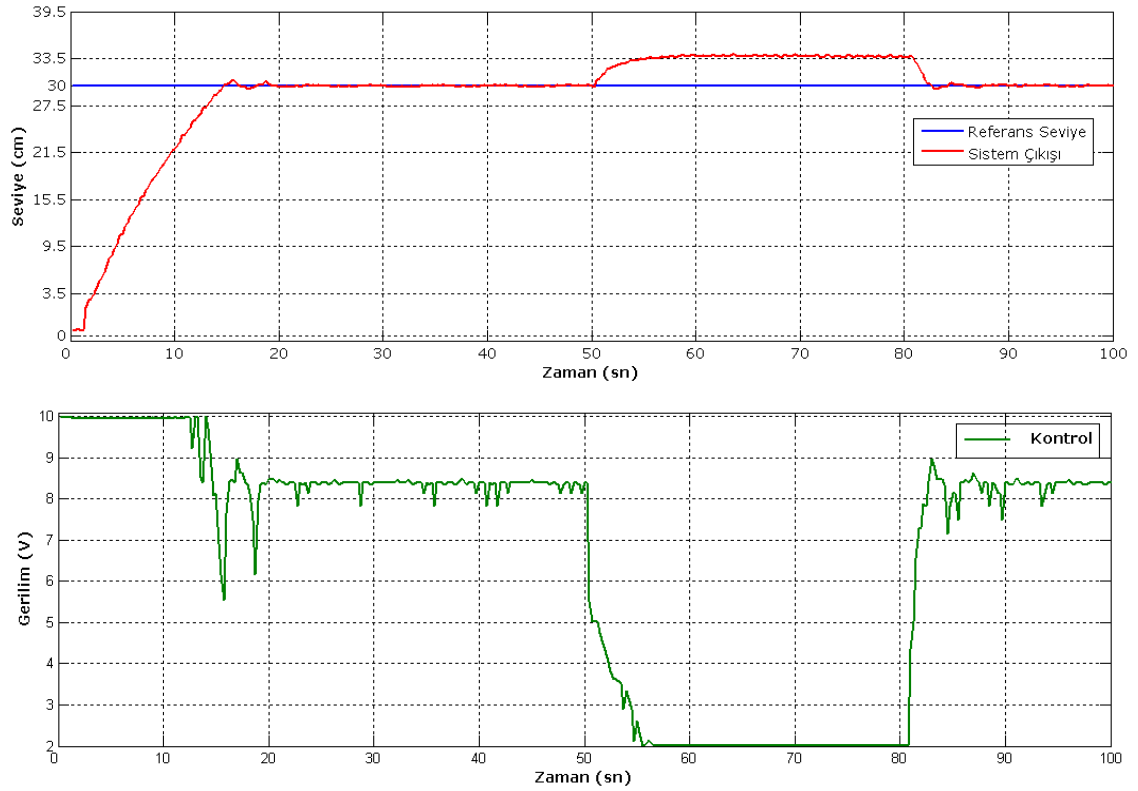
Şekil VI.24 %70 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.24’de sistemin çıkışındaki vana 48. saniyeden itibaren %70 oranında kapatılmıştır. Dolayısıyla sistem çıkışındaki tahliye işleminin daha yavaş yapılması ile bozucu uygulanmıştır. Bu andan itibaren su seviyesini istenen referansta tutabilmek için bulanık denetleyici, kontrol işaretini 5.3V’a kadar düşürmüştür ancak kalıcı hal hatasını engelleyememiştir. 84. saniyede bozucu kaldırılarak kontrol işareti kademeli olarak artmış, denetleyici su seviyesinin istenen seviyeye oturmasını sağlamıştır. 36 saniyelik bozucu süresince yaklaşık 1cm’lik bir sapma oluşmuştur. Önceki bozuculardan farklı olarak burada sistemde ilk bozucu ile birlikte görülen aşım dikkat çekicidir.



Şekil VI.25 % 80 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.25’de sistemin çıkışındaki vana 48. saniyeden itibaren % 80 oranında kapatılmıştır. Dolayısıyla sistem çıkışındaki tahliye işleminin daha yavaş yapılması ile bozucu uygulanmıştır. Bu andan itibaren su seviyesini istenen referansta tutabilmek için bulanık denetleyici, kontrol işaretini 3.7V’a kadar düşürmüştür ancak kalıcı hal hatasını engelleyememiştir. 83. saniyede bozucu kaldırılarak kontrol işareti kademeli olarak artmış, denetleyici su seviyesinin istenen seviyeye oturmasını sağlamıştır. 35 saniyelik bozucu sonunda 2.5cm’ye ulaşan sapma değeri denetleyiciden gelmesi gereken minimum işaret tanımlanan aralıkta gelmediği için artarak devam etmektedir. Aksiyon değerlerinde bu aralık küçültülerek veya veya araya eklenebilecek bir üyelik fonksiyonu ile bu sorunda ortadan kaldırılabilir.



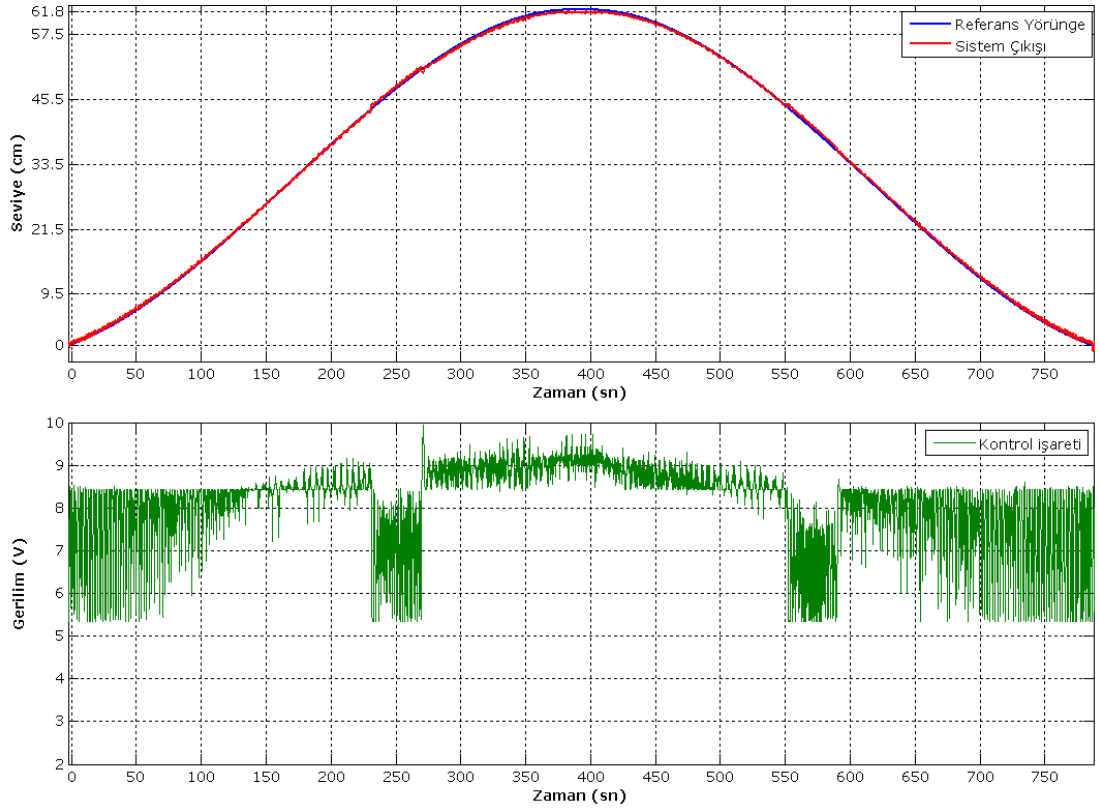
Şekil VI.26 % 90 bozucu uygulanan sistemin cevabı ve kontrol işareti

