

**T.C.  
İNÖNÜ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**UYARLAMALI VE SİNİRSEL-BULANIK SİSTEMLER KULLANARAK  
KATSAYILARI AYARLANABİLEN KESİR DERECELİ PID KONTROLÖR  
TASARIMI VE UYGULAMASI**

**HÜSEYİN ARPACI**

**DOKTORA TEZİ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**EKİM 2018**

**T.C.  
İNÖNÜ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**UYARLAMALI VE SİNİRSEL-BULANIK SİSTEMLER KULLANARAK  
KATSAYILARI AYARLANABİLEN KESİR DERECELİ PID KONTROLÖR  
TASARIMI VE UYGULAMASI**

**HÜSEYİN ARPACI**

**DOKTORA TEZİ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**EKİM 2018**

**Tezin Başlığı:** Uyarlamalı ve Sinirsel-Bulanık Sistemler Kullanarak Katsayıları Ayarlanabilen Kesir Dereceli PID Kontrolör Tasarımı ve Uygulaması

**Tezi Hazırlayan:** Hüseyin ARPACI

**Sınav Tarihi:** 18/10/2018

Yukarıda adı geçen tez, jürimizce değerlendirilerek Elektrik-Elektronik Mühendisliği Anabilim Dalında Doktora Tezi olarak kabul edilmiştir.

### Sınav Jüri Üyeleri

<b>Tez Danışmanı:</b>	<b>Doç. Dr. Ömer Faruk ÖZGÜVEN</b> İnönü Üniversitesi	.....
	<b>Prof. Dr. Yakup DEMİR</b> Fırat Üniversitesi	.....
	<b>Prof. Dr. Mehmet Salih MAMİŞ</b> İnönü Üniversitesi	.....
	<b>Prof. Dr. Nusret TAN</b> İnönü Üniversitesi	.....
	<b>Doç. Dr. Fevzi HANSU</b> Siirt Üniversitesi	.....

**Prof. Dr. Halil İbrahim ADIGÜZEL**

Enstitü Müdürü

## ONUR SÖZÜ

Doktora Tezi olarak sunduđum “**Uyarlamalı ve Sinirsel-Bulanık Sistemler Kullanarak Katsayıları Ayarlanabilen Kesir Dereceli PID Kontrolör Tasarımı ve Uygulaması**” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla doğrularım.

Hüseyin ARPACI

# ÖZET

Doktora Tezi

## UYARLAMALI VE SİNİRSEL-BULANIK SİSTEMLER KULLANARAK KATSAYILARI AYARLANABİLEN KESİR DERECELİ PID KONTROLÖR TASARIMI VE UYGULAMASI

Hüseyin ARPACI

İnönü Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

109 + xiv sayfa

2018

Danışman: Doç. Dr. Ömer Faruk ÖZGÜVEN

Klasik tam sayı dereceli kontrolörlerin yerine kesir dereceli kontrolörlerin ( $PI^{\lambda}D^{\mu}$ ) kullanılması giderek yaygınlaşmakta ve tercih edilmektedir. Kesir dereceli sistemler kullanılarak, performansı yüksek ve daha dayanıklı sistem cevapları alınabilmektedir. Kesir dereceli kontrolör tasarımı için kontrolör katsayı ve parametrelerin en uygun şekilde hesaplanması gerekir. Bulanık mantık ve yapay sinir ağları kontrol yöntemleri hem birlikte hem de ayrı şekilde kullanılarak katsayıları ayarlanabilen kesir dereceli PID kontrolör tasarımlarına daha kısa sürede tasarım, hız ve dayanıklılık kazandırılmıştır.

Bu tezde, kesir dereceli kontrolör tasarım çalışmalarında, önce model referans uyarlama yöntemi ve daha sonra sinirsel-bulanık sistem yapısı kullanarak kesir dereceli PID kontrolör tasarımları gerçekleştirilmiştir. Kesir dereceli sinirsel-bulanık sistemi ile kontrolör tasarımı için uyarlamalı sinirsel bulanık çıkarım sistemi (ANFIS) kullanılarak kesir dereceli PID kontrolör yapısı oluşturulmuştur. Bu şekilde bulanık mantık ve yapay sinir ağlarının avantajları birlikte kullanılmıştır. Bu yöntemler, çevrim içi veya çevrim dışı olarak kendi kendine öğrenme, organize olma, en iyi çalışma noktasına ayarlanma kabiliyetlerine sahiptirler. Bu tezde tasarlanan kesir dereceli PID kontrolörün katsayıları

için yapay sinir ağı ve ayrıca önceden eğitilmiş ANFIS blokları kullanılmış ve tasarlanan kontrolörlerin benzetim uygulamaları gerçekleştirilmiştir. Son aşamada değiştirilmiş hiperbolik tanjant fonksiyonunu kullanan otomatik ayarlamalı sinir hücreleri ile katsayıları ayarlanabilen kesir dereceli PID kontrolör tasarlanmış ve birim basamak cevapları alınmıştır. Çalışma kapsamında tasarlanan kesir dereceli kontrolörlerin performansları, kararlılıkları ve dayanıklılıkları benzetim programları ile test edilip sistemin çıkış cevapları karşılaştırılarak yorumlanmıştır.

Çalışma sonunda çevrim içi katsayıları ayarlanabilen kontrolörlerin, diğer kontrolörlere göre performanslarının ve dayanıklılıklarının daha iyi olduğu bilgisayar benzetimlerinde saptanmıştır. Ayrıca, kontrolör katsayılarını bulma ve ayarlama işlemindeki uğraşı ve zaman kaybı en aza indirgenmiştir.

**ANAHTAR KELİMELER:** Kesir derece, kontrolör, PID,  $PI^{\lambda}D^{\mu}$ , sinirsel-bulanık sistemler, ayarlamalı sinirsel bulanık çıkarım sistemi, ANFIS, yapay sinir ağı, otomatik ayarlamalı sinir hücreleri.

## **ABSTRACT**

Ph.D. Thesis

# **DESIGN AND APPLICATION OF FRACTIONAL ORDER PID CONTROLLER WITH ADJUSTABLE COEFFICIENT BY USING ADAPTIVE AND NEURO- FUZZY SYSTEMS**

Hüseyin ARPACI

Inonu University

Graduate School of Natural and Applied Sciences

Electrical-Electronics Engineering

109+ xiv pages

2018

Supervisor: Assoc. Prof. Dr. Ö. Faruk ÖZGÜVEN

The use of fractional order controllers ( $PI^\lambda D^\mu$ ) instead of classical integer order controllers is becoming increasingly common and preferred. By using fractional order systems, system responses with high performance and robustness can be obtained. For fractional order controller design, control coefficient and parameters should be calculated in the most appropriate way. By using adaptive, fuzzy logic and artificial neural network control methods either together or separately, less design time, speed and robustness gained to the design of fractional order PID controllers with adjustable coefficients.

In this thesis, firstly a fractional order controller design with adaptive model reference method was performed. After this stage the controller was designed with neural-fuzzy system structure. A fractional PID controller structure was constructed using the adaptive neural fuzzy inference system (ANFIS) for fractional order neural-fuzzy controller design. In this way, the advantages of fuzzy logic and artificial neural networks are used together, because these methods have the ability to self-learning, self-organizing and self-adjusting to the optimum point, online or offline. Artificial neural

networks and previously trained ANFIS blocks are used as coefficients of the fractional PID controller designed in this thesis and the simulations of the designed controllers has been carried out. At the last stage a fractional order controller was designed with a auto-tuning neuron that uses the modified hyperbolic tangent function and the unit step responses were taken.

At the end of the study, the computer simulations show that the controllers whose coefficients adjusted on-line have better performance, stability and robustness than the other controllers. Furthermore, the effort and time lost in the process of finding and adjusting the controller coefficients has been reduced to minimum level.

**KEYWORDS:** Fractional, adaptive, controller, PID,  $PI^\lambda D^\mu$ , neuro-fuzzy, ANFIS, neural-network, auto-tuning neuron.



## TEŐEKKÜR

Bu alıőmanın her aőamasında yardım, öneri ve zamanını esirgmeden beni yönlendiren danışman hocam sayın Do. Dr. Ö. Faruk ÖZGÜVEN' e;

Proje ve yayın desteklerinden dolayı, İnönü Üniversitesi Bilimsel Araőtırma Projeleri Birimi (BAPB-Proje No:2011/23)'ne;

Tezim süresince bana önerilerde bulunan kıymetli bölüm hocalarıma;

Ayrıca tüm hayatım boyunca bana maddi ve manevi her türlü desteęi saęlayan kıymetli aileme ve bu tezin hazırlanmasında katkısı olan herkese;

Teőekkür ederim.

## İÇİNDEKİLER

<b>ÖZET.....</b>	<b>i</b>
<b>ABSTRACT.....</b>	<b>iii</b>
<b>TEŞEKKÜR.....</b>	<b>v</b>
<b>İÇİNDEKİLER.....</b>	<b>vi</b>
<b>SİMGELER VE KISALTMALAR .....</b>	<b>viii</b>
<b>ŞEKİLLER DİZİNİ.....</b>	<b>x</b>
<b>ÇİZELGELER DİZİNİ.....</b>	<b>xiv</b>
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. KURAMSAL TEMELLER.....</b>	<b>7</b>
2.1. Kesir Dereceli Sistemler.....	7
2.2. Kesir Dereceli Türev ve İntegralin Laplace Dönüşümü.....	9
2.3. Birim Geri Beslemeli Kesir Dereceli Kontrol Sistemleri.....	10
2.4. Birim Geri Beslemeli Kesir Dereceli PID ( $PI^{\lambda}D^{\mu}$ ) Kontrol Sistemi.....	11
2.5. Bulanık Mantık Kümeler Teorisi.....	17
2.6. Yapay Sinir Ağları – YSA.....	26
2.6.1. Biyolojik Sinir Yapısı.....	27
2.6.2. Yapay Sinir Ağlarında Öğrenme.....	29
2.6.3. Geriyayılım Ağ Modeli.....	33
<b>3. TEZDE KULLANILAN YÖNTEMLER.....</b>	<b>39</b>
3.1. Model Referans Uyarlamalı Kontrol (MRAC).....	39
3.1.1. Model Referans Uyarlamalı Kontrol (MRAC) ile $PI^{\lambda}D^{\mu}$ kontrollör Tasarımı (MRA- $PI^{\lambda}D^{\mu}$ ).....	42
3.2. Uyarlamalı Sinirsel Bulanık Çıkarım Sistemi (ANFIS - Adaptive Neuro-Fuzzy Inference System) ile $PI^{\lambda}D^{\mu}$ kontrollör (ANFIS- $PI^{\lambda}D^{\mu}$ ) Tasarımı.....	45
3.3. Otomatik Ayarlamalı Sinir Hücresi ile $PI^{\lambda}D^{\mu}$ Kontrollör Tasarımı.....	52
3.3.1. $PI^{\lambda}D^{\mu}$ Kontrollörün Ayrık Zamanda Gösterimi.....	53
3.3.2. Otomatik Ayarlamalı Sinir Hücresi.....	55

<b>4.</b>	<b>ARAŞTIRMA BULGULARI.....</b>	<b>62</b>
4.1.	Model Referans Uyarlamalı Kesir Dereceli PID ( $MRA-PI^\lambda D^\mu$ ) Kontrolör Tasarım Uygulamaları.....	62
4.2.	Uyarlamalı Sinirsel Bulanık Denetim Sistemi (ANFIS) ile Kesir Dereceli PID (ANFIS- $PI^\lambda D^\mu$ ) Kontrolör Tasarım Uygulamaları.....	69
4.3.	Otomatik Ayarlamalı Sinir Hücresi ile Kesir Dereceli PID ( $PI^\lambda D^\mu$ ) Kontrolör Tasarım Uygulamaları.....	86
4.3.1.	P Kontrolör.....	86
4.3.2.	$PI^\lambda$ Kontrolör.....	88
4.3.3.	$PD^\mu$ Kontrolör.....	90
4.3.4.	$PI^\lambda D^\mu$ Kontrolör.....	91
<b>5.</b>	<b>TARTIŞMA VE SONUÇ.....</b>	<b>96</b>
5.1.	Tartışma.....	96
5.2.	Sonuç.....	97
<b>6.</b>	<b>KAYNAKLAR.....</b>	<b>99</b>

## SİMGELER VE KISALTMALAR

$A$	: Evrensel küme
${}_a D_t^\alpha$	: Kesir dereceli türev operatörü
$e(t)$	: Hata
$G_p(s)$	: Kontrol edilecek sistemin transfer fonksiyonu
$G_c(s)$	: Kontrolör transfer fonksiyonu
$G_p^k(s)$	: Kesir dereceli kontrol edilecek sistemin transfer fonksiyonu
$G_c^k(s)$	: Kesir dereceli kontrolör transfer fonksiyonu
$G_{mod}(s)$	: Referans modelin transfer fonksiyonu
$h(net)$	: Otomatik ayarlamalı sinir hücresinin çıkışı
${}_a I_t^\alpha$	: Kesir dereceli integral operatörü
$I$	: Otomatik ayarlamalı sinir hücresinin girişi
$J(\theta)$	: Karesel hata
$K_{cr}$	: Kritik kazanç
$K_d$	: Türev katsayı
$K_i$	: İntegral katsayı
$K_p$	: Oransal katsayı
$L$	: Gecikme zaman sabiti
$N_p(s)$	: Transfer fonksiyonunun payı
$O_i$	: Düğüm çıkışları / Sinir çıkışı
$P$	: Açık çevrimli sistemin kutup sayısı
$P_{cr}$	: Kritik periyot
$r(t)$	: Sistem girişi
$T$	: Karakteristik zaman sabiti
$U$	: Evrensel küme
$u(t)$	: Kontrolör çıkışı
$y(t)$	: Sistem çıkışı
$y_d$	: İstenen çıkış
$y_m$	: Model çıkışı
$\omega$	: Limit çevrimin frekansı
$w_i$	: Sinir hücresinde öğrenme
$\alpha, \beta, \gamma$	: Ayarlama parametreleri

$\lambda, \mu$	: İntegral / Türev dereceleri
$\theta$	: Kontrol parametresi
$\phi$	: Eşik değeri
$\Phi$	: Faz marjı
$\mu_A(x)$	: x' in A alt kümesine üyelik fonksiyonu
$\Gamma(\cdot)$	: Euler Gama fonksiyonu
ADALINE	: Adaptive Linear Neuron
ANFIS	: Uyarlamalı sinirsel bulanık denetim sistemi (Adaptive Neuro Fuzzy Inference System)
ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup>	: Uyarlamalı bulanık denetim sistemi ile kesir dereceli PID kontrolör
CRONE	: Kesir Dereceli Dayanıklı Kontrol (Commande Robuste D' ordre Non Entier)
FLC	: Bulanık mantık denetleyici (Fuzzy Logic Controller)
MIMO	: Çoklu giriş çoklu çıkış (Multi Input Multi Output)
MRAC	: Model referans uyarlamalı kontrol (Model Reference Adaptive Control)
MRA-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup>	: Model referans uyarlamalı (Model Reference Adaptive ) kesir dereceli PID kontrolör
NID	: Kesir dereceli türev (Non integer derivative)
NIPID	: Kesir dereceli PID kontrolör (Non integer PID controller)
P	: Oransal (Proportional) kontrol
PI	: Oransal-Integral (Proportional-Integral) kontrol
PD	: Oransal-Türev (Proportional-Derivative) kontrol
PID	: Oransal-Integral-Türev (Proportional-Integral-Derivative) kontrol
PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup>	: Kesir dereceli PID kontrolör
SA	: Sinir ağları
SOM	: Özörgütlemeli harita ağı (Self Organizing Map)
TGTÇ	: Tek girişli tek çıkışlı
ANN	: Yapay sinir ağları (Artificial Neural Network)
ANN-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup>	: Yapay sinir ağları ile kesir dereceli PID kontrolör

## ŞEKİLLER DİZİNİ

Şekil 2.1.	Birim geri beslemeli kesir dereceli kontrol sistemi.....	10
Şekil 2.2.	Birim geri beslemeli kesir dereceli PID ( $PI^{\lambda}D^{\mu}$ ) kontrol sistemi.....	12
Şekil 2.3.	(a) Tam dereceli PID kontrolör için $\lambda-\mu$ düzlemi (b) Kesir dereceli PID kontrolör için $\lambda-\mu$ düzlemi.....	13
Şekil 2.4.	Birim basamak S-cevabı .....	14
Şekil 2.5.	Birim geri beslemeli sistem.....	15
Şekil 2.6.	$\tilde{A}$ bulanık kümesi için üyelik fonksiyonu.....	18
Şekil 2.7.	Üçgen ve Yamuk Üyelik Fonksiyonları.....	18
Şekil 2.8.	Gauss Eğrisi.....	19
Şekil 2.9.	Çan eğrisi grafiği.....	19
Şekil 2.10.	(a) S tipi sigmoid fonksiyonunun grafiği (b) Z tipi sigmoid fonksiyonunun grafiği.....	20
Şekil 2.11.	Evrensel küme üzerinde birden fazla bulanık kümenin gösterimi.....	20
Şekil 2.12.	$\tilde{A}$ ve $\tilde{B}$ bulanık kümelerinin birleşimi.....	21
Şekil 2.13.	$\tilde{A}$ ve $\tilde{B}$ bulanık kümelerinin kesişimi.....	21
Şekil 2.14.	$\tilde{A}$ bulanık kümesinin tümleyeni.....	21
Şekil 2.15.	$\tilde{A}$ bulanık kümesinin tümleyeni ve tümleme işlemleri.....	22
Şekil 2.16.	FLC blok diyagramı.....	23
Şekil 2.17.	İki girişli tek çıkışlı ANFIS yapısı.....	24
Şekil 2.18.	Biyolojik sinir hücre yapısı.....	27
Şekil 2.19.	Yapay sinir hücre yapısı [ $x_i$ ( $i=1, 2, \dots, n$ ), $w_i$ ( $i=1, 2, \dots, n$ )].....	28
Şekil 2.20.	Çok katmanlı yapay sinir ağı.....	29
Şekil 2.21.	(a) Doğrusal (b) Rampa (c) Basamak (d) Sigmoid fonksiyonları.....	30
Şekil 2.22.	3 katmanlı YSA.....	33
Şekil 2.23.	Geri yayılım ağ modeli.....	34
Şekil 3.1.	Model Referans Uyarlamalı Kontrol (MRAC) blok diyagramı.....	40
Şekil 3.2.	Kesir dereceli integral kullanan MRAC algoritması.....	41
Şekil 3.3.	Model referans uyarlamalı $PI^{\lambda}D^{\mu}$ blok diyagramı.....	42
Şekil 3.4.	Kesirli dereceli MRA- $PI^{\lambda}D^{\mu}$ kontrolörünün blok diyagramı.....	44
Şekil 3.5.	$PI^{\lambda}D^{\mu}$ kontrolör içeren sistemin blok diyagramı.....	46