Şekil VI.26’da sistemin çıkışındaki vana 50. saniyeden itibaren % 90 oranında kapatılmıştır. Bu andan itibaren su seviyesini istenen referansta tutabilmek için bulanık denetleyici, kontrol işaretini 2V’a kadar düşürmüştü ancak 3.5cm’lik sapmayı gene de engelleyememiştir. 83. saniyede bozucu kaldırılarak kontrol işareti kademeli olarak artmış, denetleyici su seviyesinin istenen seviyeye oturmasını sağlamıştır.

VI.2.4. Genetik – Bulanık Denetleyicinin Yörünge Takip Cevapları

Burada, KKO - Bulanık denetleyicide olduğu gibi yine sistem girişine Denklem II.1’de tanımlanan fonksiyon 2 parçalı referans olarak verilmiştir. Şekil VI.27’deki yörünge katsayısı 20 olarak ayarlanmış ve sisteme referans olarak verilmiştir. Sistem 61.8cm’ye geldiğinde ise aynı eğimle ilk konumuna gelmesi için gerekli yörünge oluşturularak sistemin boşalması sırasındaki cevabı incelenmiştir. Sistemin dolum ve boşalım sırasında yörüngeyi başarılı bir şekilde takip ettiği görülmüş ancak 10cm’nin altındaki referans seviyeleri için yüksek titreşimden kaynaklanan seviye değişimleri kontrol işaretindeki salınımlara da sebep olmuştur. Ancak, bu referans yörüngeyi takip etmeyi engellememiştir. Sisteme 230 ve 560.

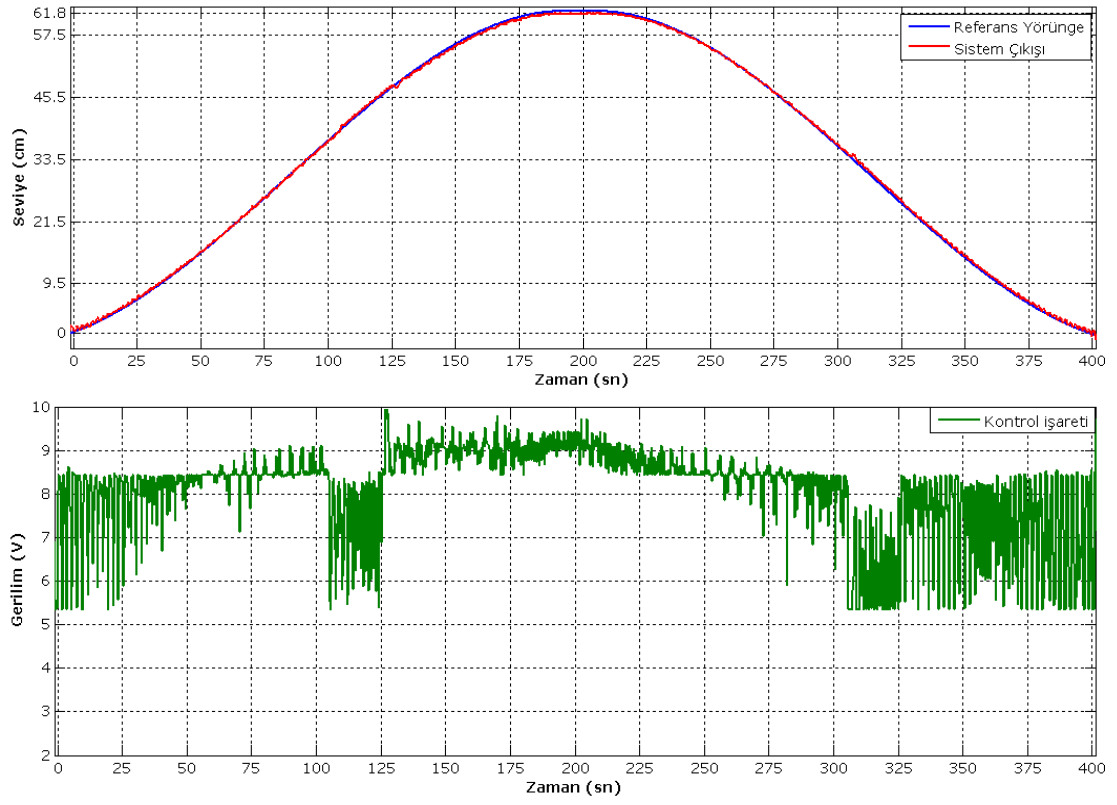
saniyelerde 50 saniye süresince %50 bozucular verilmesine rağmen sistem referans yörüngeyi izlemiştir.