<b>Şekil 3.6.</b>	$K_p$ , $K_i$ ve $K_d$ katsayıları çevrimiçi ayarlanabilen $PI^\lambda D^\mu$ kontrollör sisteminin blok diyagramı.....	46
<b>Şekil 3.7.</b>	Tek giriş-Tek çıkış ANFIS yapısı.....	48
<b>Şekil 3.8.</b>	ANFIS- $PI^\lambda D^\mu$ kontrollör tasarımı.....	50
<b>Şekil 3.9.</b>	ANFIS- $PI^\lambda D^\mu$ ' nin $K_p$ , $K_i$ ve $K_d$ parametrelerinin ayarlanması.....	50
<b>Şekil 3.10.</b>	Öğrenme verilerini elde etmek için kullanılan $PI^\lambda D^\mu$ kontrollör sisteminin blok diyagramı.....	51
<b>Şekil 3.11.</b>	Otomatik ayarlamalı sinir hücreleri ile $PI^\lambda D^\mu$ kontrol sisteminin blok diyagramı.....	52
<b>Şekil 3.12.</b>	Otomatik ayarlamalı sinir hücresi temel yapısı.....	55
<b>Şekil 3.13.</b>	Farklı $a$ ve $b$ değerleri için hiperbolik tanjant fonksiyonu.....	55
<b>Şekil 3.14.</b>	Otomatik ayarlamalı sinir hücresi.....	57
<b>Şekil 3.15.</b>	$\phi$ , $a$ , $b$ ve parametrelerine göre otomatik ayarlamalı sinir hücresinin blok diyagramı.....	58
<b>Şekil 3.16.</b>	Denklem (3.63.a)' ya göre $a$ parametresinin MATLAB – Simulink gösterimi.....	58
<b>Şekil 3.17.</b>	Denklem (3.63.b)' ye göre $b$ parametresinin MATLAB – Simulink gösterimi.....	59
<b>Şekil 3.18.</b>	Denklem (3.63.c)' ye göre $\phi$ parametresinin MATLAB– Simulink gösterimi.....	60
<b>Şekil 3.19.</b>	Otomatik ayarlamalı sinir hücreleri ile $PI^\lambda D^\mu$ kontrollör.....	61
<b>Şekil 4.1.</b>	$PI^\lambda D^\mu$ kontrollör için sistemin birim basamak cevabı.....	63
<b>Şekil 4.2.</b>	MRA-PID ve MRA- $PI^\lambda D^\mu$ kontrollörler için birim basamak çıkış cevapları.....	64
<b>Şekil 4.3.</b>	Azalan $K$ değerleri için MRA-PID kontrollörün sistem birim basamak çıkış cevapları.....	65
<b>Şekil 4.4.</b>	Azalan $K$ değerleri için MRA- $PI^\lambda D^\mu$ kontrollörün sistem birim basamak çıkış cevapları.....	65
<b>Şekil 4.5.</b>	$PI^\lambda D^\mu$ kontrollör için sistemin birim basamak cevabı.....	67
<b>Şekil 4.6.</b>	MRA-PID ve MRA- $PI^\lambda D^\mu$ kontrollörler için birim basamak çıkış cevapları ( $K = 1$ ).....	67
<b>Şekil 4.7.</b>	Azalan $K$ değerleri için MRA-PID kontrollörün sistem birim basamak çıkış cevapları.....	68

<b>Şekil 4.8.</b>	Azalan $K$ değerleri için MRA-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolörün sistem birim basamak çıkış cevapları.....	68
<b>Şekil 4.9.</b>	DC motorun eşdeğer devresi.....	69
<b>Şekil 4.10.</b>	FLC ile DC motor hız kontrol blok diyagramı.....	70
<b>Şekil 4.11.</b>	$K$ (0.01), $J$ (0.01), $L$ (0.5) ve $b$ (0.1) için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> , FLC ve ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör sistemlerinin birim basamak cevapları.....	71
<b>Şekil 4.12.</b>	Artan $K$ değerleri için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör birim basamak cevapları.....	71
<b>Şekil 4.13.</b>	Artan $K$ değerleri için FLC birim basamak cevapları.....	72
<b>Şekil 4.14.</b>	Artan $K$ değerleri için ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör birim basamak cevapları.....	72
<b>Şekil 4.15.</b>	Artan $J$ değerleri için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör birim basamak cevapları.....	73
<b>Şekil 4.16.</b>	Artan $J$ değerleri için FLC birim basamak cevapları.....	73
<b>Şekil 4.17.</b>	Artan $J$ değerleri için ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör birim basamak cevapları.....	74
<b>Şekil 4.18.</b>	Artan $L$ değerleri için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör birim basamak cevapları.....	74
<b>Şekil 4.19.</b>	Artan $L$ değerleri için FLC birim basamak cevapları.....	75
<b>Şekil 4.20.</b>	Artan $L$ değerleri için ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör birim basamak cevapları.....	75
<b>Şekil 4.21.</b>	Artan $b$ değerleri için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör birim basamak cevapları.....	76
<b>Şekil 4.22.</b>	Artan $b$ değerleri için FLC birim basamak cevapları.....	76
<b>Şekil 4.23.</b>	Artan $b$ değerleri için ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör birim basamak cevapları.....	77
<b>Şekil 4.24.</b>	FLC içeren ısı değiştirici sistemin kontrol blok diyagramı.....	79
<b>Şekil 4.25.</b>	Aynı DC kazancı için $k = 0.068$ , zaman sabiti $a = 23$ ve zaman gecikmesi $T = 19.4$ için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> , FLC ve ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolörlerin birim basamak cevapları.....	79
<b>Şekil 4.26.</b>	Artan DC kazanç ( $k$ ) değerleri için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolörünün birim basamak cevapları ( $K_p=20.50$ , $K_i=1.22$ , $K_d=80$ , $\lambda=80$ , $\mu=0.80$ ).....	80
<b>Şekil 4.27.</b>	Artan DC kazanç ( $k$ ) değerleri için FLC' nin birim basamak cevapları	80
<b>Şekil 4.28.</b>	Artan DC kazanç ( $k$ ) değerleri için ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolörünün birim basamak cevapları.....	81
<b>Şekil 4.29.</b>	Artan zaman sabitleri ( $a$ ) için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolörünün birim basamak cevapları.....	82
<b>Şekil 4.30.</b>	Artan zaman sabitleri ( $a$ ) için FLC' nin birim basamak cevapları.....	82



<b>Şekil 4.31.</b>	Artan zaman sabitleri ( $a$ ) için ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolörünün birim basamak cevapları.....	83
<b>Şekil 4.32.</b>	Artan zaman gecikme sabitleri ( $T$ ) için PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolörünün birim basamak cevapları.....	84
<b>Şekil 4.33.</b>	Artan zaman gecikme sabitleri ( $T$ ) için FLC' nin birim basamak cevapları.....	84
<b>Şekil 4.34.</b>	Artan zaman gecikme sabitleri ( $T$ ) için ANFIS-PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolörünün birim basamak cevapları.....	85
<b>Şekil 4.35.</b>	MATLAB – Simulink’ te otomatik ayarlamalı sinir hücresi ile P kontrolör blok diyagramı.....	86
<b>Şekil 4.36.</b>	Artan $\tau$ değerleri için (1/4, 1, 4) kontrol sisteminin birim basamak cevapları.....	87
<b>Şekil 4.37.</b>	Artan $K$ değerleri için (1/4, 1, 4) kontrol sisteminin birim basamak cevapları.....	87
<b>Şekil 4.38.</b>	MATLAB – Simulink’ te otomatik ayarlamalı sinir hücresi ile PI <sup><math>\lambda</math></sup> kontrolör blok diyagramı.....	88
<b>Şekil 4.39.</b>	Artan $\tau$ değerleri (1/4, 1, 4) için kontrol sisteminin birim basamak cevapları.....	89
<b>Şekil 4.40.</b>	Artan $K$ değerleri (1/4, 1, 4) için kontrol sisteminin birim basamak cevapları.....	89
<b>Şekil 4.41.</b>	MATLAB – Simulink’ te otomatik ayarlamalı sinir hücresi ile PD <sup><math>\mu</math></sup> kontrolör blok diyagramı.....	90
<b>Şekil 4.42.</b>	Artan $\tau$ değerleri (1/4, 1, 4) için kontrol sistemin birim basamak cevapları.....	90
<b>Şekil 4.43.</b>	Artan $K$ değerleri (1/4, 1, 4) için kontrol sistemin birim basamak cevapları.....	91
<b>Şekil 4.44.</b>	MATLAB – Simulink’ te otomatik ayarlamalı sinir hücresi ile PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör sistemin blok diyagramı.....	92
<b>Şekil 4.45.</b>	Artan $\tau$ değerleri (1/4, 1, 4) için kontrol sistemin birim basamak cevapları.....	92
<b>Şekil 4.46.</b>	Artan $K$ değerleri (1/4, 1, 4) için kontrol sistemin birim basamak cevapları.....	93
<b>Şekil 4.47.</b>	Her iki yöntemle göre PI <sup><math>\lambda</math></sup> D <sup><math>\mu</math></sup> kontrolör sisteminin birim basamak cevapları.....	94

## ÇİZELGELER DİZİNİ

<b>Çizelge 2.1.</b>	S - cevabına göre birinci parametre ayarlama çizelgesi [70].....	15
<b>Çizelge 2.2.</b>	S – cevabına göre ikinci parametre ayarlama çizelgesi [70].....	15
<b>Çizelge 2.3.</b>	Kritik kazanç ve periyoda göre birinci parametre ayarlama çizelgesi [70].....	16
<b>Çizelge 2.4.</b>	Kritik kazanç ve periyoda göre ikinci parametre ayarlama çizelgesi [70].....	16
<b>Çizelge 2.5.</b>	Kritik kazanç ve periyoda göre üçüncü parametre ayarlama çizelgesi [70].....	16
<b>Çizelge 3.1.</b>	Farklı $r$ değerleri için $D_r(z)$ ' in pay ve paydası [131].....	54

## 1. GİRİŞ

Kontrol mühendisliği; elektrik, elektronik, mekanik ve bilgisayar tabanlı tüm endüstriyel üretim sistemlerinin ve hizmet sektörünün amaçlanan ve planlanan biçimde çalışmasını sağlayan bilgi ve teknolojileri üreten ve uygulayan bir mühendislik alanıdır. Bu hedef için klasik kontrol, dijital kontrol, nonlinear kontrol, dayanıklı kontrol (robust control), uyarlamalı (adaptif) kontrol, akıllı kontrol (bulanık mantık, sinir ağları) gibi farklı kontrol metotları ile sistemlerin kontrolü gerçekleştirilir.

PID (Proportional-Integral-Derivative Oransal-İntegral-Türevsel) kontrolör, basit yapısı ve dayanıklı performans özellikleri nedeniyle ile çokça tercih edilmektedir. Bu kontrolörlerin parametrelerini ( $K_p$ ,  $K_i$ ,  $K_d$ ) hesaplamak için bir çok metot mevcuttur [1-4]. Ziegler-Nichols metotları, Cohen-Coon kuralları, Åström-Hägglund yöntemi ve gelişmiş Ziegler-Nichols yöntemi PID kontrolörün katsayılarını hesaplayan yöntemlerden bir kısmını oluştururlar.

Kesirli dereceli operatörlerin kullanılması fikri çok eskilere dayanmakla birlikte, son yirmibeş yılda birçok araştırma alanında kullanılmaya başlanması ile çok popüler olmuştur. Otomatik kontrolde Oustaloup [5] dinamik sistemlerin kontrolü için kesirli dereceli bir algoritma geliştirmiş ve CRONE (“Comande Robuste D’Ordere Non Entier” “Kesir Dereceli Dayanıklı Kontrol”) olarak adlandırdığı sistemin PID kontrolöre karşı performansı üzerinde çalışmıştır. Podlubny, klasik PID kontrolörü kesir dereceli PID ( $PI^\lambda D^\mu$ ) kontrolör yapısına dönüştürüp bu kontrolörün klasik PID kontrolöre göre daha iyi bir performans sergilediğini göstermiş ve dinamik sistemlerin kontrolünde, kesir dereceli kontrolörler kullanılmaya başlanmıştır [6]. Fakat bu çalışmalar hesapsal güçlüklerden dolayı kısıtlı kalmıştır. Daha sonraki yıllarda kesir dereceli kontrol yapıları ile ilgili çalışmalar artmıştır [7 -13].

Çok farklı uygulama çalışmalarında, kesir dereceli kontrolör ve kontrol sistemleri kullanılmaya başlanmıştır. Örneğin [14]’de, hareket kontrol uygulamaları için kesir dereceli kontrol tartışılmıştır. Altı-ayaklı bir robot için kesir dereceli kontrol uygulaması [15]’ de verilmiştir. Esnek şaftlı bir DC motorun kesir dereceli PID ( $PI^\lambda D^\mu$ ) ile kontrol edilmesi ile ilgili bir çalışma [16]’da verilmiştir. Esnek uzay aracı yükseklik kontrolü için kesir dereceli kontrolör kullanılması [17]’de önerilmiştir. [18]’de kesirli frekans iletim sistemleri üzerinde bir deneysel çalışma yapılmıştır. Kesir dereceli Wien Köprü Osilatörü [19]’da önerilmiştir. Kesirli kapasitörler kullanılarak güç faktörü (P.F.-Power Factor) düzeltilmesi [20]’de çalışılmıştır. Smith Predictor kullanılarak kesirli dalga

denkleminin kesir dereceli sınır kontrolü [21]'de verilmiştir. Literatür taramasından da görüldüğü gibi bu konuda yapılan pratik uygulamalar gün geçtikçe artmaktadır.

Kesir dereceli kontrolörün parametrelerini hesaplamak için birçok yöntem mevcuttur [22-33]. Bu yöntemlerden en çok kullanılanlardan ilki, S-cevabına dayalı ayarlama kuralları, ikinci ise kritik kazanca dayalı ayarlama kurallarıdır [24-26].

Uyarlamalı kontrol, ilk defa 1952' de Draper ve Li tarafından düşünülmüştür [34]. 1995'de Astrom ve Wiltenmark konuyla ilgili uçak uçuş kontrol sistemini uyarlamalı olarak düşünerek önemli adım atmışlardır [35]. İstenen ve gerçek olan işaretler arasındaki hata işaretini referans model kullanarak elde etmişler ve bu hata işareti, sisteme istenen davranışı yaptıracak şekilde kontrolör parametrelerinin değiştirilmesinde kullanılmıştır. Bu tür sistemler, Model Referans Uyarlamalı Kontrol sistemleri (MRAC) olarak belirtilir. 1960'da Li ve Van Der Velden, kontrol çevrimindeki limit çevrim tarafından yaratılan parametre belirsizliklerinin kompanzasyonuna dayanan bir başka uyarlamalı kontrol sistemi üzerinde çalıştılar [36]. Astrom ve Wiltenmark, uyarlamalı kontrolün diğer bir önemli konusu olan kendi kendini ayarlayabilen sistemleri (Self-Tuning Regulators, STR) 1971'de geliştirmişlerdir [35-37]. Bu çalışmalardan sonra STR' lere olan ilgi bütün dünyada hızla artmıştır. Diğer önemli bir hamle, 1974' de Monopoli'nin kararlı MRAC tasarımına argüman hatası (argumented error) yaklaşımını tanımlamasıdır [38]. Ayrıca, hem sürekli zaman hemde ayrık-zaman uyarlamalı sistemler için pek çok çalışmalarda yapılmıştır [39-44]. Sadece kararlılık sorunu değil, aynı zamanda uyarlamalı kontrole farklı yaklaşımların birleştirilmesi ve MRAC ile STR sistemlerinin gerçekleştirilmesi sorunu ile karşılaşmıştır. Fakat uyarlamalı kontrol günümüzde dikkate değer teorik ve algoritmik ilerlemelerle, oldukça yüksek bir olgunluk seviyesine ulaşmıştır ve büyük bir uygulama potansiyeline sahiptir [45].

Bulanık mantık 1965 yılında University of California, Berkeley'den Dr. Lotfi Zadeh tarafından, doğal dildeki belirsizliği modellemek için ortaya konmuştur [46]. Bulanık mantık ilk kez 1973 yılında, H. Mamdani ve S. Assilian tarafından bir buhar makinesinde uygulanmıştır [47].