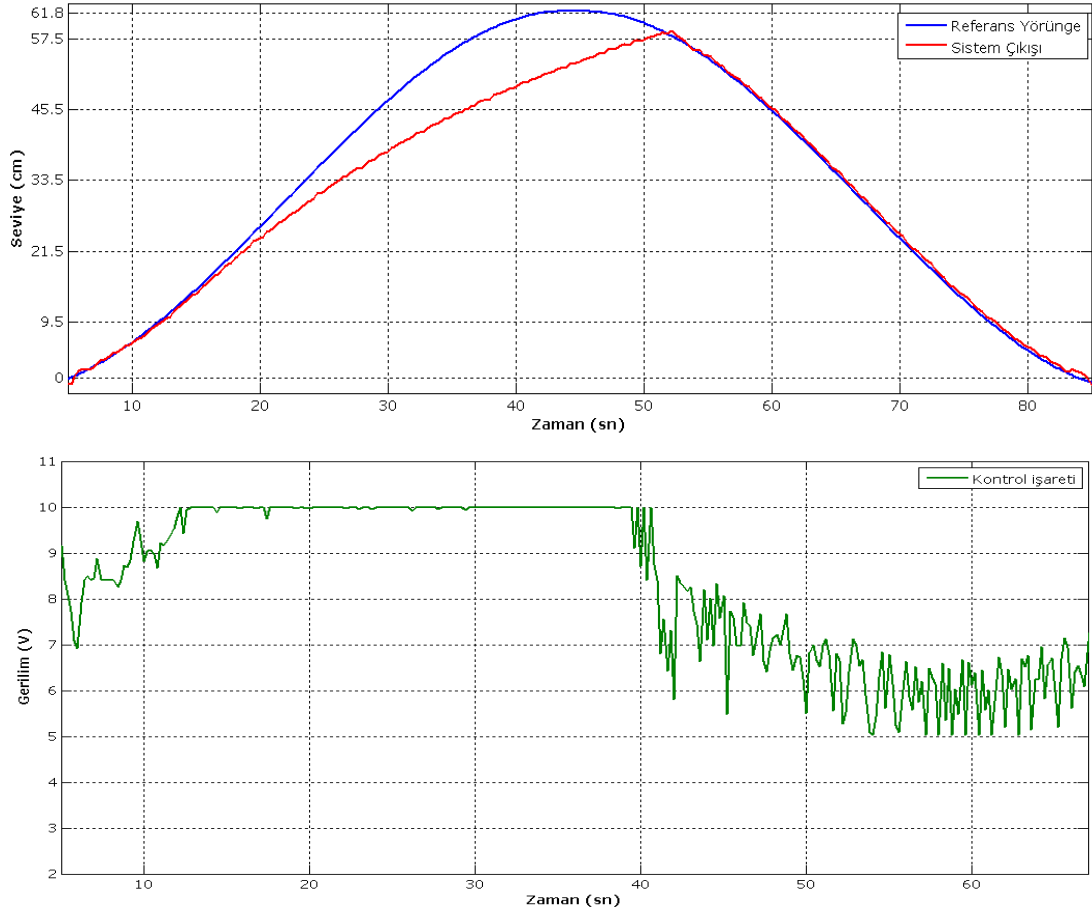
Şekil VI.27 Yörünge katsayısı 20 olan eğri için sistem cevabı ve kontrol işareti

Şekil VI.28’de görüldüğü gibi, referans eğrinin yörünge katsayısı 10 yapılarak sisteme verilmiştir. Sisteme 110. ve 310. saniyelerde 50 saniye süresince %50 bozucular verilerek denetleyici davranışı incelenmiştir. Sistem, bu referans yörüngeye %50 bozucular verilmesine karşın referansa oturmuştur.

Şekil VI.29’da görüldüğü gibi, referans yörünge katsayısı 2 yapılarak sisteme verilmiştir. Sistem referans yörüngeyi 15. saniyeye kadar çok iyi takip etmesine rağmen eğimin yüksekliğinden kaynaklanan sert çıkışa denetleyici en yüksek çıkış işaretini üretmesine rağmen erişememiş ve referans yörüngeyi yakalamış ve inişe kadar referansı takip etmiştir.



Şekil VI.28 Yörünge katsayısı 10 olan eğri için sistem cevabı ve kontrol işareti



Şekil VI.29 Yörünge katsayısı 2 olan eğri için sistem cevabı ve kontrol işareti

BÖLÜM VII

KKO - BULANIK, GENETİK – BULANIK DENETLEYİCİ DENEY SONUÇLARININ KARŞILAŞTIRILMASI

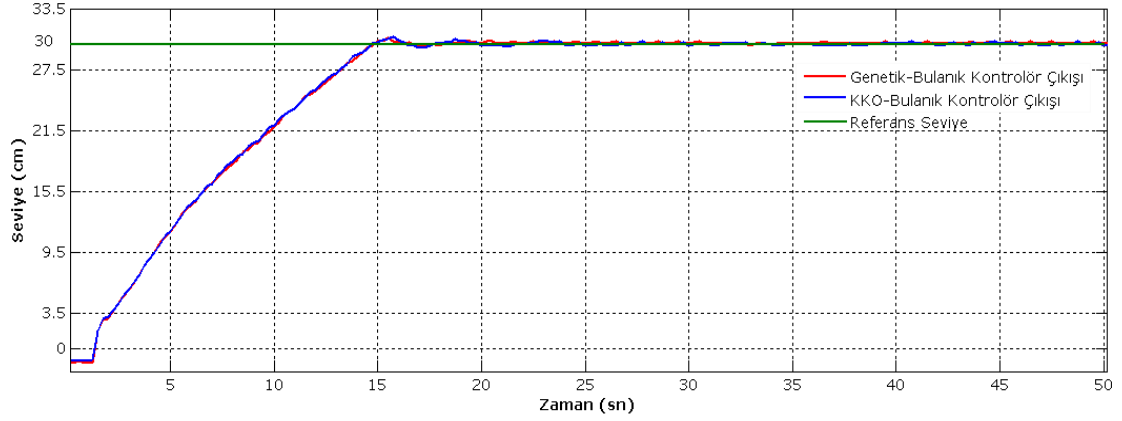
VII.1. GİRİŞ

Bu bölümde, Karınca Koloni Optimizasyonu (KKO) ile Genetik algoritmanın bulanık kontrol ile birlikte kullanılarak performanslarının karşılaştırılması amaçlanmıştır. Her iki algoritma optimizasyon algoritması olduğu için, GuntRT 512 su seviye sisteminde kullanılan bulanık denetleyicinin üyelik fonksiyonu sınır değerleri ve aksiyon değerlerinin elde edilmesinde kullanılmıştır. KKO ve Genetik algoritmaları ile parametre optimizasyonu yapabilmek için, sistemin matematik modeline ihtiyaç olduğundan dolayı su seviye sistemi YSA ile modellenmiştir. Elde edilen YSA modeline ile bulanık denetleyici parametreleri, her iki algoritma ile de optimize edilmiştir. KKO - Bulanık ve Genetik – Bulanık denetleyici Matlab’de simülasyon olarak ayrı ayrı çalıştırılmıştır. Her iki denetleyici su seviye sisteminde kullanılmış ve sistem davranışları Bölüm VI’da ele alınmıştır. Tüm simülasyon çalışmalarında ve deneylerde örnekleme zamanı 0.25sn olarak kullanılmıştır.

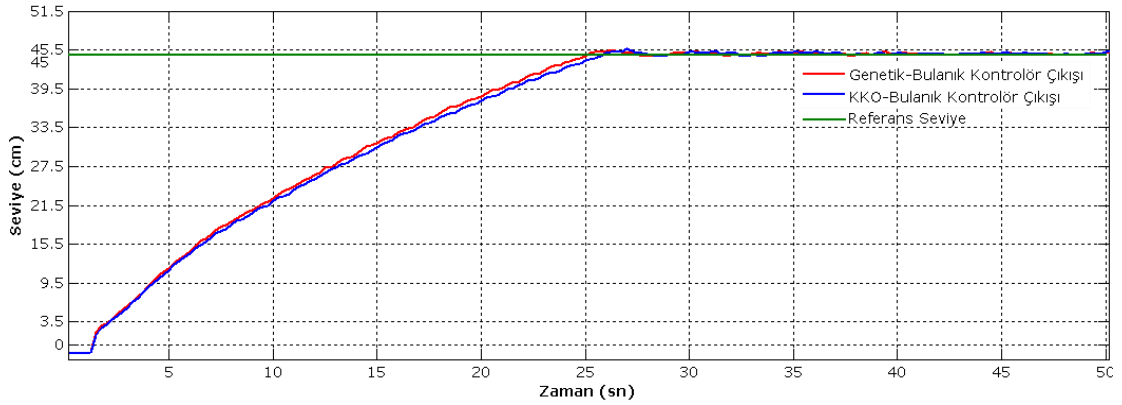
Algoritma performansları değerlendirilirken basamak girişler için; yükselme zamanı (t_r), yerleşme zamanı (t_s), maksimum aşım (y_m) ve kalıcı hal hatası (e_{ss}), referans yörünge için ise RMS hata değeri değerlendirmeye alınmış, algoritma performansları karşılaştırmalı olarak ortaya konmuştur. Aynı zamanda örnekleme zamanı 0.01sn ve 2sn aralıklarında seçilerek optimizasyon algoritmalarının performansları incelenmiştir. Bu sayede kullanılan sistem için örnekleme zamanı değişimlerinin sistem davranışına etkisi incelenmiştir.

VII.1. Farklı Referans Seviyeleri için KKO ve Genetik Algoritma Davranışları

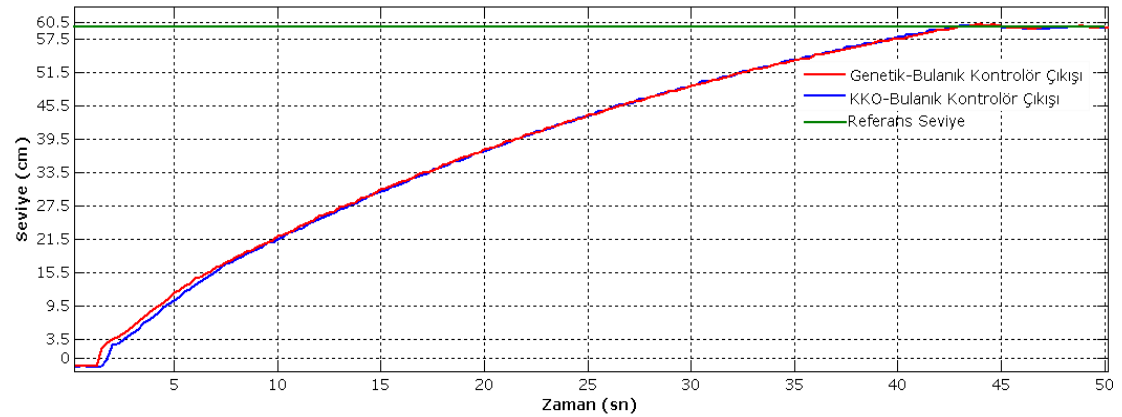
KKO – Bulanık ve Genetik – Bulanık denetleyiciler, sıvı seviye sisteminde 30cm, 45cm ve 60 cm referans seviyeleri için ayrı ayrı çalıştırılmıştır. Her iki algoritma ile elde edilen çıkış eğrileri Şekil VII.1, Şekil VII.2 ve Şekil VII.3’de verilmiştir. Şekillerden de anlaşılacağı üzere her iki optimizasyon algoritmasının da gerçek zamanlı performansları oldukça yakındır.



Şekil VII.1 30cm referans seviyesi için gerçek sistem ve simülasyon çıkışları



Şekil VII.2 45cm referans seviyesi için gerçek sistem ve simülasyon çıkışları



Şekil VII.3 60cm referans seviyesi için gerçek sistem ve simülasyon çıkışları

Referans seviyeler için KKO algoritması Genetik algoritmaya davranışları incelendiğinde her iki algoritma da farklı bulanık denetleyici parametreleri ile çalışmasına rağmen elde ettikleri değerler ile çalıştırılan sistem oldukça benzer davranış göstermektedir. Tablo VII.1’de her iki algoritmanın referans seviyeleri (30cm, 45cm, 60cm) için yükselme zamanı, oturma zamanı, maksimum aşım, kalıcı hal hatası değerleri verilmiştir.

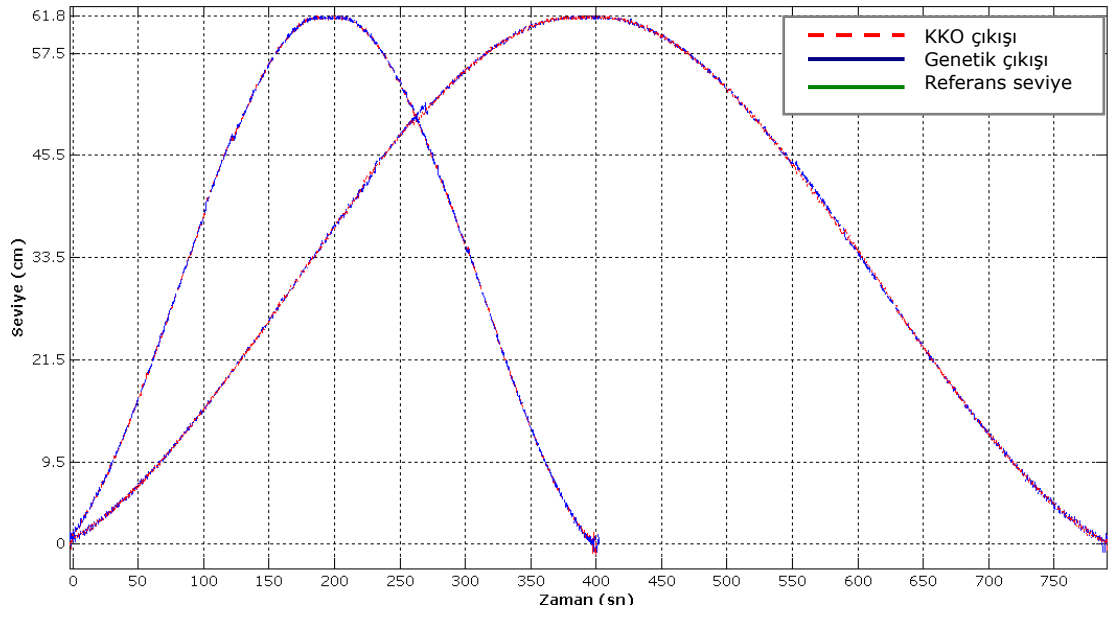
Tablo VII.1’de verilen sistem çıkışları incelendiğinde her iki algoritma ile elde edilen parametrelerin sistem davranışı üzerindeki etkisinin oldukça benzer olduğu görülebilir. Burada referans seviye değeri arttıkça, sistem çıkışlarındaki maksimum aşım ve kararlı hal hata değerlerinin birbirlerine çok yakın olduğu ve seviye artışı ile birlikte düştüğü, yükselme zamanı ile yerleşme zamanı değerlerinin de tüm seviyeler için oldukça yakın olduğu görülmektedir.