Yapay Sinir Ağları (YSA)'nın gelişiminin başlangıç yılı 1942 olarak kabul edilmektedir. McCulloch ve Pitts ilk hücre modelini bu tarihte geliştirmişlerdir [48]. Bununla birlikte birkaç hücrenin ara bağlaşımını incelemişlerdir. 1949 yılında Hebb hücre bağlantılarını ayarlamak için ilk öğrenme kuralını önermiştir [49]. Günümüzde kullanılan kuralların temeli, algılayıcı (perceptron) modelini ve öğrenme kuralını

geliştiren Rosenblatt tarafından 1958'de atılmıştır [50]. Widrow ve Hoff 1960-1962 yılında, ADALINE (Adaptive Linear Neuron) ve LMS kuralını geliştirmiştir [51]. 1970 yılında, Minsky ve Papert, algılayıcının kesin analizini yapmıştır ve algılayıcının karmaşık lojik fonksiyonlar için kullanılmayacağını ispatlamışlar [52]. Bu tarihten sonra sinir ağları üzerine yapılan çalışmalar azalmış ve durma noktasına gelmiştir. Bu durgunluk 1960'dan 1990'lara kadar devam etmiştir. Bu durgunluğun en önemli sebeplerinden birisi, YSA' nın bilgi işlemede alternatifi olan günümüzün sayısal bilgisayarlarının, yarı iletken teknolojisi ile yoğun, büyük çapta ucuz ve güvenilir gerçekleştirme imkânını bulmamasıdır. Seri olarak çalışan hızlı birimlerden oluşmuş sayısal bilgisayarlar, aritmetik işlemlerde yüksek hız, kapasite ve güvenilirlik sağlamışlardır. Ancak tüm bunların yanında bazı bilim adamları (Grossberg, Amari, Fukushima, Kohonen, Taylor, ...) çalışmalarına devam etmişlerdir [53-58]. Hopfield ve Tank 1985 yılında YSA' nın birçok problemi çözebilecek kabiliyette olduğunu göstermişlerdir [59]. Optimizasyon gibi teknik problemleri çözmek için doğrusal olmayan Hopfield ağını geliştirmişlerdir. 1982-1984 yılında Kohonen, öz düzenlemeli haritayı tanımlamıştır [57] ve günümüzde kendi adıyla anılan eğitimcisz öğrenen bir ağ geliştirmiştir. Rumelhart' da 1986 yılında geriye yayılım algoritmasını bulmuştur [60]. Chua ve Yang, 1988 yılında hüresel sinir ağlarını geliştirmişlerdir [61].

YSA' nın kontrol mühendisliği alanındaki uygulamalarına örnek olarak; esnek kollu robotun kontrolü, bir model helikopterin kontrolü, çok değişkenli (mafsallı) robotun yörünge koordinasyonu, iki sıvı tank sisteminin akış seviye kontrolü, sıcaklık kontrolü, endüstriyel robot kontrolü, sualtı araç kontrolü, evlerde enerji yönetimi için yapay sinir ağı tabanlı kontrol verilebilir [62-69].

Kontrolör tasarımı için kontrol parametrelerin hesaplanması gerekir. Bu konuda literatürde kesir dereceli PI, PD veya PID kontrolörlerin parametrelerinin hesaplanması ve kontrolör tasarımları ile ilgili çalışmalar vardır [22-33, 70-73]. Sinirsel-bulanık mantık (neuro-fuzzy) veya sinir ağı sistemleri kullanarak, çeşitli sistemler için çevrimiçi (on-line) veya çevrimdışı (offline) öğrenme algoritmaları ile  $PI^{\lambda}D^{\mu}$  kontrolörün katsayıları ayarlanabilmiştir [25, 74-77]. Entegre şasi kontrolünde ANFIS ve  $PI^{\lambda}D^{\mu}$  kontrolör algoritmaları kullanılarak gerçekleştirilen ortak simülasyonlarda, entegre şasi kontrol sisteminde aracın denge ve güvenliği arttırılabilmıştır [78].

Performans, kararlılık ve dayanıklılık için pek çok çalışma yapılmış olmakla birlikte yapılan çalışmaların çoğu tamsayı dereceli sistemler içindir. [79]'da bulanık mantık kullanılarak tasarlanan PID kontrolör ile doğrusal olmayan sistemlerin kontrolü

gerçekleştirilmiştir. Tek Giriş–Tek Çıkış zamanla değişmeyen lineer bir sistemin, kararlılığını ve dayanıklılığını sağlayan PID kontrolörün bir grafik tekniği ile tasarımı [80]'de verilmiştir. Tek Giriş–Tek Çıkış zaman gecikmeli bir sistemin, sağlam bir performansla kontrolünü sağlayan PID kontrolörün Oransal-İntegral-Türev katsayılarını bulmak için sadece sistemin frekans cevabına bağlı kalan bir yöntem [81]'de verilmiştir.

Bir kontrol sisteminin performansının iyi, kararlı ve dayanıklı olması daima tercih edilir. Literatürde kesir dereceli kontrol sistemlerinin kararlılığını, performansını inceleyen çalışmalar mevcuttur [42, 74, 82-86]. Bu çalışmaların bir çoğunda, tamsayı dereceli kontrol sistemlerinde kullanılan tekniklerin benzeri, kesirli dereceli kontrol sistemlerine de uygulanmış veya kesir dereceli kontrol sistemlerinde yeni teknikler geliştirilmiştir.

Bu konuda kullanılabilir yeni tekniklerin bulunması ve geliştirilmesi için bir çok araştırmalar yapılmıştır [13, 87-89]. Ayrıca bu çalışmalarda sinirsel-bulanık mantık veya sinir ağları ile daha pratik uygulanabilir ve kolay eğitilebilir sistemler üzerinde çalışılmıştır.

Bir çok araştırmacı kesir dereceli kontrol sistemleri için benzetim ve uygulama çalışmaları yapmıştır [8, 9, 90]. Yapılan çalışmaların bir kısmı, kesir dereceli geçiş fonksiyonlarının ya da kontrolörlerin tamsayı dereceli karşılığının elde edilebilmesi üzerine yoğunlaşmaktadır. Bu konuda geliştirilecek her yeni teori, yapılacak her çalışma kesir dereceli kontrol sistemlerinin kullanılabilirliğine ve uygulanabilirliğine katkı sağlayacaktır. Yapılan çalışmalarda, kesir dereceli sistemlerin tamsayı dereceli sistemlere göre teorik olarak daha iyi benzetim sonuçları verdiği görülse de, bu sistemlerin kullanılabilir olması için reel olarak gerçekleştirilmesine ihtiyaç vardır.

Robot uygulamalarından motor hız kontrolüne, Wien köprüsü osilatöründen güç faktörü (P.F.) düzeltilmesine kadar çok geniş bir yelpazede, kesir dereceli kontrolörler ve kesir dereceli kontrol sistemleri ile ilgili çalışmalar yapılmıştır [15-20]. Kesir dereceli işlem yapabilen elemanların geliştirilmesi ile kontrol uygulamalarının her alanında kesir dereceli sistemlerin performansı test edilmeye başlanmıştır [84-86]. Dolayısı ile bu konuda elde edilecek sonuçların pratik sistemlerde uygulanması ve performanslarının test edilmesi önem taşımaktadır. Bu nedenle elde edilecek sonuçlar kesir dereceli kontrolörlerin endüstride yaygın kullanımına yardımcı olacaktır.

Kesir dereceli matematiğin teorik alt yapısının daha güçlenmesi, bu konunun çok farklı alanlara uyarlanmasını sağlayacaktır. Çok farklı alanlarda kesir dereceli matematiğin ve kesir dereceli kontrol sistemlerinin uygulandığı görülmektedir [15-20,

91, 92]. Kesir dereceli kontrol sistemleri alanında yapılan çalışmalardan bazıları; nonlinear kontrol sistemlerinde kaos, durum uzay modelinin incelenmesi, kontrol sistemlerinde bozulmaların bastırılması, işaret işleme uygulamaları, ayırık kontrol uygulamaları, zaman gecikmeli sistem uygulamaları, uyarlamalı kontrol, yapay sinir ağları, zaman tabanlı analiz gibi çalışmalardır.

Sinirsel-bulanık mantık ve sinir ağları sistemlerde gerek kontrolör kısmında, gerekse kontrol edilen sistem bölümünde bugüne kadar tamsayı dereceli sistemler kullanılmıştır, günümüze kadar kesir dereceli sistemleri kullanan çok az çalışma mevcuttur [87-89, 93-96]. Bu çalışmada kontrolör kısmında kesirli yapı kullanarak çalışmalar yapılmıştır. Kontrolör kısmına sinirsel-bulanık mantık veya sinir ağları eklenerek daha iyi performansa sahip, daha dayanıklı bir sistem tasarlanması hedeflenmiştir. Bu konunun içerisinde sinir ağları, bulanık mantık ve kesir dereceli hesaplama birleştirilerek daha geniş bir teori tabanı oluşturulmaya çalışılmıştır.

Bu tezde, model referans uyarlamalı kesir dereceli ( $MRU-PI^{\lambda}D^{\mu}$ ), ANFIS ile kesirli dereceli ( $ANFIS-PI^{\lambda}D^{\mu}$ ) ve otomatik ayarlamalı sinir hücresi ile kesirli dereceli PID kontrolörler tasarlanmış, benzetim uygulamaları ile sistemlerin kontrolü gerçekleştirilmiştir. Bu yöntemler ile tasarlanan kontrol sistemlerine birim basamak giriş fonksiyonu uygulanmış ve çıkış cevapları alınmıştır. Performans ve dayanıklılık eğrileri elde edilmiş ve çıkan sonuçların karşılaştırılmasına göre hangi kontrol sisteminin daha dayanıklı ve daha iyi bir performansa sahip olduğunun belirlenmesinin yanı sıra hangi sistemin daha kolay, daha pratik ve daha uygulanabilir olabileceği değerlendirilmiştir. Geliştirilen bu yöntemlerin farklı kontrol sistemlerine uyarlanması ile ilgili çalışmalar yapılmıştır. Tasarımların benzetim sonuç grafikleri ve çıktıları gözlemlenerek yapılan değerlendirme sonucunda tasarım metotlarının pratiklik, dayanıklılık ve hız performansları değerlendirilmiş ve bu tezde önerilen yöntemlerin başarılı olduğu belirlenmiştir.

Bu tez çalışmasında, yukarıda adı geçen metotlardan uyarlamalı kontrol ile akıllı kontrol (bulanık mantık, sinir ağları) birlikte ve/veya ayrı kullanılarak kesir dereceli kontrolörler tasarlanmıştır. Bu kontrolörler ile tamsayı sistemlerin kontrolündeki performansları incelenmiş ve en uygun modelin belirlenmesine çalışılmıştır.

Tezde yapılan çalışmalar kesir dereceli kontrol sistemleri için kontrolör tasarımı, parametre belirleme teknik ve yöntemleri, performans ve dayanıklılık analizleri şeklinde sıralanabilir.

Tez beş bölümden oluşmaktadır. Birinci bölümde literatür taraması ile yapılan çalışmalar incelenmiş ve bu tezdeki tasarımlar tanıtılmıştır. İkinci bölümde kesir dereceli kontrol sistemleri, bulanık mantık teorisi, yapay sinir ağları hakkında kuramsal temeller anlatılmıştır. Üçüncü bölümde tezde kullanılan yöntemler verilmiştir. Dördüncü bölümde elde edilen sonuçlar değerlendirilip yorumlanmıştır. Beşinci bölümde ise sonuç ve tartışmalara yer verilmiştir.



## 2. KURAMSAL TEMELLER

### 2.1. Kesir Dereceli Sistemler

Türev ve integral derecelerinin herhangi bir reel sayı olduğu, tamsayı olma zorunluluğunun bulunmadığı diferansiyel denklemlerle gösterilen sistemler kesir dereceli sistemler olarak tanımlanır.

Kesir dereceli integro-diferansiyel operatörü Denklem (2.1)'deki gibi tanımlanmaktadır [97, 98].

$${}_a D_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \text{Re}(\alpha) > 0 \\ 1 & \text{Re}(\alpha) = 0 \\ \int_a^t (d\tau)^{-\alpha} & \text{Re}(\alpha) < 0 \end{cases} \quad (2.1)$$

Burada;  $a$  ve  $t$ : integralin alt ve üst sınırlarını  $\alpha$ : türev veya integral derecesini göstermektedir. Kesir dereceli cebirsel işlemler için farklı tanımlar verilmektedir. Bunlardan en fazla kullanılan denklemler Denklem (2.2 - 2.10) ile gösterilmiştir. Denklem (2.2-2.6)' da Grünwald–Letnikov kesir dereceli türev denklemleri verilmiştir.  $\alpha$  dereceli bir integro-diferansiyel denklem aşağıdaki şekilde ifade edilir:

$${}_a D_t^\alpha f(x) = \lim_{h \rightarrow \infty} \frac{\Delta_h^\alpha f(t)}{h^\alpha} \quad (2.2)$$

Burada;  $h$  adım sayısı olup,

$$\Delta_h^\alpha f(t) = \sum_{j=0}^{\infty} (-1)^j \binom{\alpha}{j} f(t - jh) \quad (2.3)$$

şeklinde ifade edilirse ve toplam işleminin  $r$  gibi sonlu bir sayıya kadar yapılması durumunda;

$$\sum_{j=0}^r (-1)^j \binom{\alpha}{j} = \frac{1}{\Gamma(1-\alpha)} \frac{\Gamma(r+1-\alpha)}{\Gamma(r+1)} \quad (2.4)$$

olacaktır. Eğer  $h = \frac{t-a}{r}$  olarak kabul edilirse Denklem (2.2)

$${}_a D_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{\lfloor (t-a)/h \rfloor} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (2.5)$$

$${}_a D_t^\alpha f(t) = \lim_{h \rightarrow 0} \left\{ \frac{\left( \frac{t-a}{r} \right)}{\Gamma(-\alpha)} \sum_{j=0}^{r-1} \frac{\Gamma(j-\alpha)}{\Gamma(j+1)} f\left( t - j \left( \frac{t-a}{r} \right) \right) \right\} \quad (2.6)$$

şekline dönüşür. Burada  $\Gamma(\cdot)$  Euler gama fonksiyonudur. Riemann-Liouville kesir dereceli integral ifadesi, Denklem (2.7)'deki gibi tanımlanır [98].

$${}_a D_t^\alpha f(t) = \frac{1}{-\alpha} \int_a^t (t-\tau)^{-\alpha-1} f(\tau) d\tau, \quad \alpha < 0 \quad (2.7)$$

$n-1 \leq \alpha < n$  olmak şartıyla Denklem (2.7)'nin  $n$ . dereceden türevi alınırsa

$${}_a D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_a^t (t-\tau)^{n-\alpha-1} f(\tau) d\tau, \quad n > 0 \quad (2.8)$$

olur. Denklem (2.8),  $f(t)$  fonksiyonunun  $(n-\alpha)$ . dereceden kesir dereceli türevine karşılık gelmektedir. Burada  $n$  tamsayıdır. Denklem (2.4, 2.6 ve 2.8)'de kullanılan Euler gama fonksiyonu,

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, \quad t > 0 \quad (2.9)$$

şeklinindedir. Burada  $z \geq 0$  ve bir tamsayıdır. Denklem (2.9)'un  $z+1$  için integrali alınırsa;

$$\Gamma(z+1) = \int_0^\infty e^{-t} t^{(z+1)-1} dt = \int_0^\infty e^{-t} t^z dt = \left[ -e^{-t} t^z \right]_{t=0}^{t=\infty} + z \int_0^\infty e^{-t} t^{z-1} dt = z\Gamma(z) \quad (2.10)$$

olur.

## 2.2. Kesir Dereceli Türev ve İntegralin Laplace Dönüşümü

Bir  $f(t)$  fonksiyonunun Laplace dönüşümü Denklem (2.11)' deki gibidir.

$$F(s) = L\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt \quad (2.11)$$

$\alpha$  kesir dereceli türev veya integral operatörünün Laplace dönüşümü, kontrol sistemlerinin ve modellerin transfer fonksiyonlarının bulunmasında ve analizlerinin yapılmasında kolaylık sağlar.  ${}_0D_t^\alpha f(t)$  şeklinde  $\alpha$  kesir derecesine sahip bir ifadenin Laplace dönüşümü ise;

$$L\{{}_0D_t^\alpha f(t)\} = \int_0^{\infty} e^{-st} \frac{d^\alpha f(t)}{dt^\alpha} dt \quad (2.12)$$

olur [2]. Bu ifadenin kesir dereceli türevi, Denklem (2.13)' de verilmiştir.

$$L\{{}_0D_t^\alpha f(t)\} = s^\alpha F(s) - \sum_{k=0}^{n-1} s^k {}_0D_t^{\alpha-k-1} f(0) \quad (2.13)$$

Başlangıç şartları sıfır alınır, kesir dereceli türevin Laplace dönüşümü,

$$L\{{}_0D_t^\alpha f(t)\} = s^\alpha F(s) \quad (2.14)$$

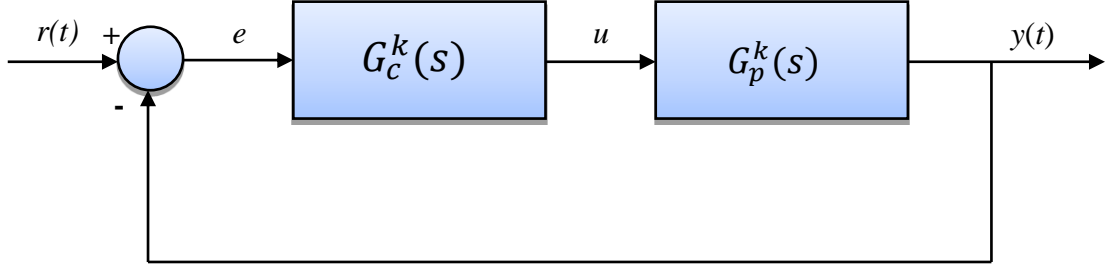
olur. Kesir dereceli integralin Laplace dönüşümü ise,

$$L\{{}_0I_t^\alpha f(t)\} = s^{-\alpha} F(s) \quad (2.15)$$

şeklindedir. Gama fonksiyonunun karmaşık çarpma işlemlerinden kurtulmak için kesir dereceli sistemlerin analizinde, Laplace dönüşümü kullanılır.

### 2.3. Birim Geri Beslemeli Kesir Dereceli Kontrol Sistemleri

Tek Girişli Tek Çıkışlı (TGTC) birim geri beslemeli kesir dereceli kontrol sistemi, Şekil 2.1' deki gibidir.