Tablo VII.1 Kalıcı ve geçici hal cevaplarına ilişkin parametrelerin karşılaştırılması

Ref. Seviye	Algoritma	t_r	t_s	y_m (Os)	e_{ss}
30 cm	KKO	11.25	15.25	0.57 (%1.9)	0.08
	Genetik	11.5	15.5	0.53 (%1.8)	0.07
45 cm	KKO	19.25	27.5	0.50 (%1.1)	0.06
	Genetik	19.75	29	0.46 (%1.0)	0.05
60 cm	KKO	34.25	43.25	0.42 (%0.7)	0.03
	Genetik	35	44.75	0.21 (%0.35)	0.02

VII.2. Yörünge Takibinde KKO ve Genetik Algoritma Davranışları

Su seviye sistemine 10 ve 20 derece eğimlerdeki referans yörüngeleri verilmiştir. Şekil VII.4’de farklı eğimlerdeki referans yörüngeler için sistem çıkışları ve bu çıkışlardan elde edilen RMS değerleri de Tablo VII.2’de verilmiştir. Burada da basamak girişlerdeki sistem cevaplarında olduğu gibi her iki algoritmanın performanslarının oldukça yakın olduğunu söylemek mümkündür.



Şekil VII.4 Farklı eğimlerdeki yörüngeler için sistem cevapları

Tablo VII.2 Farklı eğimlerdeki referans için sistem çıkışı RMS değerleri

Eğim	Algoritma	RMS
10	KKO	0.0057
	Genetik	0.0058
20	KKO	0.0113
	Genetik	0.0117

BÖLÜM VIII

TARTIŞMA ve DEĞERLENDİRME

VIII.1. GİRİŞ

Yapılan bu çalışmada öncelikle GuntRT 512 sıvı seviye kontrol sisteminin Yapay Sinir Ağı (YSA) modeli oluşturulmuştur. Pnomatik oransal valf ve su tankından oluşan bu sistemin sıvı seviyesinin kontrolü Bulanık denetleyici ile gerçekleştirilmiş, denetleyicinin aksiyon değerleri ve üyelik fonksiyonları sınır değerleri esnek programlama yöntemi olarak bilinen genetik ve karınca kolonisi algoritmaları ile optimize edilmiştir. Elde edilen denetleyici tasarımı ile sistem gerçek zamanlı olarak çalıştırılmış ve elde edilen sonuçlar simülasyon sonuçları ile ve her iki algoritmanın performansını görmek için birbiri ile karşılaştırılmıştır.

Bulanık denetleyici parametrelerinin GA ve KKO ile elde edilebilmesi için sistemin matematiksel modeli gerektiğinden, sistem YSA ile modellenmiştir. Elde edilen modele farklı referans değerleri giriş olarak verilmiş ve sistemin çıkışını, girişi ile arasında en az hata ile takip etmesini sağlayan Genetik-Bulanık ve KKO-Bulanık denetleyiciler tasarlanmıştır. Bu denetleyiciler ile sıvı seviye ünitesi gerçek zamanlı kontrol edilmiş, ve performansları farklı kriterler ile değerlendirilerek incelenmiştir.

Bulanık denetleyici tasarımı genel olarak bir uzmanın tecrübesine ihtiyaç duyar. Bu tür bir uzman tecrübesinin elde edilememesinin imkansız, çok zor veya çok uzun zaman aldığı durumlar mevcuttur. Genetik algoritma ve karınca koloni optimizasyonu algoritması bu tür durumlara çözüm bulmak için kullanılabilecek alternatiflerdir. Sıvı seviye sisteminin kontrolü için Sugeno tipinde bir bulanık denetleyici tasarlanmıştır. Bulanık denetleyicide hata için 5 ve hatanın değişimi için 3'er adet üyelik fonksiyonu olmak üzere 15 kural ve üyelik fonksiyonu sınır değerleri için 24 değişken tanımlanmış, optimizasyon algoritmalarının bu 39 değeri optimize etmesi hedeflenmiştir.

Gerçekleştirmiş olduğumuz genetik bulanık denetleyicide, SGA algoritması ve Pittsburg yaklaşımı kullanılarak, bulanık denetleyici aksiyon ağırlık değerleri ve üyelik fonksiyonlarının değerleri tek bir kromozom içinde kodlanmış, ceza fonksiyonu yaklaşımını ve RMS hatasını kullanan bir uygunluk fonksiyonu geliştirilmiştir. Bu sayede referansı izleyebilecek en iyi denetleyici davranışı için uygun parametreler elde edilebilmiştir. Genetik algoritmanın performansı, genetik algoritma parametrelerine ve bunların değerlerine bağlıdır. Bu parametrelerin birbirleri ile olan ilişkisi lineer değildir ve sistemden sisteme değişmektedir.

KKO ile bulanık denetleyici katsayıları bulunurken, verilen arama uzayında KKO'nun en kısa yolu arama yöntemi ile kombinasyon yaparak çözüm üretir ve verilen sistem için bulanık denetleyici parametreleri bulunabilir. Burada kullanılan ödül ve ceza puanları ile feromon buharlaşma katsayıları, kullanılan karınca sayısı algoritmanın performansını sistemden sisteme farklılık gösterecek şekilde değiştirebilir.