Şekil 2.1. Birim geri beslemeli kesir dereceli kontrol sistemi

Burada;  $G_c^k(s)$  : Denklem (2.16)'da tanımlanan kesir dereceli kontrolörü,  $G_p^k(s)$  : Denklem (2.17)'de gösterildiği gibi tam dereceli veya kesir dereceli sistemi temsil etmektedir.

$$G_c^k(s) = \frac{b_1 s^m + b_2 s^{m-1} + \dots + b_m s + b_{m+1}}{s^n + a_1 s^{n-1} + a_2 s^{n-2} + \dots + a_{n-1} s + a_n} \quad m, n: \text{gerçek sayılardır} \quad (2.16)$$

$$G_p^k(s) = \frac{b_1 s^{\beta_1} + b_2 s^{\beta_2} + \dots + b_m s^{\beta_m}}{a_1 s^{\alpha_1} + a_2 s^{\alpha_2} + \dots + a_n s^{\alpha_n}} \quad \beta_i \text{ ve } \alpha_j \quad (i=0 \dots m, j=0 \dots n) \quad (2.17)$$

Sistemin, kapalı çevrim kesir dereceli bir kontrol sistemi olabilmesi için kontrolörün veya kontrol edilen sistemlerden en az birinin kesir dereceli olması gerekmektedir. Şekil 2.1'de verilen TGTC kesir dereceli transfer fonksiyonu, Denklem (2.18) ve (2.19)'da verilen diferansiyel denklemler ile ifade edilebilir:

$$\frac{y(t)}{r(t)} = \frac{b_0 D^{\beta_0} + b_1 D^{\beta_1} + b_2 D^{\beta_2} + \dots + b_m D^{\beta_m}}{a_0 D^{\alpha_0} + a_1 D^{\alpha_1} + a_2 D^{\alpha_2} + \dots + a_n D^{\alpha_n}} \quad (2.18)$$

$$a_n D^{\alpha_n} y(t) + \dots + a_1 D^{\alpha_1} y(t) + a_0 D^{\alpha_0} y(t) = b_m D^{\beta_m} r(t) + \dots + b_1 D^{\beta_1} r(t) + b_0 D^{\beta_0} r(t) \quad (2.19)$$

Burada  $a_n > \dots > a_1 > a_0$ ,  $\alpha_n > \dots > \alpha_1 > \alpha_0$  olmak şartıyla  $\alpha_k$  ve  $\beta_k$  ( $k = 0,1,2,\dots$ ) rastgele reel sayılar ve  $a_k, b_k$  ( $k = 0, 1, 2,\dots$ ) katsayılarıdır. Laplace dönüşümü ile Denklem (2.18)'in  $s$  domenindeki ifadesi,

$$G^k(s) = \frac{Y(s)}{R(s)} = \frac{b_m s^{\beta_m} + \dots + b_1 s^{\beta_1} + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + \dots + a_1 s^{\alpha_1} + a_0 s^{\alpha_0}} \quad (2.20)$$

olur.  $G^k(s)$ , Şekil 2.1'de verilen kesir dereceli kontrol yapısının transfer fonksiyonudur.

#### 2.4. Birim Geri Beslemeli Kesir Dereceli PID ( $PI^{\lambda}D^{\mu}$ ) Kontrol Sistemi

Kontrol sistemlerinde PID kontrolörler, basit yapısı ve kolay tasarımı nedeni ile fazla tercih edilmektedir [2-4]. PID kontrolörler oransal, integral ve türev operatörlerinden oluşur. Oransal operatör sistemde oluşan anlık hatayı, integral operatörü geçmişte sistemde oluşan hataların toplamını ve türev operatörü ise hatanın zamana bağlı ivmesini ifade eder. Bu kontrolörler; otomotiv, uçuş kontrolü, motor sürücüler, manyetik ve optik hafızalar gibi farklı birçok alanda uygulanmaktadır [68, 99, 100]. Tam dereceli bir PID kontrolörün transfer fonksiyonu Denklem (2.21) ve (2.22)'daki gibidir:

$$u(t) = K_p e(t) + K_i D^{-1} e(t) + K_d D e(t) \quad (2.21)$$

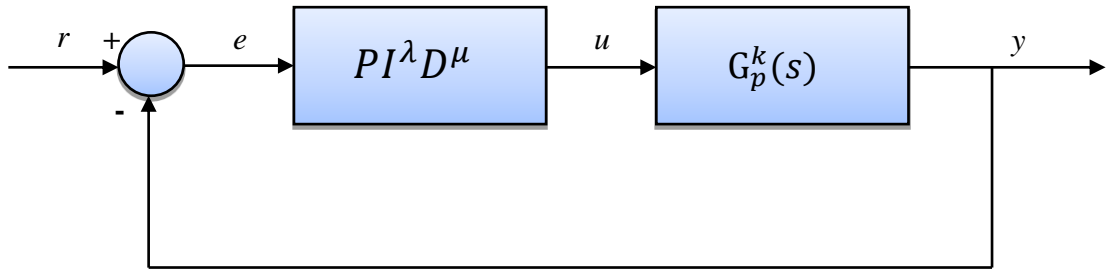
$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (2.22)$$

Burada  $K_p$  oransal,  $K_i$  integral,  $K_d$  ise türev operatörlerinin katsayılarıdır. PID kontrolörün  $K_i$  veya  $K_d$  katsayılarından birinin sıfır olması durumunda, PD veya PI kontrolör yapısına dönüşürler. Bu iki kontrolör toplandığında ise klasik PID kontrolöre dönüşmektedir. PD kontrolörde  $K_d$  katsayısının değişmesi durumunda sistem birim basamak cevabının sönümlenme oranını ve yükselme zamanını etkilemektedir. Fakat kalıcı durum hatası üzerinde etkisi yoktur. PI kontrolör ise  $K_i$  katsayısının değişmesi durumunda sistem birim basamak cevabının yükselme zamanını ve kalıcı hatasını etkilemektedir. Kontrol edilen sistemin çıkış cevabının istenen performansa sahip

olabilmesi için PID kontrolörün  $K_p$ ,  $K_i$  ve  $K_d$  katsayılarının sistemin davranışına göre belirlenmesi gerekir. Şekil 2.2'deki PID kontrolörün integral ve türev dereceleri kesirli olarak alınırsa transfer fonksiyonları Denklem (2.23), (2.24)'deki gibi ifade edilir.

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) \quad (2.23)$$

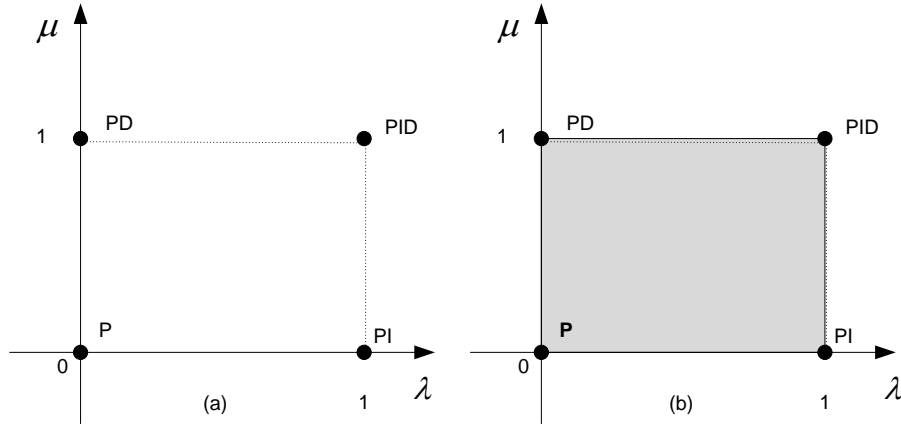
$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (2.24)$$



**Şekil 2.2.** Birim geri beslemeli kesir dereceli PID ( $PI^\lambda D^\mu$ ) kontrol sistemi

Burada;  $K_p$ : oransal,  $K_i$ : kesir dereceli integral,  $K_d$ : ise kesir dereceli türev operatörlerinin katsayılarıdır,  $0 < (\lambda, \mu) < 1$  olmak şartıyla,  $\lambda$ : integral ve  $\mu$ : türev operatörlerinin kesir dereceleridir.  $\lambda$  ve  $\mu$  reel sayılardır.  $\lambda = 0, \mu = 0$  için P kontrolör,  $\lambda = 1, \mu = 0$  için PI kontrolör,  $\lambda = 0, \mu = 1$  için PD kontrolör,  $\lambda = 1, \mu = 1$  için PID kontrolör elde edilir. Kesir dereceli PID ( $PI^\lambda D^\mu$ ) kontrolör, Şekil 2.1'deki yapıda  $G_c^k(s)$  olarak seçilirse, birim geri beslemeli kontrol sistemi kesir dereceli bir kontrol sistemine dönüşecektir. Kesir dereceli PID ( $PI^\lambda D^\mu$ ) kontrolörler, tam dereceli sistemlerde ve ayrıca kesir dereceli sistemlerin kontrolünde de kullanılabilir [6]. Bu kontrolörlerin, tam dereceli PID kontrolörlere göre kontrol kalitesinin ve dayanıklılığının daha iyi olduğu tesbit edilmiştir [7, 9].

Şekil 2.3. (a) ve (b)'de sırasıyla tam ve kesir dereceli PID kontrolörün  $\lambda$ - $\mu$  düzlemindeki gösterilimi verilmiştir.

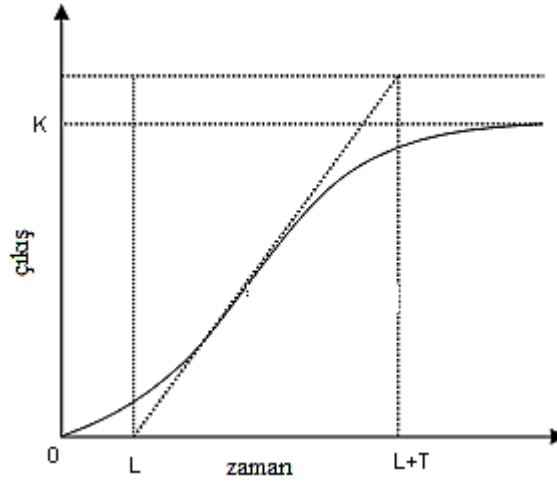


**Şekil 2.3. (a) Tam dereceli PID kontrolör için  $\lambda$ - $\mu$  düzlemi (b) Kesir dereceli PID kontrolör için  $\lambda$ - $\mu$  düzlemi**

Tam dereceli PID kontrolör  $\lambda$ - $\mu$  düzlemi üzerinde 4 nokta ile ifade edilirken, kesir dereceli PID kontrolör ise taralı alandaki sonsuz nokta ile ifade edilmektedir. Tam dereceli PID kontrolör  $K_p$ ,  $K_i$ ,  $K_d$  katsayılarına sahip olmasına rağmen, kesir dereceli PID ( $PI^\lambda D^\mu$ ) kontrolör ise  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$  gibi 5 tane katsayısı mevcuttur. Katsayıların fazla olması geniş ayarlama alanında kontrol sistemine daha esnek bir müdahale imkânı sunmaktadır. Fakat, kontrol edilen sisteme göre katsayı ayarlanması zorluk oluşturmaktadır.  $PI^\lambda D^\mu$  kontrolör, PID kontrolörde olduğu gibi  $K_d$  veya  $K_p$ ' nin sıfır olma durumlarına göre kesirli  $PI^\lambda$  veya  $PD^\mu$  kontrolörde olabilir.

Kontrol sisteminin çıkış cevabının optimum noktada olabilmesi için (hızlı yükselme zamanı, en az aşım oranı, en düşük sönümlenme zamanı, sıfır kalıcı durum hatası vb...) kontrolör katsayılarının iyi ayarlanması gerekir. Kontrolöre en uygun katsayıların bulunması, kontrolör tasarım çalışmalarının en önemli bölümünü oluşturur. Dolayısıyla bu konu üzerinde oldukça fazla çalışmalar yapılmıştır [22-26, 30]. Klasik PID kontrolörler için en iyi bilinen kontrolör parametrelerini bulma yöntemleri Ziegler-Nichols tarafından verilmiştir [1]. Özellikle, kontrol edilecek sistemin matematiksel modeli bilinmiyorsa Ziegler-Nichols yöntemi kullanarak PID kontrolör katsayıları bulunabilir. Son yıllarda kesir dereceli  $PI^\lambda D^\mu$  kontrolörlerin parametrelerinin bulunması ile ilgili yeni çalışmalar hız kazanarak artmaktadır. Örneğin, endüstriyel uygulamalar için  $PI^\lambda D^\mu$  kontrolörlerin beş parametresinin, beş farklı özelliğe göre oluşturulan beş denklemden elde edilmesi [31]'de çalışılmıştır. "Simetrik Optimum Metot" tekniğini

kullanarak  $PI^\lambda$  kontrolörlerin parametrelerinin ayarlanması [30]' da çalışılmıştır. [26]'da yapılan çalışmada ise klasik yöntemlerle kesir dereceli PID kontrolör parametrelerinin bulunması ve karesel toplam hata (Integral Squared Error-ISE) yöntemi ile  $PI^\lambda D^\mu$  kontrolörün  $\lambda$  ve  $\mu$  değerlerinin elde edilmesi sağlanmıştır. Literatür taramasında görüleceği gibi PID ve  $PI^\lambda D^\mu$  kontrolörler için optimum parametreleri elde etmek amacıyla bunlara benzer birçok yeni çalışma yapılmıştır [1-4, 23-29, 70-73]. Bu tezde  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  katsayılarını ayarlama kurallardan ilki birim basamak S-cevabına dayalı diğeride kritik kazanca dayalı ayarlama kuralıdır [24, 70, 71]. İlk kurallar kümesi olan S-cevabına dayalı ayarlama kurallarında sisteme birim basamak uygulanır ve çıkışta S cevabı alınır. Sistemin transfer fonksiyonu  $G(s) = \frac{K}{1+Ts} e^{-Ls}$  şeklinde kabul edilir ve birim basamak için verdiği S cevabından Şekil 2.4.'deki gibi ise gecikme ( $L$ ), zaman sabiti ( $T$ ) grafik üzerinden bulunduktan sonra  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  parametreleri  $L$  ve  $T$ ' ye göre Çizelge 2.1 ve 2.2' ye göre hesaplanır.



Şekil 2.4. Birim basamak S-cevabı

Örneğin Çizelge 2.1'de kesir dereceli PID kontrolörün oransal katsayısı Denklem 2.25'deki gibi hesaplanabilir.

$$P = -0.0048 + 0.2664L + 0.4982T + 0.0232L^2 - 0.0720T^2 - 0.0348TL \quad (2.25)$$



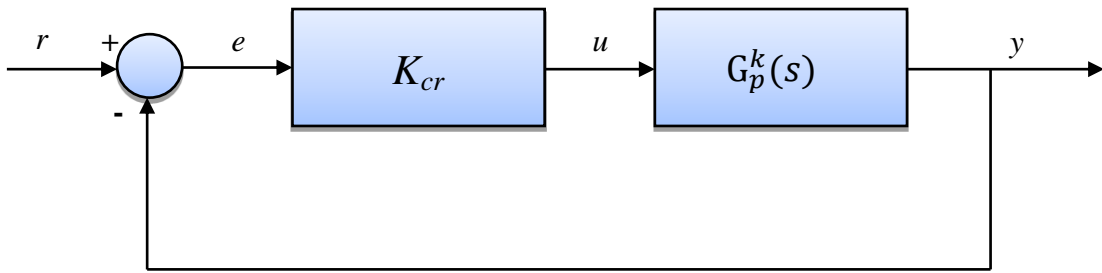
**Çizelge 2.1.** S-cevabına göre birinci parametre ayarlama çizelgesi

$0.1 \leq T \leq 5$ ve $L \leq 2$ için parametreler						$5 \leq T \leq 50$ ve $L \leq 2$ için parametreler				
	P	I	$\lambda$	D	$\mu$	P	I	$\lambda$	D	$\mu$
1	-0.0048	0.3254	1.5766	0.0662	0.8736	2.1187	-0.5201	1.0645	1.1421	1.2902
L	0.2664	0.2478	-0.2098	-0.2528	0.2746	-3.5207	2.6643	-0.3268	-1.3707	-0.5371
T	0.4982	0.1429	-0.1313	0.1081	0.1489	-0.1563	0.3453	-0.0229	0.0357	-0.0381
$L^2$	0.0232	-0.1330	0.0713	0.0702	-0.1557	1.5827	-1.0944	0.2018	0.5552	0.2208
$T^2$	-0.0720	0.0258	0.0016	0.0328	-0.0250	0.0025	0.0002	0.0003	-0.0002	0.0007
LT	-0.0348	-0.0171	0.0114	0.2202	-0.0323	0.1824	-0.1054	0.0028	0.2630	-0.0014

**Çizelge 2.2.** S-cevabına göre ikinci parametre ayarlama çizelgesi

$0.1 \leq T \leq 50$ ve $L \leq 0.5$ için parametreler					
	P	I	$\lambda$	D	$\mu$
1	-1.0574	0.6014	1.1851	0.8793	0.2778
L	24.5420	0.4025	-0.3464	-15.0846	-2.1522
T	0.3544	0.7921	-0.0492	-0.0771	0.0675
$L^2$	-46.7325	-0.4508	1.7317	28.0388	2.4387
$T^2$	-0.0021	0.0018	0.0006	-0.0000	-0.0013
LT	-0.3106	-1.2050	0.0380	1.6711	0.0021

Kontrolör parametrelerinin bulunmasında kullanılan ikinci kurallar kümesinde Şekil 2.5'te olduğu gibi kontrol sistemi osilasyona girene kadar  $K_{cr}$  değiştirilir. Bu noktada osilasyon işaretinin periyodu ( $P_{cr}$ ) belirlenir. Osilasyon periyodu (kritik periyot)  $P_{cr}$  ve osilasyon kazancı (kritik kazanç)  $K_{cr}$ 'ye göre Çizelge 2.3, 2.4 ve 2.5'e göre  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  parametreleri hesaplanır.



**Şekil 2.5.** Birim geri beslemeli sistem

**Çizelge 2.3.** Kritik kazanç ve periyoda göre birinci parametre ayarlama çizelgesi

$K_{cr}P_{cr} \leq 64$ ve $P_{cr} \leq 8$ için parametreler						$64 \leq K_{cr}P_{cr} \leq 640$ ve $P_{cr} \leq 8$ için parametreler				
	P	I	$\lambda$	D	$\mu$	P	I	$\lambda$	D	$\mu$
1	0.4139	0.7067	1.3240	0.2293	0.8804	-1.4405	5.7800	0.4712	1.3190	0.5425
$K_{cr}$	0.0145	0.0101	-0.0081	0.0153	-0.0048	0.0000	0.0238	-0.0003	-0.0024	-0.0023
$P_{cr}$	0.1584	-0.0049	-0.0163	0.0936	0.0061	0.4795	0.2783	-0.0029	2.6251	-0.0281
$1/K_{cr}$	-0.4384	-0.2951	0.1393	-0.5293	0.0749	32.2516	-56.2373	7.0519	-138.9333	5.0073
$1/P_{cr}$	-0.0855	-0.1001	0.0791	-0.0440	0.0810	0.6893	-2.5917	0.1355	0.1941	0.2873

**Çizelge 2.4.** Kritik kazanç ve periyoda göre ikinci parametre ayarlama çizelgesi

$P_{cr} \leq 2$ için parametreler					
	P	I	$\lambda$	D	$\mu$
1	1.0101	10.5528	0.6213	15.7620	1.0101
$K_{cr}$	0.0024	0.2352	-0.0034	-0.1771	0.0024
$P_{cr}$	-0.8606	-17.0426	0.2257	-23.0396	-0.8606
$P_{cr}^2$	0.1991	6.3144	0.1069	8.2724	0.1991
$K_{cr}P_{cr}$	-0.0005	-0.0617	0.0008	0.1987	-0.0005
$1/K_{cr}$	-0.9300	-0.9399	1.1809	-0.8892	-0.9300
$1/P_{cr}$	-0.1609	-1.5547	0.0904	-2.9981	-0.1609
$K_{cr}/P_{cr}$	-0.0009	-0.0687	0.0010	0.0389	-0.0009
$P_{cr}/K_{cr}$	0.5846	3.4357	-0.8139	2.8619	0.5846

**Çizelge 2.5.** Kritik kazanç ve periyoda göre üçüncü parametre ayarlama çizelgesi

$0.2 \leq P_{cr} \leq 5$ ve $1 \leq K_{cr} \leq 200$ için parametreler					
	P	I	$\lambda$	D	$\mu$
1	-1.6403	-92.5612	0.7381	-8.6771	0.6688
$K_{cr}$	0.0046	0.0071	-0.0004	-0.0636	0.0000
$P_{cr}$	-1.6769	-33.0655	-0.1907	-1.0487	0.4765
$K_{cr}P_{cr}$	0.0002	-0.0020	0.0000	0.0529	-0.0002
$1/K_{cr}$	0.8615	-1.0680	-0.0167	-2.1166	0.3695
$1/P_{cr}$	2.9089	133.7959	0.0360	8.4563	-0.4083
$K_{cr}/P_{cr}$	-0.0012	-0.0011	0.0000	0.0113	-0.0001
$P_{cr}/K_{cr}$	-0.7635	-5.6721	0.0792	2.3350	0.0639
$\log_{10}(K_{cr})$	0.4049	-0.9487	0.0164	-0.0002	0.1714
$\log_{10}(P_{cr})$	12.6948	336.1220	0.4636	16.6034	-3.6738

Örneğin Çizelge 2.3’de kesir dereceli PID kontrolörün oransal katsayısı aşağıdaki gibi hesaplanabilir.

$$P = 0.4139 + 0.0145 K_{cr} + 0.1584 P_{cr} - \frac{0.4384}{K_{cr}} - \frac{0.0855}{P_{cr}} \quad (2.26)$$

Kesir dereceli türev ve integral operatörleri ile tanımlanan sistemlerin modellenmesi her ne kadar popüler olsa da, bu alandaki çalışmaların çoğu lineer sistemler üzerinde yoğunlaşmıştır. Aynı şekilde, lineer sistemler için kararlılık konusunda bir çok çalışmalar mevcuttur [6, 7].

## 2.5. Bulanık Mantık Küme Teorisi

Bulanık mantık, insanın düşünme biçimini modellemeye çalışır. Bulanık mantığın klasik yöntemlerden farkı, kesinliklerle çalışmaması ve niteliksel tanımlamalara imkân sağlamasıdır. İnsanlar kesin tanımların ve ölçümlerin yapılamadığı durumlarda, belirsiz ifadeler kullanırlar. Bulanık mantık ile belirsizlikler ifade edilebildiği için karmaşık sistemlerin modellenmesini kolaylaştırırlar. Bulanık mantık, klasik mantıktaki cevap verilemeyen durumları kapsar. Bu da, bulanıklığın ve bulanık mantığın temeli olarak kabul edilir. Bilindiği gibi klasik mantıkta sadece doğru (1) ve yanlış (0) değer alabilen işlemleri ele alır. Bulanık mantık ise klasik mantığı kapsayan ve olayların hangi oranlarda (0 -1 arasında) gerçekleştiğini belirleyen çoklu mantık sistemidir. Bulanık mantık, bu oranları günlük hayatta kullandığımız sözel değişkenlere üyelik dereceleri atayarak gerçekleştirir. Bir nesnenin bir kümeye olan üyelik derecesinin değeri 0 ile 1 arasında değişir. Burada 1 nesnenin kümeye tam üye olduğunu gösterirken, 0 ise nesnenin bulanık kümenin bir üyesi olmadığını gösterir. Klasik kümede bir eleman ya kümenin içindedir ya da dışındadır. Bulanık bir kümede ise bir eleman, tamamen veya kısmen içinde veya dışında olabilir. Üyelik fonksiyonları bir evrensel kümenin alt kümesidir ve elemanın ağırlık derecesini ( $\mu$ ) belirler. Eğer küme elemanı alt kümenin tamamen elemanı ise üyelik fonksiyonu 1 değerini ( $\mu = 1$ ) alırken, tamamen dışında 0 değerini ( $\mu = 0$ ) alır. Eğer üyelik fonksiyonunun değeri 0 ile 1 arasında ise ( $0 < \mu < 1$ ), bu o elemanın kısmen bulanık kümeye ait olduğunu gösterir. Fakat bir elemanın üyelik değeri klasik kümelerde  $\{0,1\}$  gibi iki sayı ile sınırlı iken, bulanık kümelerde  $[0,1]$  aralığında herhangi bir reel sayıyı alabilmektedir.  $U$  evrensel kümesinde bir alt küme olan  $A$  kümesinin  $\mu_A(x)$  üyelik fonksiyonu aşağıdaki gibi tanımlanmaktadır [92, 93].

$$A = \{x, \mu_A(x), x \in U\}$$

$$A \text{ klasik bir küme ise } \mu_A(x) \rightarrow \{0,1\}$$

$$\tilde{A} \text{ bulanık bir küme ise } \mu_A(x) \rightarrow [0,1]$$

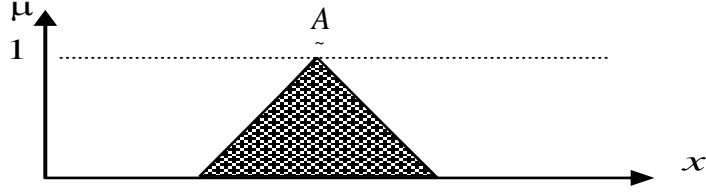
ve bir bulanık  $\tilde{A}$  kümesi aşağıdaki gibi yazılır.

$$\tilde{A} = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots \right\} = \left\{ \sum_i \frac{\mu_A(x_i)}{x_i} \right\} \quad (2.27)$$

$X$  evrensel kümesi sürekli ise, bulanık  $\tilde{A}$  kümesi,

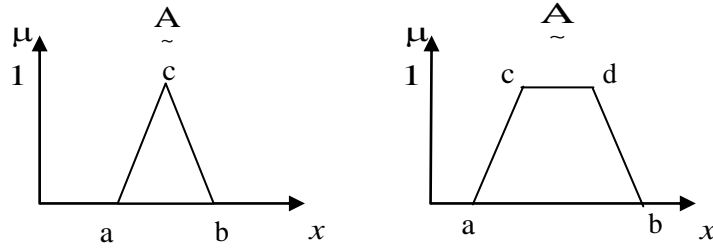
$$\tilde{A} = \left\{ \int \frac{\mu_{\tilde{A}}(x)}{x} \right\} \quad (2.28)$$

gibi gösterilir.



**Şekil 2.6.** A bulanık kümesi için üyelik fonksiyonu

Üyelik derecelerinin 0'dan 1'e ne şekilde değişeceğini üyelik fonksiyonu belirler, Şekil 2.6. Üyelik fonksiyonunun şekli, kümenin ifade etmek istediği uygulama alanına göre değişiklik gösterir. En çok kullanılan üyelik fonksiyonlarına üçgen, yamuk, Gauss, Cauchy, S ve Z şeklindeki Sigmoid fonksiyonları örnek olarak verilebilir. Üçgen ve yamuk fonksiyonları parçalı-doğrusal fonksiyonlardır. Şekil 2.7'de gösterilen bu fonksiyonların grafiksel gösterimleri, oluşturulmaları ve hesaplamaları oldukça kolaydır.



(a) Üçgen

(b) Yamuk

**Şekil 2.7.** Üçgen ve Yamuk Üyelik Fonksiyonları

Denklem 2.29'da verilen üçgen üyelik fonksiyonu için,  $(a, 0)$  başlangıç,  $(c, 0)$  tepe ve  $(b, 0)$  bitiş noktalarıyla tanımlanır.

$$\mu_{\tilde{A}}(x) = \text{Max} \left\{ \min \left[ \frac{x-a}{c-a}, \frac{b-a}{b-c}, 0 \right] \right\} \quad (2.29)$$

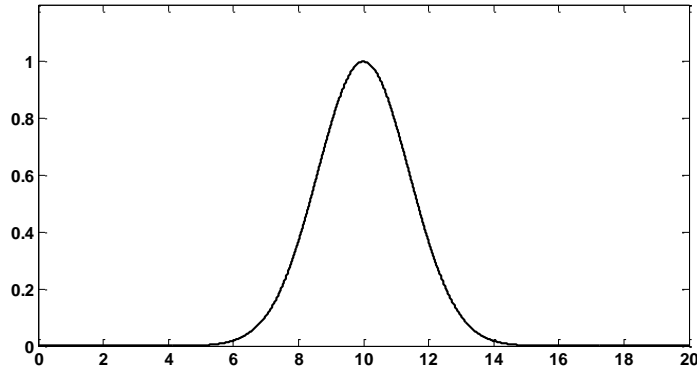
Denklem 2.30'da verilen yamuk üyelik fonksiyonu ise  $(a,0)$  başlangıç,  $(c, a)$  ve  $(d, a)$  tepe ve  $(b, 0)$  bitiş noktalarıyla tanımlanmaktadır.

$$\mu_{\tilde{A}}(x) = \text{Max} \left\{ \min \left[ \frac{x-a}{c-a}, 1, \frac{b-a}{b-d}, 0 \right] \right\} \quad (2.30)$$

Gauss üyelik fonksiyonu Denklem 2.31'deki gibi tanımlanır.

$$\mu_A(x) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (2.31)$$

Burada  $c$  Gauss eğrisinin merkezini,  $\sigma$  ise genişliğini ayarlayan parametrelerdir.  $c = 10$  için Gauss eğrisi Şekil 2.8'de ki gibidir.

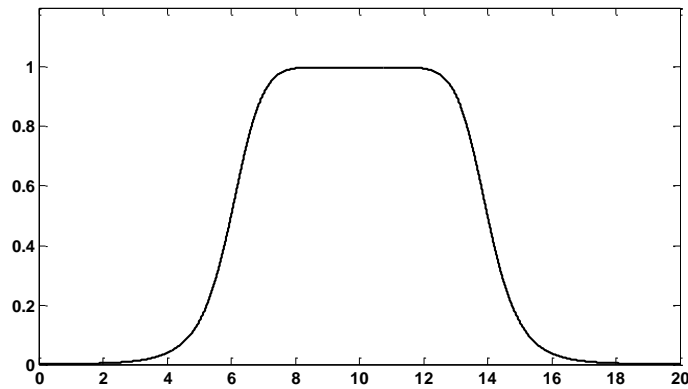


**Şekil 2.8.** Gauss Eğrisi

Şekil 2.9'da gösterilen Cauchy üyelik fonksiyonu ise genelleştirilmiş çan eğrisi olarak da bilinen üyelik fonksiyonudur.

$$\mu_A(x) = \frac{1}{\left(1 + \left|\frac{x-c}{\sigma}\right|^{2n}\right)} \quad (2.32)$$

Denklem 2.32 ile tanımlanan bu fonksiyonda  $c$  eğrinin merkezini,  $\sigma$  taban genişliğini ve  $n$  ise tavan genişliğini belirleyen parametrelerdir.

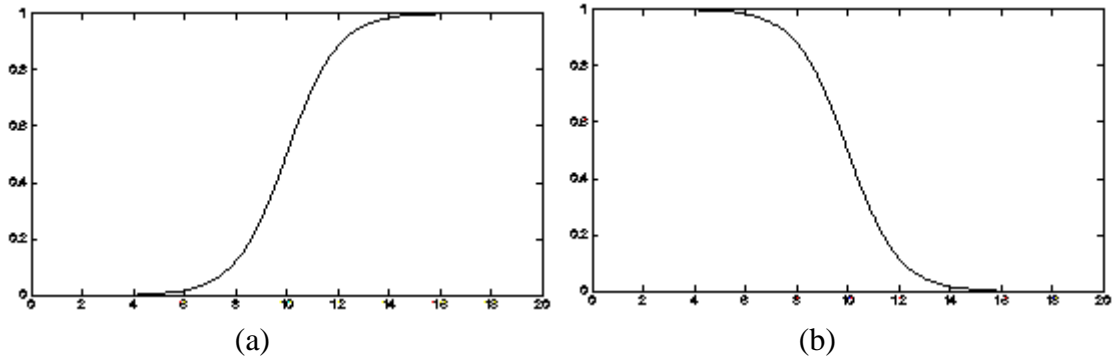


**Şekil 2.9.** Çan eğrisi grafiği

S ve Z şeklindeki sigmoid fonksiyonları Denklem 2.33'deki gibi tanımlanır.

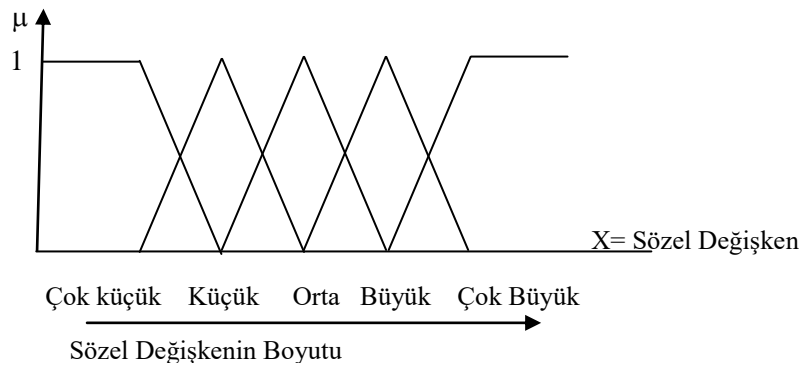
$$\mu_A(x) = \frac{1}{(1 + e^{-(x-m)\sigma})} \quad (2.33)$$

$\sigma$ 'nın işareti; fonksiyonun artan mı (S), Şekil 2.10 (a), yoksa azalan mı (Z), Şekil 2.10 (b), olduğunu; değeri ise artma veya azalmanın şeklini ifade eder. Pozitif değerleri S tipi, negatif değerleri ise Z tipi eğri oluşturur.  $m$  fonksiyonun merkezi yani eğimli kısmın orta noktasıdır.



**Şekil 2.10.** (a) S tipi sigmoid fonksiyonunun grafiği (b) Z tipi sigmoid fonksiyonunun grafiği

Birden fazla üyelik fonksiyonunun evrensel küme üzerindeki gösterimi Şekil 2.11'deki gibidir. Pek çok kavram ve büyüklük, sözel olarak derecelendirilebilir veya sınıflandırılabilir.

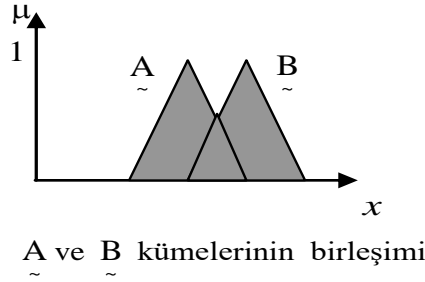


**Şekil 2.11.** Evrensel küme üzerinde birden fazla üyelik fonksiyonunun gösterimi

$X$  evrensel küme üzerinde,  $A$  ve  $B$  şeklinde iki bulanık küme tanımlayalım. Bulanık kümelerde işlemler, klasik kümelerde olduğu gibidir. Bulanık kümelerde birleşim (Şekil

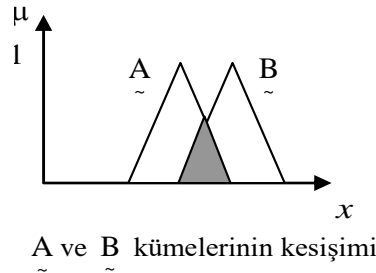
2.12), kesişim (Şekil 2.13) ve tümlleme (bulanık kümenin tersi) (Şekil 2.14) işlemlerine ilişkin temel tanımlar Denklem 2.34, 2.35 ve 2.36'da verilmiştir:

$$\text{Birleşim: } \mu_{\underline{A} \cup \underline{B}}(x) = \max(\mu_{\underline{A}}(x), \mu_{\underline{B}}(x)) \quad (2.34)$$



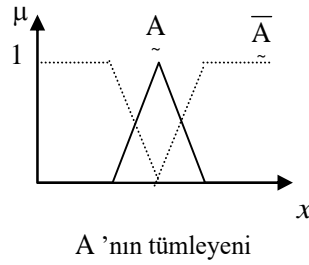
**Şekil 2.12.** A ve B bulanık kümelerinin birleşimi

$$\text{Kesişim: } \mu_{\underline{A} \cap \underline{B}}(x) = \min(\mu_{\underline{A}}(x), \mu_{\underline{B}}(x)) \quad (2.35)$$



**Şekil 2.13.** A ve B bulanık kümelerinin kesişimi

$$\text{Tümlleme: } \mu_{\underline{X} - \underline{A}}(x) = 1 - \mu_{\underline{A}}(x) \quad (2.36)$$