Tez sürecinde yaşanan bazı zorluklar vardır. Bunlar; sistemde kullanılan 1 fazlı asenkron motorun zamanla ısınması ve bu ısınmadan kaynaklanan performans düşüklüğü sebebi ile sistem modellemesinde sorunlar yaşanmıştır. Bu sorunu ortadan kaldırmak için sistem aralıklarla motor soğutularak çalıştırılmıştır. Benzer şekilde şebeke gerilimindeki değişimden kaynaklanan sorunlar da regülatör kullanılarak ortadan kaldırılmıştır. Sistemde kullanılan V/I dönüştürücü Advantech ADAM 2010 modülü de daha hassas çalışan Phoenix modülü ile değiştirilmiştir. Sistemde topraklamadan kaynaklanan gürültü problemi de tüm bağlantılar yeniden yapılarak ortadan kaldırılmıştır.

Sistemin dinamik modelini çıkarmak için kullanılan YSA yönteminde farklı katman ve nöron sayılı YSA yapıları eğitilmiş ve sonuçları değerlendirilmiştir. Katman ve nöron sayısının artması, elde edilecek YSA modelinin eğitim süresini de arttırmıştır.

Bu çalışmanın uzantısında aşağıdaki konular incelenebilir:

- KKO algoritması çok farklı proseslerde uygulanabilirliğinin gösterilmesi gereklidir. Bu KKO algoritmasının popülaritesini arttıracak diğer optimizasyon algoritmaları ile de farklı sistemler için karşılaştırılmasına olanak verecektir.

- Optimize edilecek parametrelerin sayısı (kural veya üyelik fonksiyonu sayısı) değiştirilerek sistem performansı incelenebilir. Ayrıca giriş değişkeni olarak kullanılan hata yerine yüzde hata değişkeni kullanılarak referans seviyeye göre değişen hata değerlerine göre çalışması sağlanabilir.

- 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 olabilir.

- 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.

- Sisteme bozucu verme için kullanılan açma kapatma valfi de bozucu oranı veri toplama kartı ile okunarak sistem modelleme de bozuculu modelleme uygulaması sistemin bozuculara vereceği cevabın da sistem davranışının istenen şekilde davranmasını sağlayacak şekilde etkileyecektir.

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

REFERANSLAR

- [1] Satean, T.; Santi, W.: "Level Control in Horizontal Tank by Fuzzy Logic Control," *SICE-ICASE International Joint Conference*, Busan, Korea, October, (2006).
- [2] Kaplanoğlu, Erkan.: "GuntRT 512 Sıvı Seviye Kontrol Ünitesinin Bulanık Mantık Kontrol Yöntemiyle Kontrolü", *Yüksek Lisans Tezi*, Marmara Üniversitesi Teknik Eğitim Fakültesi, İstanbul, Türkiye, (2000).
- [3] Chengwei, Li.; Jiandong, L.: "The Application of Immune Genetic Algorithm in PID Parameter Optimization for Level Control System," *International Conference on Automation and Logistics*, Jinan, China, August, (2007).
- [4] Mohamed, I.: "Statistical Analysis of Neural Network Modeling and Identification of Nonlinear Systems With Memory," *IEEE*, (2002).
- [5] Jose de, J.; Xiaou, L.: "Time Delay Nonlinear System Modelling via Neural Networks," *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, June, (2002).
- [6] Hakim, N.; Kaufman, J.: "A Discrete-Time Neural Network Model for Systems Identification," *IEEE*, (2000).
- [7] Liu, L.; Huang, D.: "Dynamic System Modeling with Multilayer Recurrent Fuzzy Neural Network," *International Conference on Computational Intelligence and Security*, Shanghai, China, October (2004).
- [8] Valeri, R.; Michael, M.: "The Fuzzy Ant," *IEEE*, (2007).
- [9] Yongsheng, Z.; Baoying, L.: "A New Method for Optimizing Fuzzy Membership Function," *International Conference on Mechatronics and Automation*, Harbin, China, August, (2007).
- [10] Baojiang, Z.; Shiyong, L.: "Design of a Fuzzy Logic Controller by Ant Colony Algorithm with Application to an Inverted Pendulum System," *International Conference on Systems, Man, and Cybernetics*, Taipei, Taiwan, October, (2006).
- [11] Dorigo, M.; Stutzle, M.: "Ant Colony Optimization," *Massachusetts Institute of Technology*, Cambridge, USA, (2004) 15-20.
- [12] Chia, F.; Chiang, L.: "Ant Colony Optimization Algorithm for Fuzzy Controller Design and Its FPGA Implementation," *IEEE*, (2007).

- [13] Weixing, Z.; Jianliang, C.: "Optimal design of Fuzzy controller based on Ant Colony Algorithms," *International Conference on Mechatronics and Automation*, Luoyang,, China, June, **(2006)**.
- [14] Yi-Sheng, Z.; Lin-Ying, L.: "Optimal Design for Fuzzy Controllers by Genetic Algorithms," *IEEE*, **(2000)**.
- [15] Dorigo, M.: "Ant Algorithms for Discrete Optimization," *Artificial Life*, 5 **(1999)** 137.
- [16] Engin, O.: "Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu," **(2001)**.
- [17] Topuz, V.: "Bulanık Genetik Proses Kontrolü," *Doktora Tezi*, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye **(2002)**.
- [18] Özçalık, H. R.; Küçüktüfekçi, A.: "Dinamik Sistemlerin Yapay Sinir Ağları ile Düz ve Ters Modellenmesi," *KSÜ Fen ve Mühendislik Dergisi*, 6 **(2003)** 26-35.
- [19] Hutchins, K.: Identifying Nonlinear Dynamic Systems Using Neural Nets and Evolutionary Programming," *IEEE*, **(2004)**.
- [20] Yaochu, J.; Jingping, J.; Jing, Z.: "Neural Network Based Fuzzy Identification and Its Application to Modeling and Control of Complex Systems," *IEEE*, **(1995)**.
- [21] 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.
- [22] Uçar, A.: "Fuzzy Neural ve Yapay Sinir Ağları ile Lineer Olmayan Sistemlerin Modellenmesi ve Kontrolü," *Yüksek Lisans Tezi*, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ, Türkiye **(2000)**.
- [23] Dong-hai, Z.; Li, L.; Fan, J.: "Fuzzy Neural Network for Nonlinear-Systems Model Identification," *IEEE*, **(2003)**.
- [24] Ming-Guang, Z.; Wen-Hui, Li.: "Single Neuron PID Model Reference Adaptive Control Based On rbf Neural Network," *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics*, Dalian, China, August, **(2006)**.
- [25] Manish, S.; Tekin, M.: "System Identification for the Hodgkin-Huxley Model using Artificial Neural Networks," *Proceedings of International Joint Conference on Neural Networks*, Orlando,Florida, August, **(2007)**.
- [26] Eduard, P.; Sven, N.: "Adaptive Output Feedback Linearization for a Class of NN-based ANARX Models," *IEEE*, **(2007)**.