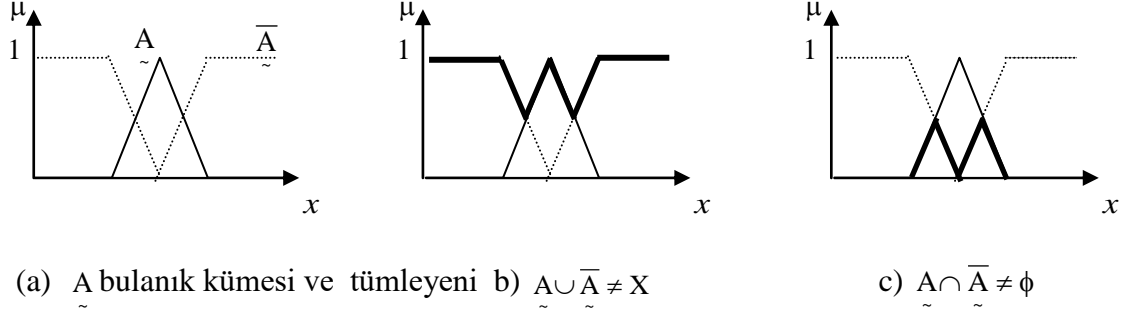
**Şekil 2.14.** A bulanık kümesinin tümlleyeni

Tümlleme işlemi hariç temel küme özellikleri, bulanık kümelerle klasik kümelerde aynıdır. Bulanık küme ile tümlleyeni arasındaki işlemler Denklem 2.37 ve 2.38'deki gibidir.

$$\tilde{A} \cup \tilde{A} \neq X \quad (2.37)$$

$$\tilde{A} \cap \tilde{A} \neq \phi \quad (2.38)$$

Bu işlemlerin grafiksel gösterimi ise Şekil 2.15’de gösterilmiştir:



**Şekil 2.15.**  $\tilde{A}$  bulanık kümesinin tümleyeni ve tümleme işlemleri

Bulanık yapılar, seçilen giriş değerlerinden çıkış değerlerinin elde edilmesini sağlamak için bulanık küme teorisini kullanan sistemlerdir. Bu sistemlerin en büyük avantajı, sözel verilerin ve insan deneyimlerinin bulanık modele katılması ile çözüme ulaşılmasıdır. Bulanık çıkarım sistemi, bulanık **Eğer- İse (If – Then)** kuralları adı verilen bulanık kurallara dayanan bir sistemdir. Bu model **Eğer** (öncül kısım) ve **İse** (soncul kısım)’ lardan oluşur. Bulanık kuralların genel formu aşağıdaki gibidir;

*Kural 1:* If  $x = A_1$  and  $y = B_1$  then  $z = N_1$

*Kural 2:* If  $x = A_2$  and  $y = B_2$  then  $z = N_2$

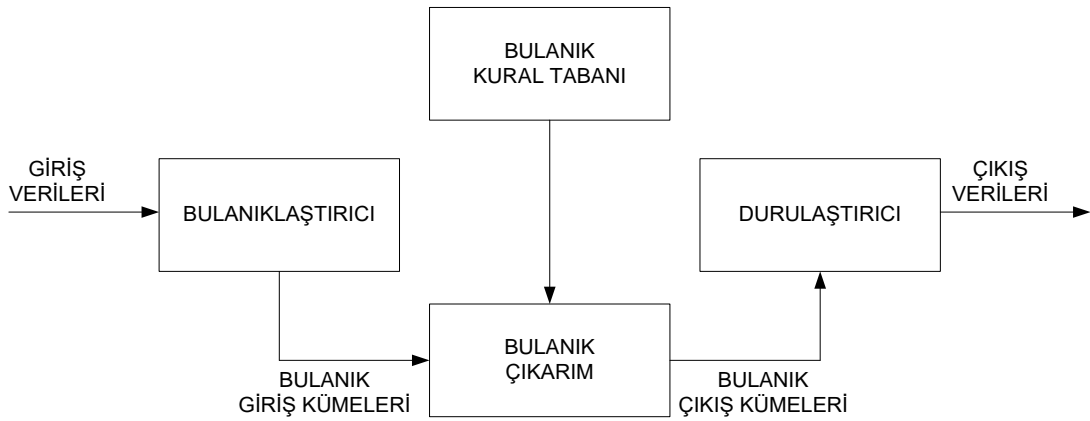
Bulanık mantık kontrolörler (Fuzzy Logic Controller – FLC), pek çok önemli uygulamanın tasarlanmasında ve geliştirilmesinde önemli rol alırlar. Üyelik fonksiyonlarının ve kural tabanının seçimi, istenilen kontrolör performansın alınması için önemlidir. Şekil 2.16’da FLC’nin iç blok diyagramı gösterilmiştir. Kural tabanı, veri tabanı ve bulanık çıkarım mekanizması FLC yapısını oluşturan temel kısımlardır. Kural tabanı bulanık kuralların seçimini içerir, veri tabanı bulanık kurallarda kullanılan üyelik fonksiyonlarının bilgilerini içerir ve çıkarım birimi kurallar doğrultusunda verilen giriş ve koşullara bağlı olarak çıkışları belirler. FLC her bir birimi aşağıdaki işlemleri yerine getirir [101-103]:

#### I. Bulanıklaştırma (Fuzzification)

- 1) Bulanık giriş ve çıkış vektörleri için normalizasyon;



- 2) Bulanık giriş ve çıkış vektörleri için üyelik fonksiyonlarının sayısı ve şeklinin seçimi;
  - 3) Bulanık girdilerin her net değeri için üyelik derecelerinin hesaplamak;
- II. Bulanık çıkarım (Fuzzy inference)
- 4) Pratik sistem operasyonu açısından kural tabanının tamamlanması;
  - 5) Kural tabanında saklanan aktif kuralları belirlemek;
  - 6) Seçilen bulanıklaştırma yöntemine göre her kuralın katkıda bulunduğu üyelik derecesi ve çıkarım sonucunun hesaplanması;
- III. Durulaştırma veya Netleştirme (Defuzzification)
- 7) Uygun netleştirme yöntemi ile kesin bulanık çıkış değerinin bulunması.



**Şekil 2.16.** FLC blok diyagramı

Mamdani ve Sugeno bulanık çıkarım sistemleri, literatürde en çok bahsi geçen sistemlerdir . Bu iki popüler bulanık çıkarım sistemleri sırasıyla:

1. Mamdani tipi bulanık model: Bu yöntem, uzman bilgisi gerektiren ve her türlü problemin çözümüne uygulanabilen ve yaygın olarak kullanım alanı olan bir bulanık mantık çıkarım yöntemidir [104]. Bu model 5 adımda oluşur.
  - I) Girdilerin bulanıklaştırılması: Eğer (öncül) kısmındaki bulanık ifadeleri kullanarak girdi değişkenlerine 0 ile 1 arasında değişen üyelik derecelerini belirler.
  - II) Bulanık küme mantıksal işlemcilerin (and, or) uygulanması
  - III) Bulanık mantık işlemlerini kullanarak kural ağırlıklarının belirlenmesi
  - IV) Sonuçların toplanması: Her bir kuralın çıktısını temsil eden bulanık kümelerin birleştirilmesi
  - V) Netleştirme: Toplam bulanık küme sonuçlarının tek bir sayıya dönüştürülmesi

2. Sugeno tipi bulanık model [105]: Mamdani bulanık mantık yönteminin bir uyarlamasıdır. Bu nedenle, Mamdani tipi bulanık modeldeki işlemler bu modelde de geçerlidir. İki yöntem arasındaki temel fark, çıkıştaki üyelik fonksiyonlarıdır. Sugeno tipi bulanık modellemede, çıkış üyelik fonksiyonları sadece lineer veya sabittir. Mamdani bulanık modelde ise çıkış üyelik fonksiyonları üçgen veya tek değer şeklinde verilir. Sugeno tipi bulanık modelde, çıkış üyelik fonksiyonları 0. ve 1. derece doğru denklemi şeklinde olduğu durumlarda 1. derece Sugeno bulanık model olarak adlandırılırlar. Böylece Sugeno tipi bulanık model, Mamdani tipi bulanık modelden daha karmaşık ve gösterim açısından daha elverişlidir. 1. derece Sugeno bulanık model aşağıdaki gibi tanımlanır [102].

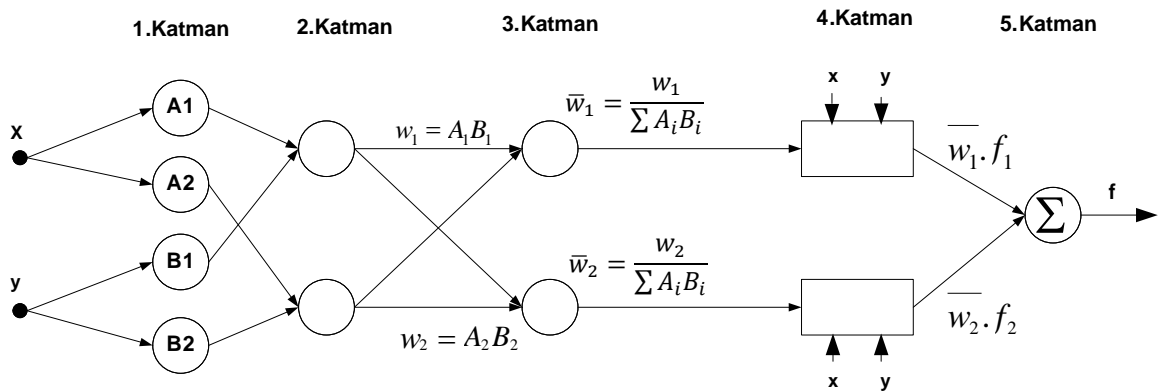
$$\text{If } x = A \text{ and } y = B, \text{ then } z = f(x,y) = px+qy+r$$

İki kural içeren 1.derece bir Sugeno bulanık modelin kuralları;

$$\text{Kural 1: If } x = A_1 \text{ and } y = B_1 \text{ then } z = f_1 = p_1x + q_1y + r_1$$

$$\text{Kural 2: If } x = A_2 \text{ and } y = B_2 \text{ then } z = f_2 = p_2x + q_2y + r_2$$

şeklinde yazılır. Burada  $A_1, A_2$  ve  $B_1, B_2$  üyelik fonksiyonlarıdır.  $p, q$  ve  $r$  ise Sugeno tipi çıkış fonksiyonlarının sabit parametreleridir. Sugeno tipi bulanık model kullanan ANFIS yapısı Şekil 2.17’de gösterilmiştir.



Şekil 2.17. İki girişli tek çıkışlı ANFIS yapısı

Şekil 2.17’de ANFIS modelinde kurallar aşağıdaki gibi tanımlanır:

*Kural 1:* If  $x = A_1$  and  $y = B_1$  then  $z = C_1$

*Kural 2:* If  $x = A_2$  and  $y = B_2$  then  $z = C_2$

Şekil 2.17’deki gibi iki girişli tek çıkışlı bir ANFIS mimarisinde 5 katman bulunur.  $A_1, A_2$  ve  $B_1, B_2$  üyelik fonksiyonları gösterirken,  $p, q$  ve  $r$  eğitim işlemi boyunca belirlenecek olan tasarım parametrelerini göstermektedir.

**1. Katman:** Her bir düğümün çıkışı, giriş değerlerine ve her bir düğümde kullanılan üyelik fonksiyonundan hesaplanan üyelik derecelerinden oluşmaktadır ve düğümlerin çıkışı aşağıda verilen ifade ile hesaplanır [106].

$$O_i^1 = \mu_{A_i}(x) \quad i=1, 2 \quad (2.39.a)$$

$$O_i^1 = \mu_{B_i}(y) \quad i=1, 2 \quad (2.39.b)$$

Burada  $\mu_{A_i}(x)$  ve  $\mu_{B_i}(y)$  herhangi bir bulanık üyelik fonksiyonu ile bulunabilir. Bu tez çalışmasında  $A_i, B_i$  için Denklem 2.40’daki gibi çan üyelik fonksiyonu kullanılmıştır.

$$\mu(x) = \frac{1}{\left[1 + \left|\frac{x - c_i}{a_i}\right|^{2b_i}\right]} \quad i=1, 2 \quad (2.40)$$

Burada  $\{a_i, b_i, c_i\}$  çan üyelik fonksiyonunun şeklini ( $c$ : eğrinin merkezini,  $a$ : taban genişliğini,  $b$ : tavan genişliğinin) değiştiren parametrelerdir.

**2. Katman:** Bu katmandaki her bir düğümün çıkışı, kuralların çarpımı ile elde edilen ağırlığı,

$$O_i^2 = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y), \quad i=1, 2 \quad (2.41)$$

şeklinde hesaplanır.

3. Katman: Bu katmandaki  $i$ . çıkış, bu düğüme gelen ağırlığın tüm ağırlıkların toplamına oranı ile Denklem 2.42'deki gibi belirlenir.

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (2.42)$$

Burada  $\bar{w}_i$  normalize edilmiş ağırlık olarak adlandırılır.

4. Katman: Bu katmandaki her bir  $i$  düğüm çıkışı aşağıdaki fonksiyona sahiptir:

$$O_i^4 = \bar{w}_i \cdot f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (2.43)$$

Burada  $\bar{w}_i$  üçüncü katmanın çıkışıdır ve  $\{p_i, q_i, r_i\}$  parametre kümesidir.

5. Katman: Bu katmanda çıkışı, gelen bütün işaretlerin toplamının toplam ağırlıklara oranı olarak Denklem (2.44)'deki gibi hesaplanır.

$$O^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.44)$$

## 2.6. Yapay Sinir Ağları – YSA

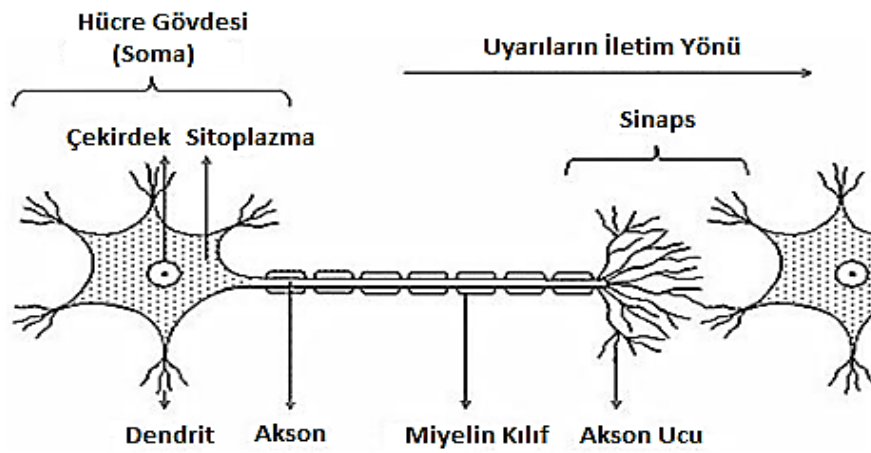
İnsan beynindeki sinir hücrelerine benzeyecek şekilde yapay sinir hücrelerinin değişik bağlantı geometrileri ile birbirlerine bağlanması sonucu oluşan sistemlerdir. YSA paralel hesaplama tekniğini kullanırlar [107]. Bir problemi örnekler üzerinden eğitilerek öğrenirler. YSA bilgileri sinir hücreleri arasındaki bağlantılar üzerindeki ağırlık değerleri ile saklarlar.

İnsan beynindeki çalışma sisteminin yapay olarak oluşturulma çabalarının bir sonucu olarak YSA ortaya çıkmıştır. Bir sinir ağı insan beynindeki sinir hücrelerinin, ya da yapay olarak işlemcilerin birbirlerine değişik etki seviyeleri ile bağlanması sonucu oluşan karmaşık bir sistem olarak düşünülebilir. SA bugün elektrik ve bilgisayar mühendisliği için önemli araştırma konularından biri haline gelmiştir. Mühendislik uygulamalarında geniş çaplı kullanımın nedeni, klasik tekniklerle çözümü zor olan problemlerin üstesinden gelmesidir [112-115]. Bilgisayarlar çarpma, bölme gibi

matematiksel ve algoritmik hesaplama işlemlerinde hız ve doğruluk açısından insan beyninden yüzlerce kat daha fazla başarı gösterebilirken, insan beyninin öğrenme ve tanıma gibi işlevlerini yeteri kadar gerçekleştiremiyorlar. Bu nedenle YSA teorisi geliştirilmiş ve son 50 yıl içerisinde tercih edilebilir bir çalışma alanı olmuştur [116].

YSA insan beyninin çalışma yapısı örnek alınarak geliştirilmeye çalışılmıştır ve aralarında yapısal olarak bazı benzerlikler olduğu için öncelikle biyolojik sinir yapısına bakmak gerekir [110].

### 2.6.1. Biyolojik Sinir Yapısı

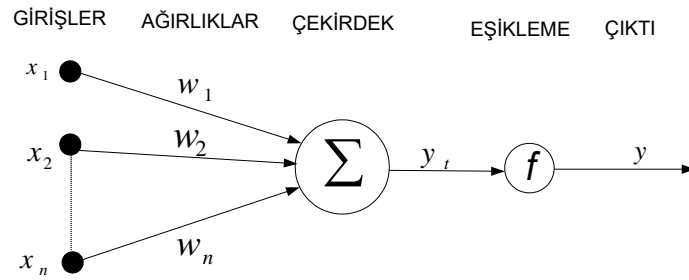


Şekil 2.18. Biyolojik sinir hücre yapısı

İnsanın bilgi işleme olayı beyninde gerçekleşir. En karmaşık sinir ağı Cerebral Cortex denilen “beyin”dir. İnsan beynindeki sinir sisteminin ise en basit yapısı sinir hücreleridir. Beyinde yaklaşık olarak  $10^{10}$  adet sinir hücresi vardır. Hücre başına bağlantı sayısı ise yaklaşık olarak  $10^4$  seviyelerindedir. Beynin çalışma frekansı 100 Hz’dir. Fiziksel boyutları ise 1.3 kg ve  $0.15 \text{ m}^2$  kesitlidir. Beyindeki sinir hücreleri vücudun değişik yerleri ile bilgi alış veriş yaparlar.

Şekil 2.18’de gösterildiği gibi sinir hücresi, soma adı verilen hücre gövdesi, dendrit denilen kıvrımlı uzantılar ve aksondan oluşur. Dendrit’lerin işlevi hücreye gelen girişleri toplamaktır. Dendrit tarafından alınan işaretler hücrede birleştirildikten sonra bir çıkış darbesi üretilip üretilmeyeceğine karar verilir. Eğer bir iş yapılacaksa üretilen çıkış darbesi aksonlar tarafından taşınarak diğer sinir hücrelerine veya terminal organlara iletilir. Beyindeki korteksde her sinir hücresinin bir karşılığı vardır. Bir sinir hücresinin çıkışı ona bağlı olan bütün sinir hücrelerine iletilir. Fakat korteks, işin yapılabilmesi için hangi sinir hücresi harekete geçirilecekse, sadece ona komut gönderir.

Somanın içinde ve çevresinde sodyum, kalsiyum, potasyum ve klor iyonları vardır. Potasyum yoğunluğu sinir hücresinin içinde, sodyum yoğunluğu ise dışındadır. Somanın zarı elektriksel olarak uyarılınca zar, sodyum ve kalsiyum gibi diğer iyonların içeri geçmesine izin verir ve somanın iç durumunu değiştirir. Sinir hücreleri arasındaki bağlantılar hücre gövdesinde veya "sinaps" adı verilen dendritlerdeki geçişlerde olur. Sinir sistemi milyarlarca sinir hücresi ile tek bir sinir hücresinden çıkan aksonun 10000 kadar diğer sinir hücresini bağlayan bir ağdır. Sinapslarla düzeltilen işaretleri taşıyan aksonlar ve dendritlerle içiçe geçmiş sinir hücreleri bir sinir ağı oluştururlar [117].

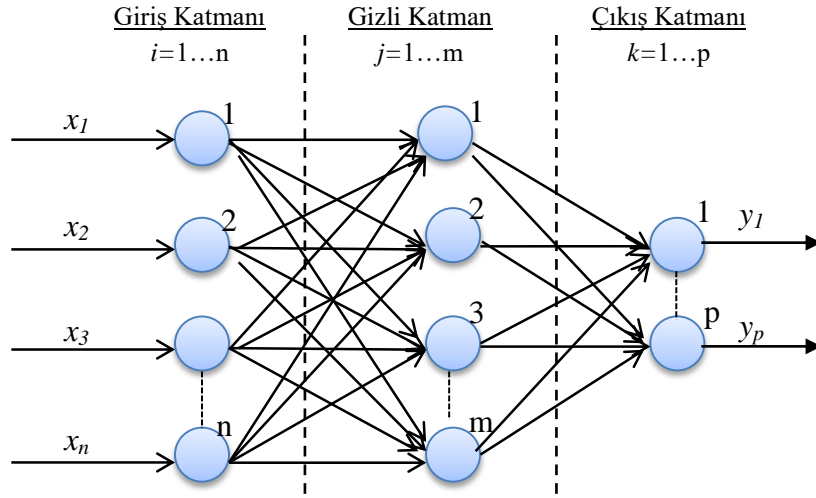


**Şekil 2.19.** Yapay sinir hücre yapısı [ $x_i$  ( $i=1, 2, \dots, n$ ),  $w_i$  ( $i=1, 2, \dots, n$ )]

Şekil 2.19'daki yapay sinir hücresinin girişleri  $x_i$  ile gösterilmiştir ve herbir girişin ağırlık katsayısı (bu katsayı giriş ile hücre arasındaki bağlantının kuvvetini gösterir)  $w_i$  ile gösterilmiştir. Giriş sinyalleri ağırlıklarla çarpıldıktan sonra çekirdekte toplanarak tüm girdi sinyallerinin ağırlıklı toplamları elde edilir.  $y_i$  çekirdekte elde edilen toplam sinyali göstermektedir ve bu toplam sinyal aktivasyon ( $f$ ) fonksiyonuna giriş olarak yönlendirilir. Sinapsis üzerindeki aktivasyon fonksiyonundan çıkan sonuç sinyali  $y$  ile belirtilmiş ve diğer bir sinir hücresine gönderilir.

Yapay sinir hücresinin görevi;  $x_i$  girdilerine karşılık  $y$  çıktı sinyalini oluşturmak ve bu sinyali diğer sinir hücrelerine iletmektir. Her  $x_i$  ile  $y$  arasındaki ilişkiyi temsil eden  $w_i$  ağırlıkları, her yeni giriş ve çıkış sinyaline göre tekrar güncellenir. Bu ayarlama sürecine yapay sinir hücresinin öğrenme süreci denir. Öğrenmenin tamamlanabilmesi için  $w_i$  ağırlıklarındaki değişim kararlı duruma gelene kadar yeni giriş-çıkışlarla sistem beslenir. Kararlılık sağlandığı zaman hücre öğrenmesini tamamlamıştır.

Bir çok yapay sinir hücrelerinin birleşiminden oluşan çok katmanlı yapının tümü yapay sinir ağı olarak nitelendirilir. Böylece Şekil 2.20'de gösterildiği gibi çok katmanlı yapay sinir ağı modeli kurulmuş olur.



Şekil 2.20. Çok katmanlı yapay sinir ağı

## 2.6.2. Yapay Sinir Ağlarında Öğrenme

Sinir hücresine giren sinyaller katsayılarla çarpılarak toplamları alınır. Eğer toplam sinyal eşik değerinden büyük ise o hücre çıkışı aktif ( $y = 1$ ) değilse hücre pasif ( $y = 0$ ) olur. Bu çalışma biçimi yapay sinir hücresinin sınıflandırma yapabilmesini sağlar. Girdilere 1 ya da 0 cevabını vererek sınıflandırma yapabilen bir hücre, hangi girdilere 1 hangilerine 0 diyeceği hakkında karar vermiş olur. “Karar vermek” ve “sınıflandırmak”, öğrenme sürecinin temel yapı taşlarını oluşturmaktadır [107, 117].

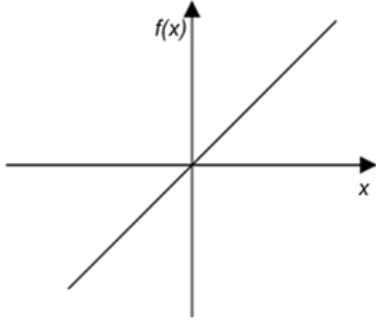
Denklem 2.45’de ifade edildiği gibi YSA’nın girişleri  $w_i$  ağırlıkları ile çarpılıp toplanarak hücre çıkışı hesaplanır [117].

$$y_t = \sum_{i=1}^p x_i * w_i \quad (2.45)$$

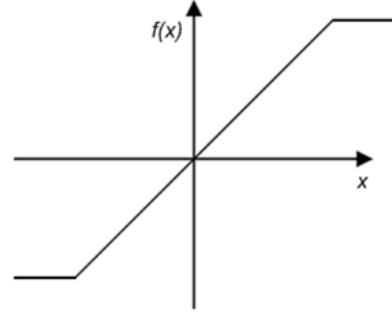
$y_t$  şeklinde toplanan değer çekirdek tarafından akson üzerinden sinapsise gönderilir. Sinapsis gelen bu toplam sinyal değerini aktivasyon fonksiyonundan geçirdikten sonra çıkış sinyalini verir.

$$y = f(y_t) \quad (2.46)$$

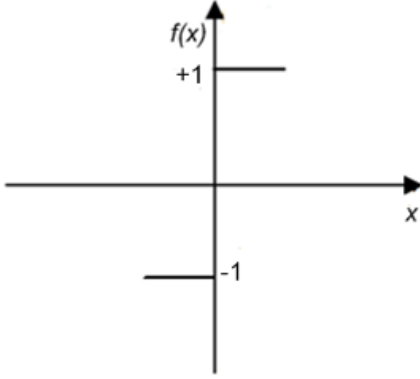
Denklem (2.46)’da  $f$  fonksiyonu matematiksel herhangi bir fonksiyon olabilir. Fakat yapay sinir ağı modellerinde 4 tip fonksiyonun daha çok kullanıldığı görülür. Bu fonksiyon şekilleri Şekil 2.21’de gösterilmiştir [107, 109, 117]:



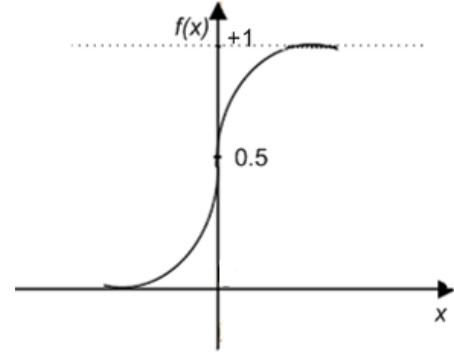
(a)  $f(x) = \alpha \cdot x$



(b)  $f(x) = \begin{cases} +\tau & : x \geq \lambda \\ x & : |x| < \lambda \\ -\tau & : x < -\lambda \end{cases}$



(c)  $f(x) = \begin{cases} 1: & x > 0 \\ 0: & x = 0 \\ -1: & x < 0 \end{cases}$



(d)  $f(x) = \frac{1}{1 + e^{-x}}$

**Şekil 2.21.** (a) doğrusal (b) rampa (c) basamak (d) sigmoid aktivasyon fonksiyonları

Yapay sinir hücrelerinde öğrenme ağırlıklar ( $w$ ) üzerinde gerçekleşir. Yapay sinir hücrelerinde öğrenme süreci iki şekilde gerçekleşir;

- Danışmanlı öğrenme (supervised learning)
- Danışmansız öğrenme (unsupervised learning)

Danışmanlı öğrenmede yapay sinir ağı bir öğretici tarafından kendisine dışarıdan verilen girişler yardımı ile eğitilir. Bu öğretimde, girişlere ( $x_i(t)$ ) karşılık sinir ağının nasıl bir sonuç ( $d(t)$ ) vermesi gerektiği önceden bilinmektedir ve yapay sinir ağının ağırlıklarını buna göre güncellenmektedir. Burada bilinen çıkış değeri  $d(t)$  ile yapay sinir ağının verdiği  $y$  sonucu karşılaştırılarak aradaki hata ile ağırlıklar  $w_i$  güncellenir. [117].



$$w_{i,yeni}(t+1) = w_{i,eski}(t) + (\mu * [d(t) - f(y_t)] * x_i(t)) \quad (2.47)$$

Danışmansız öğrenme ise yapay sinir ağının dışarıdan herhangi bir yardım almadan kendisine verilen bilgileri kendi içerisinde kıyaslama yaparak sınıflandırması ile oluşan öğrenme biçimidir. Yapay sinir ağı bunu yapabilmek için ilk aldığı örneği (ya da örnekleri) bir sınıf olarak belirler. Daha sonra gelecek olan tüm girdi değerlerini o sınıfa benzetmeye çalışır. Bu şekilde, tüm girdi değerlerini kendi aralarında benzeyip benzememelerine göre ayırt eder. Bu öğrenmede önemli noktalardan bir tanesi girdiler sisteme ne kadar çok sayıda girilirse, sinir ağının öğrenme oranı o oranda artar ve vereceği cevaplardaki hata payı düşer.

Yapay sinir ağları çoğunlukla bu iki öğrenme metodundan ya birini ya da hibrid denilen karma modeli kullanırlar. Öğrenme sürecinde sadece bu öğrenme metotları ile ağırlıkların ( $w$ ) güncellenmesinden ibaret değildir. Bu süreç üzerinde başka parametrelerinde etkileri vardır. Örneğin güncellemede kullanılan Denklem (2.47)'da yer alan ve öğrenme katsayısı olarak ifade edilen  $\mu$  parametresi de öğrenme sürecini etkiler.  $\mu$  katsayısı öğrenmenin süresinin ve doğruluğunun ilişkisini ayarlayan bir parametredir. Bu parametrenin değerinin seçimi son derece önemlidir. Bu değer büyük almak öğrenmenin kaba ve kısa süreli olmasına sebep olurken, küçük alınması ise öğrenmenin hassas ve uzun süreli olmasına neden olur. Dolayısıyla  $\mu$  değerinin ne çok büyük ne de çok küçük alınmaması gerekir. Çünkü büyük alınan değerle istenilen sonuca ulaşılamazken, küçük alınan değerle de istenen sonuca çok düşük bir performans ile çok uzun bir sürede ulaşılabacaktır. Bu sıkıntıyı ortadan kaldırmak için yapay sinir ağlarındaki öğrenme süreçlerinde belli döngü sayısı geçildikten sonra  $\mu$  parametresinin değeri azaltılır. Böylece, öğrenme sürecinin başında algoritma büyük adımlarla sonuca doğru ilerlerken, belli bir döngü sayısından sonra adımlarını küçülterek ilerler. Bu da algoritmanın hızlı ve hassas biçimde sonuca gitmesini sağlar. Bunun için  $\mu$  değeri sabit bir katsayı ile çarpılır [117, 118].

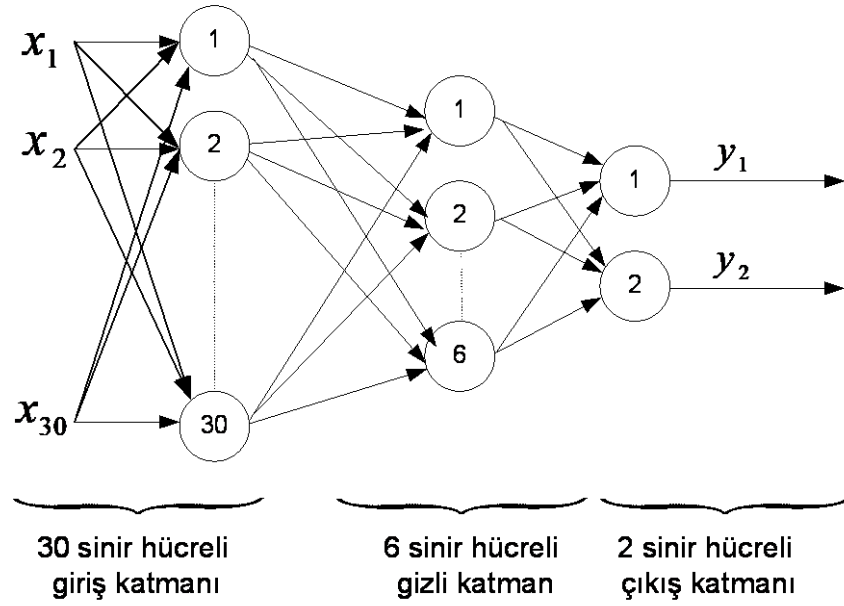
$$\mu = \mu * sabit \quad (2.48)$$

Öğrenmeyi etkileyen başka bir değişkende verilerin büyüklüğü ve döngü sayısıdır. Yapay sinir ağının doğru çözümü yakalayabilmesi ancak yeterli girdiye sahip olması ile

olur. Örneğin bir yörenin yağmur yağı ile ilgili tahminde bulunacak olan bir sinir ağının 30 günlük verileri kullanarak bu tahminleri doğru yapması beklenemez. Bir sinir ağının ne kadarlık veri ile doğru sonuçları yakalayabileceği ile ilgili ön çalışma yapılması gerekir. Problemin tipine göre veri miktarı ve o verilerin işleneceği döngü sayısının değişik olacağına dikkate alınmalıdır. Bir yapay sinir ağının bir problemi daha iyi öğrenebilmesi, daha hassas ve doğru cevaplar verebilmesi için problemin kendi içinde de döngü sayısı değiştirilebilir. Örneğin, hava tahmini yapacak olan sinir ağına 10 senelik hava durumu bilgileri bir kere verildiğinde sinir ağının vereceği cevap gerçeğe çok uzak olacaktır, YSA'ya 1000 kere aynı veriyi girdiğimizde alacağımız cevap daha hassas olacaktır. Sonuç olarak döngü sayısı hem problemden probleme ve hemde problemin kendi içindeki hassasiyete göre değişebilir. Döngü sayısında ne çok olmalı ne de az olmalı, çünkü çok fazla döngü performansı azaltıp sonuca ulaşma süresini uzatırken, az sayıda döngü de kaba sonuçlar vereceğinden sonuca ulaşmayı engelleyecektir. Yapay sinir ağlarında, her problem için döngü sayısı deneme yanılma yöntemiyle belirlenmeye çalışılır [117].

Sinir ağının katman sayısında öğrenmeyi etkileyen önemli faktörlerden biridir. Şekil 2.20'deki gibi 3 katmandan oluşan bir yapay sinir ağı modelinin karmaşık problemler için bile yeterli olmaktadır. Ancak problemden probleme ya da modelden modele katman sayısı değişiklik gösterebilir. Bu 3 katmanlı modeldeki katmanlar sırasıyla; girdi katmanı, gizil katman ve çıktı katmanlarıdır.

Öğrenmeyi etkileyen bir başka faktör de katmanlarda kullanılan sinir hücrelerinin sayısıdır. Şekil 2.22'de 1. katmanda (girdi) 30 tane sinir hücresi, gizli katmanda 6 tane sinir hücresi ve çıktı katmanında ise 2 tane sinir hücresi yer almaktadır. Buradaki sayılar kullanıcının giriş ve çıkış sayılarını dikkate alarak belirlemiş olduğu sayılardır. Her katmanda ne kadar sinir hücresi olacağı önceden bilinemez, döngü sayısı tesbitinden olduğu gibi deneme yanılma yöntemi ile belirlenir. Katmanlarda çok sayıda sinir hücresi kullanmak yapıyı karmaşıktırıp ve performansı düşürürken az sayıda hücre kullanmak da sinir ağının problemi öğrenememesine neden olur [116-118].



Şekil 2.22. 3 katmanlı YSA

Yapay sinir ağlarının iki işlevi vardır. Bunlardan biri kendisine verilen girişler kullanarak öğrenme işlevi, diğeri de öğrendikten sonra yeni girişler ile bir sorgulama yapıldığında buna doğruya yakın cevap vermesidir. Cevabı verirken sinir ağı karşılaştığı yeni girişe önceden bildiklerinden birine benzeterek cevap verir. Bu çalışma şekillerinden dolayıdır ki yapay sinir ağları hata toleranslı yapılar olarak değerlendirilir. Bir örnek vermek gerekirse, yüz tanıma için tasarlanan bir sinir ağı kişilerin yüzlerindeki bazı yapıları öğrenerek yüzleri tanımaya çalışırlar. Sistem kişinin yüzünü bir kere öğrendikten sonra, yüzünün tamamı verilmese bile doğru sonuç verebilir. Bu da göstermektedir ki girdinin tamamı verilmese bile yapay sinir ağı doğru cevaplar verebilir [117].