- [27] Davor, L.; Vladimir, K.: "Fault Diagnosis of a Hydraulic Actuator using Neural Network," *IEEE*, (2004).
- [28] Stevan, M.: "Neural Networks for Modelling of Dynamic Systems with Hysteresis," *IEEE*, (2002).
- [29] Andrzej, D.; Waldemar, G.: "Nonlinear Control Systems Modelling: Global vs. Local," *Proceedings of the American Control Conference Anchorage, Poland, May*, (2002).
- [30] Shi, L.; Sepehri, N.: "Adaptive Fuzzy-Neural-Based Multiple Models for Fault Diagnosis of a Pneumatic Actuator", *American Control Conference Boston, Massachusetts, June 30 – July 2*, (2004)
- [31] Lo, P.; Chan, Y.; Wong, A.; Rad, K, Cheung.: "Fuzzy-genetic algorithm for automatic fault detection in HVAC systems," *Science Direct*, (2006).
- [32] Masoud, M.: "Application of Genetic Algorithms in Fuzzy Rules Generation," *IEEE*, (1995).
- [33] Shinn, Y.; Shinn, J.: "Design of High Performance Fuzzy Controllers Using Flexible Parameterized Membership Functions and Intelligent Genetic Algorithms," " *IEEE*, (2002).
- [34] Işık, Y.: "Genetik Algoritma Tabanlı Bulanık Kontrolün Uçuş Kontrol Sistem Tasarımına Uygulanması," *Doktora Tezi*, Anadolu Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye (2006).
- [35] Şafak, C.: "Pnömatik Motorun Hız Kontrolü İçin Genetik-Bulanık Denetleyici Tasarımı," *Yüksek Lisans Tezi*, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye (2005).
- [36] Erol, O.: "Doğrusal Olmayan Sistemlerin Bulanık Modellenmesi ve Yeni Bir Genetik Arama Algoritması ile En İyi Denetleyici Tasarımı," *Doktora Tezi*, İstanbul Teknik Üniversitesi Fen Bilimleri enstitüsü, İstanbul, Türkiye (2001)
- [37] Um, T.; Joo, Y.; Kong Y.; Chun, I.; Kim, S.; Bang, J.: "Optimization of Design Parameters of a Pneumatic System for Solid Freeform Fabrication System Using Genetic Algorithm, " *IEEE*, (2003)
- [38] Wang, J.; Wang, J. D.; Daw, N.; Wu, Q. H.: "Identification of Pneumatic Cylinder Friction Parameters Using Genetic Algorithms", *IEEE/ASME Transactions on Mechatronics*, Vol. 9, No. 1, March (2004)
- [39] Teo, L.; Ian, S.; Marzuki, K.: "Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm," *IEEE*, (1999).
- [40] JingYuan, Z.; YeDe, L.: "Application of Genetic Algorithm in Optimization of Fuzzy Control Rules," *IEEE*, (2006).

- [41] Xiao-Ping, Z.; Yan-Jie, K.: "Optimization Design Method of Fuzzy Controller Based on GA," *IEEE*, (2005).
- [42] Dongrui, W.; Woei, T.: "A simplified type-2 fuzzy logic controller for real-time control," *ISA Transactions*, (2006).
- [43] Hisao, I.; Takashi, Y.: "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Elsevier*, (2003).
- [44] Rubén, L.; Nun Pitalúa, D.: "Tuning Fuzzy Control Rules via Genetic Algorithms," *IEEE*, (2007).
- [45] Jingzhe, Y.; Zhen, Y.: "Using Genetic Algorithms in Fine-tuning of Fuzzy Logic Controller," *International Conference on Intelligent Processing Sygcms*, Beijing, China, October, (1997).
- [46] Marcio, S.; Marconi, M.: "Optimization of Takagi-Sugeno Fuzzy Controllers Using a Genetic Algorithm," *IEEE*, (2000).
- [47] Khaled, B.; Salim, F.; Nesrine, T.: "Reduction of the number of rules in fuzzy controllers using genetic algorithm," *IEEE*, (2004).
- [48] Chan, P.; Rad, A.; Tsang, K.: "Optimization of Fused Fuzzy Systems via Genetic Algorithms," *IEEE*, (2002).
- [49] Francesco, C.; Vincenzo, G.: "Optimization of fuzzy controllers for industrial manipulators via genetic algorithms," *IEEE*, (2003).
- [50] Enwang, Z.; Alireza, K.: "Fuzzy classifier design using genetic algorithms," *Science Direct*, (2007).
- [51] Ahmed, A.; Abdel, E.: "GA tuning of Fuzzy Controller for MIMO system," *IEEE*, (2006).
- [52] Pablo, C.; Juan, Castro.: "Using Ant Colony Optimization for Learning Maximal Structure Fuzzy Rules," *IEEE*, (2005).
- [53] Taşkiran, Duygu.: "A Revised Multiple Ant Colony System For Vehicle Routing Problems With Time Windows", *Yüksek Lisans Tezi, Sabancı Üniversitesi Mühendislik Fakültesi, İstanbul, Türkiye*, (2006).
- [54] Ünal, Muhammed.: " PID Denetleyicinin Karınca Kolonisi / Genetik Algoritma Tabanlı Optimizasyonu Ve Gunt Rt 532 Basınç Prosesinin Kontrolü", *Yüksek Lisans Tezi, Marmara Üniversitesi Teknik Eğitim Fakültesi, İstanbul, Türkiye*, (2008).
- [55] Panta, L.; Dušan, T.: "Transportation Modeling: An Artificial Life approach," *IEEE*, (2002).

- [56] Michael, Galea.; Q, Shen.: "Fuzzy Rules from Ant-Inspired Computation," *IEEE*, (2004).
- [57] J, Paetz.: "Towards Ant Colony Optimization of Neuro-Fuzzy Interval Rules," *IEEE*, (2005).
- [58] Chia-Feng, J.: "Ant Colony Optimization Algorithm for Fuzzy Controller Design and Its FPGA Implementation," *IEEE*, (2007).
- [59] Yongsheng, Z.; Baoying, L.: "A New Method for Optimizing Fuzzy Membership Function," *IEEE*, (2007).
- [60] Baojiang, Z.; Shiyong, Li.: "Design of a Fuzzy Logic Controller by Ant Colony Algorithm with Application to an Inverted Pendulum System," *IEEE*, (2006).
- [61] Xiaoyu, S.; Yunlong, Z.: "Study on the combination of genetic algorithms and ant Colony algorithms for solving fuzzy job shop scheduling problems," *IMACS Multiconference on "Computational Engineering in Systems Applications*, Beijing, China, October, (2006).
- [62] Tianyou, C.: "A Weighted Algorithm of Fuzzy Logic Strategy on Water Level Control of Stem Generator," *Conference 011 Decision & Control*, California, USA, December, (1997).
- [63] Man Gyun, N.: " Design of a Genetic Fuzzy Controller for the Nuclear Steam Generator Water Level Control," *IEEE*, (1998).
- [64] Min-Soeng, K.; Chang-Hyun, K.; Rohan, M.: "Evolutionary optimization of fuzzy systems for water level control in the steam generator of nuclear power plant," *IEEE*, (2004).
- [65] Satean, T.; Santi, W.: "Level Control in Horizontal Tank by Fuzzy Logic Controller," *SICE-ICASE International Joint Conference*, Busan Korea, October, (2006).
- [66] Özarslan, M.: "GuntRT 512 Sıvı Seviye Denetim Cihazının Bulanık Denetleyici ile Denetimi", *Yüksek Lisans Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye (2005).
- [67] 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).
- [68] Şafak, C.: "Pnömatik Motorun Hız Kontrolü İçin Genetik-Bulanık Denetleyici Tasarımı," *Yüksek Lisans Tezi*, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye (2005).
- [69] Can, B.: "The optimal control of ITU Triga Mark II Reactor," *The Twelfth European Triga User's Conference*, NRI Bucuresti-Pitesti, Romania, (1992).

- [70] <http://www.learnartificialneuralnetworks.com/#Intro>, "Artificial Neural Networks," (06.11.2008).
- [71] http://tr.wikipedia.org/wiki/Yapay_sinir_aglari, "Yapay Sinir Ağları," (25.11.2008).
- [72] Hagan, M. T.; Demuth, H. B.; Beale, M.: *Neural Network Design*, Boston, PWS (1997).
- [73] Saraç, T.: "Yapay Sinir Ağları," Gazi Üniversitesi Endüstri Mühendisliği Bölümü, Ankara (2004).
- [74] Elmas, Ç.: *Yapay Sinir Ağları (Kuram, Mimari, Uygulama)* Seçkin Yayınları (2003).
- [75] Matlab. "Neural Network." (2008).
- [76] Psaltis, D.; Sideris, A.; Yamamura, A. A.: "A Multilayer Neural Network Controller," *IEEE*, 8 (1985) 17.
- [77] Narendra, K. S.; Parthasarathy, K.: "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, 1 (1990).
- [78] Özçalık, H. R.; Küçüktüfekçi, A.: "Dinamik Sistemlerin Yapay Sinir Ağları ile Düz ve Ters Modellenmesi," *KSÜ Fen ve Mühendislik Dergisi*, 6 (2003) 26-35.
- [79] Dorigo, M.: "Ant Algorithms for Discrete Optimization," *Artificial Life*, 5 (1999) 137.
- [80] Engin, O.: "Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu," (2001).
- [81] Mitsukura, Y.; Yamamoto, T.; Kaneda, M.: "A Design of Self-Tuning PID Controllers Using a Genetic Algorithm," *American Control Conference*, (1999) 1361-1365.
- [82] Karr, C. L.; Freeman, L. M.: *Industrial Applications of Genetic Algorithms* CRC Press (1999).
- [83] Back, T.; Homeister, F.: "Extended Selection Mechanism in Genetic Algorithms," *The Fourth International Conference on Genetics Algorithms*, USA, (1991).
- [84] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs* (1996).
- [85] Coley, D. A.: *An Introduction to Genetic Algorithms For Scientist and Engineers* (1999).

- [86] Bonabeau, E.; Dorigo, M.; Theraulaz, G.: "Inspiration For Optimization From Social Insect Behaviour," *Nature*, 406 (2000).
- [87] 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.
- [88] Colorni, A.; Dorigo, M.; Maniezzo, V.: "Distributed Optimization by Ant Colonies," *The First European Conference on Artificial Life*, Paris, France, (1992).
- [89] Alli, H.; Kaya, M.: "Genetik Algoritma Kullanarak PID Kontrol Parametrelerinin Bulunması," *Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 13 (2001) 1-8.
- [90] Alaykiran, K.; Engin, O.: "Karıncalar Kolonileri Metasezgisel ve Gezgin Satıcı Problemleri Üzerinde Bir Uygulaması," *Gazi Üniv. Müh. Mim. Fak. Der.*, 20 (2005) 69-76.
- [91] Çoban, R.: "GuntRT 512 Sıvı Seviye Denetim Cihazının Bulanık Denetleyici ile Denetimi", *Yüksek Lisans Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye (2005).
- [92] Baba, A. F.: "İTÜ Triga Mark-II Reaktörünün Bulanık Kontrolü", *Doktora Tezi*, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye (1995)
- [93] <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>, "Fuzzy," (29.01.2009).
- [94] Wong, A.; Hamouda, G.: "Optimization of fuzzy rules design using genetic algorithm," *Elsevier*, (2000).
- [95] Osman, M.; Mahmoud, A.: "A combined genetic algorithm-fuzzy logic controller (GA-FLC) in nonlinear programming," *Science Direct*, (2005).
- [96] Sanchez, E.; Shibata, T.; Zadeh, L. A.: *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives* (1997).
- [97] Yeniay, Ö.: "An Overview of Genetic Algorithms," *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*, 2 (2001) 37-49.
- [98] 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.
- [99] Varol, A.; Bingöl, Z.: "A New PID Tuning Technique Using Ant Algorithm," *American Control Conference*, (2004).

ÖZGEÇMİŞ

19.01.1978 Sakarya’da doğdu. İlköğretim’i, Kadıköy 30 Ağustos İlköğretim Okulu’nda, Lise eğitimini de Haydarpaşa Anadolu Lisesi’nde tamamladı. 2000 yılında Marmara Üniversitesi Kontrol – Bilgisayar Öğretmenliği’ni 1.’lik derecesi ile tamamladıktan sonra aynı bölümde “Veri Yapılarının Web Tabanlı Eğitimi” başlıklı çalışması ile yüksek lisansını tamamladı. 2000 yılından itibaren Özel Yüzyıl Işıl Okulları’nda Bilgisayar Öğretmenliği, 2008 yılından itibaren de aynı okulda Uluslararası Projeler Koordinatörlüğü görevini yürütmektedir.