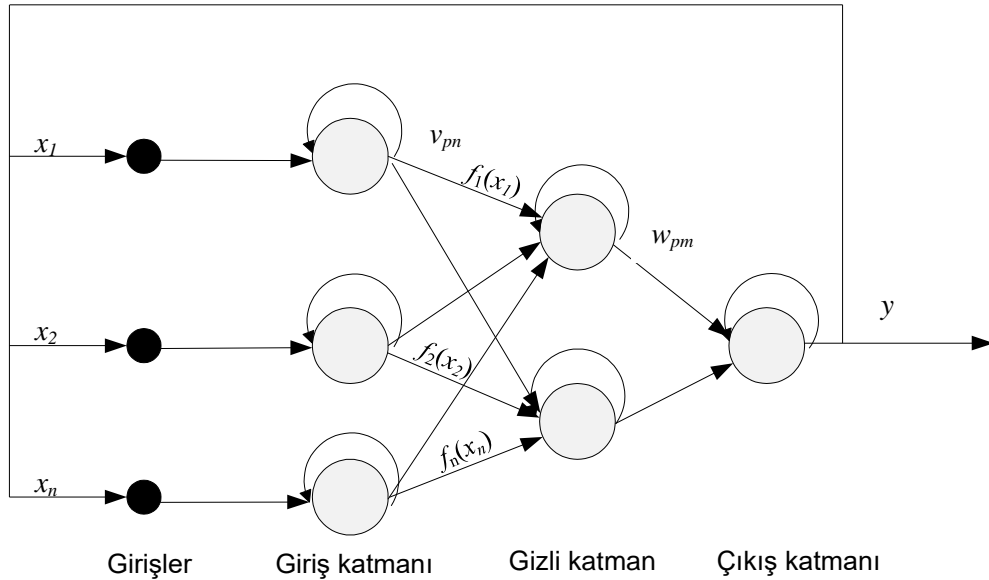
### 2.6.3. Geri Yayılım Ağ Modeli

Bu çalışmada geri yayılım ağ modeli YSA' nın eğitilmesi için kullanılmıştır. Yapay sinir ağları içerisinde her probleme rahatlıkla uyarlanabilen model geri yayılım ağ modelidir. Şekil 2.23'de gösterildiği gibi geri yayılım ağ modelinde geri beslemeli bir öğrenme mekanizması kullanılır. Burada yapay sinir ağının yapısı aslında ileri beslemeli olmasına rağmen hatanın geriye doğru yani giriş birimlerini veya ara katmanları geri beslemesinden kaynaklanan bir öğrenme olduğu için geri besleme söz konusudur. Bu

yapay sinir ağı modelinde öğretmenli öğrenme stratejisi kullanılır [111, 112, 117]. Geri besleme, bir katmandaki hücreler arasında olabileceği gibi katmanlar arasındaki hücreler arasında da olabilir. Geri yayılım ağları bu yapılarından dolayı doğrusal olmayan dinamik bir davranış gösterirler. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli sinir ağı yapıları elde edilebilir.

Geri yayılım sinir ağlarındaki öğrenmede sürekli giriş sinyali kullanılır. Aktivasyon fonksiyonu olarak daha çok türevi alınabilen sigmoid fonksiyonunu kullanır. Öğrenme işlemi için kullandığı yöntem ise delta kuralıdır. Eğitimin tam olarak gerçekleşebilmesi için sistemde alınmış her bir giriş kümesine karşılık istenen çıkış kümesinin mevcut olması gerekir. Bu şekildeki giriş ve hedef çıkışı kümeleri bir eğitim çiftini oluştururlar. Eğitimi tamamlamış olan bir sinir ağında, önceden sistemden alınmış çıkış ile ağdan elde edilen çıkışlar eşit olmalıdır. Önceden sistemden alınmış çıkış ile ağdan elde edilen çıkışlar arasındaki fark hatadır. Bu hatayı minimuma indirmek için, ağ ağırlık parametrelerindeki güncellemeyi geriye yayılım ile Denklem 2.49'daki gibi gerçekleştirir.

$$w_{i,j}(yeni) = w_{i,j}(eski) + (\mu * [d(t) - f(y_t)] * f'(y_t)) \quad (2.49)$$



Şekil 2.23. Geri yayılım ağ modeli

Geri yayılım sinir ağlarında kullanılan aktivasyon fonksiyonunun türevi alınabilir olması gerekir. Çünkü türev bir eğri üzerinde değişimi tarif eder, dolayısıyla hatanın

minimize edilmesi için, hatanın türevinin 0'a gitmesi gerekir. Geri yayılım hatanın türevini  $w_{ij}$  ağırlıkları üzerinde öğrenmekte bu sayede hatalar her bir döngü sonunda 0'a doğru yaklaşmalıdır [107, 110, 111].

Geri yayılım algoritmasında  $y$  üzerinde oluşan hata önce gizli ve çıkış katmanları arasındaki  $w_{pm}$  ağırlıkları üzerine yansıtılır. Ancak güncelleme hemen gerçekleştirilmez.  $w_{pm}$ 'lerin güncelleme yapılmamış olan hallerinde oluşan hatalar ise giriş ve gizli katman arasında yer alan  $v_{pn}$ 'ağırlıkları üzerine yansıtılır, Şekil 2.23.  $w_{pm}$  ve  $v_{pn}$ 'ler aynı anda güncellenerek geri yayılım algoritmasının paralel bir yapı içinde öğrenmesi sağlanır. Bu paralel yapı geri yayılım algoritmasının performansını yükseltir [107, 110, 117].

Geri yayılım algoritması aşağıda maddeler halinde anlatılmıştır [119].

1. Örnekler toplanır, ağırlıkların başlangıç değerleri rastgele atanır.

$$0 < w_{ij} < 1 \quad (2.50)$$

2. Hata minimum olana kadar 3 - 9 arasındaki adımları istenen sayıda tekrarlanır.

3. Her eğitim kümesi verisi için 4. adım ile 8. adım arası uygulanır.

İleri doğru sesaplama:

4. Giriş sinyallerini al ve giriş katmanlarına ilet. Girişler herhangi bir işleme uğramadan ara katmana iletilir.

$$\alpha_k^i = x_i \quad (2.51)$$

5. Gizli katmandaki nöronlara gelen her bir girdi ağırlıklarla  $\{w_1, w_2, \dots, w_n\}$  çarpılarak net girdi aşağıdaki biçimde hesaplanır.

$$\text{net}_j^a = \sum w_{kj} * \alpha_k^i \quad (2.52)$$

$w_{kj}$ : k. girdi katmanı elemanını j. gizli katman elemanına bağlayan ağırlık değerini gösterir.

Aktivasyon fonksiyonu olarak türevi alınabilir bir fonksiyon olan sigmoid fonksiyonu kullanıldığı zaman çıktı aşağıda verildiği gibi olur. Bu eşitlikteki  $\theta_j$  gizli katmanda bulunan j. elemana ait eşik değerinin ağırlığıdır.

$$\alpha_j^a = \frac{1}{1 + e^{-(\text{net}_j^a + \theta_j^a)}} \quad (2.53)$$

Bütün nöronlarda bu hesaplamalar yapıлып en son olarak çıkış katmanına ait çıkış değerleri bulununca ileri doğru hesaplama aşaması sona erer. Gizli katmandaki aktivasyon fonksiyonu ile çıkış katmanındaki aktivasyon fonksiyonu aynı olmak zorunda değildir. Bunlardan birisi sigmoid fonksiyon olurken diğeri tanjant fonksiyonu veya başka bir fonksiyon olabilir.

Geriye doğru hesaplama:

6. Ağa verilen girişler için ağın ürettiği çıkışlar ile beklenen çıkışlar ( $\beta_1, \beta_2, \dots$ ) karşılaştırılarak aradaki fark, başka bir deyişle hata değeri, ağın ağırlık değerlerine dağıtılarak bir sonraki döngüde hatanın azaltılmasına çalışılır. Çıkış katmanındaki m. nöron için hata ( $e_m$ );

$$e_m = \beta_m - \alpha_m \quad (2.54)$$

Bu değer bir işlem elemanı (nöron) için elde edilen hatadır. Toplam hatayı (E) bulmak için çıkış katmanındaki bütün hataların toplanması gerekir. Ağın eğitilmesindeki temel amaç bu hatayı minimize etmektir. Toplam hata Denklem 2.55'deki gibi formülüne edilir:

$$E = \frac{1}{2} \sum_m e_m^2 \quad (2.55)$$

Toplam hatayı minimize etmek için hata nöronlara dağıtılır. Bu işlem nöron ağırlıkları değiştirilerek olur. Bu ise gizli katman ile çıkış katmanı arasındaki ve gizli katmanlar arası veya gizli katman giriş katmanı arasındaki ağırlıkların değiştirilmesiyle sağlanır.

7. Ara katmanın j. nöronunu çıktı katmanındaki m. nörona bağlayan bağlantı ağırlıklarındaki değişim miktarına  $\Delta w_{jm}^a$  dersek t. iterasyonda ağırlığın değişim miktarı aşağıdaki gibi hesaplanır:

$$\Delta w_{jm}^a(t) = \lambda \delta_m \alpha_j^a + \mu \Delta w_{jm}^a(t-1) \quad (2.56)$$

Bu eşitlikteki  $\lambda$  öğrenme katsayısı,  $\mu$  da momentum katsayısıdır. Öğrenme katsayısı ağın performansında etkilidir. Eğitim süresini, küçük öğrenme katsayısı uzatırken, büyük öğrenme katsayısı kısaltır. Ancak çok büyük öğrenme katsayısı yakınsamayı engeller. Momentum katsayısı ise ağın öğrenim esnasında yerel bir minimum noktaya takılıp kalmamasında etkilidir. Ağırlık değişim değerinin belli bir oranda bir sonraki değişime eklenmesini sağlar ve toplam hata sıfıra daha fazla bir eğilimle yaklaşır. Eşitlikteki  $\delta_m$  ise m. çıktı ünitesinin hatasıdır.

$$\delta_m = f'(\text{net}) \cdot e_m \quad (2.57)$$

Burada,  $f'(\text{net})$  aktivasyon fonksiyonun türevidir. Sigmoid aktivasyon fonksiyonu kullanıldığı zaman bu eşitlik;

$$\delta_m = \alpha_m (1 - \alpha_m) \cdot e_m \quad (2.58)$$

biçiminde olur.

8. Değişim miktarı hesaplandıktan sonra ağırlıkların t. döngüdeki yeni değerleri Denklem 2.59'daki gibi olacaktır.

$$w_{jm}^a(t) = w_{jm}^a(t-1) + \Delta w_{jm}^a(t) \quad (2.59)$$

İkinci durumda ise girdi katmanı ile ara katman veya iki ara katman arasındaki ağırlıklar değiştirilirken çıkış katmanındaki tüm elemanlara ait hataların hesaba katılması gerekir. Bu ağırlık değişimi örneğin giriş katmanı ile ara katman arasındaki  $\Delta w^i$  ile gösterilirse değişim miktarı aşağıdaki denklemde verilmiştir.

$$\Delta w_{kj}^i(t) = \lambda \delta_j^a \alpha_k^i + \mu \Delta w_{kj}^i(t-1) \quad (2.60)$$

$\delta^a$  faktörü ise;

$$\delta_j^a = f'(\text{net}) \sum_m \delta_m w_{jm}^a \quad (2.61)$$

şeklindedir ve sigmoid fonksiyonu kullanıldığında aşağıdaki gibi olur.

$$\delta_j^a = \alpha_j^a (1 - \alpha_j^a) \sum_m \delta_m w_{jm}^a \quad (2.62)$$

Ağırlıkların yeni değerleri ise Denklem 2.63'deki gibidir.

$$w_{kj}^i(t) = w_{kj}^i(t-1) + \Delta w_{kj}^i(t) \quad (2.63)$$

Ara katman eşik değer ağırlıkları  $\theta^a$  ile gösterilirse değişim miktarı Denklem 2.64'deki gibi olur.

$$\Delta\theta_j^a(t) = \lambda\delta_j^a\alpha_k^i + \mu\Delta\theta_j^a(t-1) \quad (2.64)$$

t. iterasyon için ağırlıkların yeni değerleri ise aşağıdaki gibi hesaplanır.

$$\theta_j^a(t) = \theta_j^a(t-1) + \Delta\theta_j^a(t) \quad (2.65)$$

9. Eğer hata, önceden belirlenen hata toleransına ulaşmışsa veya sonlandırma kriteri sağlanmışsa eğitim durdurulur. Aksi halde 4. Adımdan başlayarak işlem tekrarlanır.

Bu basamaklar çok katmanlı algılayıcıların öğrenmesi tamamlanıncaya kadar, yani gerçekleşen çıkışlar ile beklenen çıkışlar arasındaki hataların kabul edilebilir bir düzeye inmesine kadar devam eder. Ağın öğrenmesi için bir durdurma kriterinin olması gerekmektedir. Bu durum ise genellikle hatanın belirli bir düzeyin altına düşmesi olarak alınır.



### 3. TEZDE KULLANILAN YÖNTEMLER

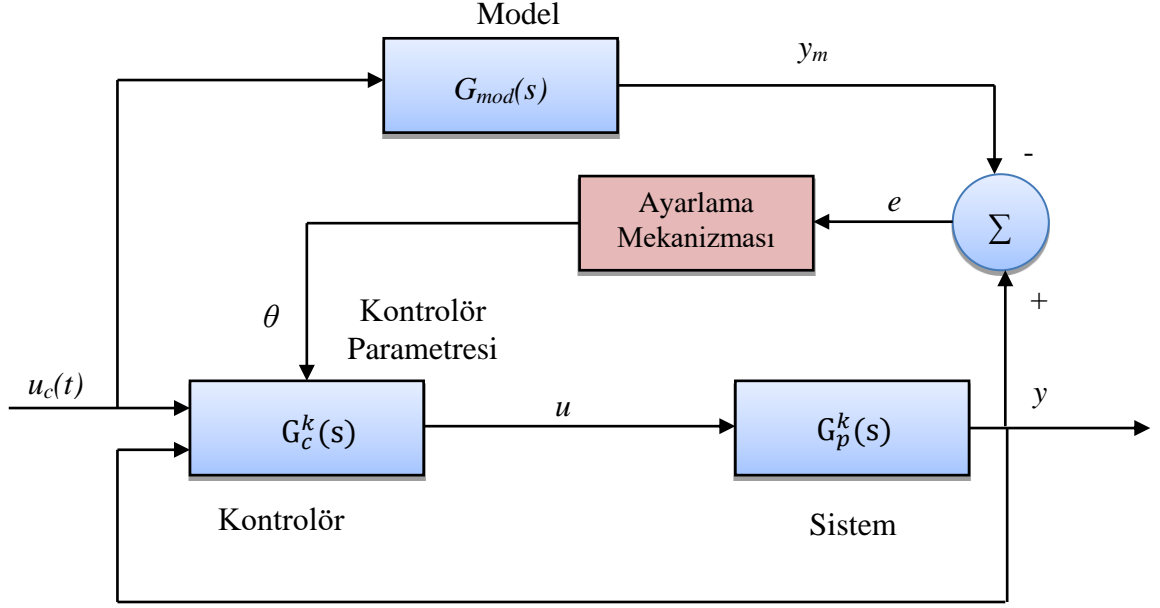
Bu tezde, Kesir dereceli PID kontrolör tasarımı için Model Referans Uyarlamalı Kontrol (MRAC), Uyarlamalı Sinirsel Bulanık Çıkarım Sistemi (ANFIS) ve Otomatik Ayarlamalı Sinir Hücresi yöntemleri kullanılmıştır.

#### 3.1. Model Referans Uyarlamalı Kontrol (MRAC)

Model Referans Uyarlamalı Kontrol (MRAC) daha çok sistem parametrelerinde ve giriş işaretinde belirsizliklerin ve önceden öngörülmeven değişimlerin bulunduğu durumları barındıran sistemlerde kullanılır. Uyarlamalı kontrolörler basit olarak sayısal donanım ile gerçekleştirilebilirler. Bu nedenle bu konuya olan ilgi son zamanlarda oldukça artmıştır. Mikroelektronik elemanlardaki maliyetin azalması ve hızlarının artması uyarlamalı kontrolörlerin tasarımını kolaylaştırmıştır. Böylece bu kontrolör için geniş endüstriyel uygulamalar yapılmıştır.

Uyarlamalı kontrolün kullanıldığı alanlar uçak kontrolü, füze kontrolü, elektriksel sürücülerin kontrolü, metalurjiye ait işlemlerin kontrolü örnek olarak verilebilir. [120]'da bir İHA'nın (İnsansız Hava Aracı) uzunlamasına ve yanlamasına hareketi için uyarlamalı model referans yöntemi kullanılmış, çeşitli otomatik uçuş kontrol algoritmaları geliştirilmiş ve bilgisayar ortamında uygulamaları yapılmıştır. [121]'de ise model referans uyarlamalı kontrolün klasik yöntemlerinden biri olan Lyapunov kararlılık kriteri ve akıllı yöntemlerden biri olan bulanık sistem yaklaşımıyla F-16 uçağının boylamasına doğrusal modeli kullanılarak adaptif kontrol çalışması yapılmıştır.

Şekil 3.1'de gösterilen MRAC yapısında sistem çıkışı ( $y$ ) ile model çıkışı ( $y_m$ ) arasındaki fark, hata olarak ayarlama mekanizmasına iletilir. Ayarlama mekanizması hataya göre kontrolör parametresini güncelleyerek hatayı sıfırlamaya veya minimuma düşürmeye çalışır. Burada,  $G_{\text{mod}}(s)$  referans modelin,  $G_c^k(s)$  kontrolörün ve  $G_p^k(s)$  ise kontrol edilecek sistemin transfer fonksiyonunu,  $r$  referans sinyali,  $u$  kontrol sinyalini,  $y$  sistem çıkışını,  $y_m$  referans model çıkışını,  $e$  referans model ile kontrol edilen sistem arasındaki hatayı,  $\theta$  parametre vektörünü göstermektedir.



Şekil 3.1. Model Referans Uyarlamalı Kontrol (MRAC) blok diyagramı

Bu tezde MRAC için MIT kuralı kullanılmıştır. Bu kurala göre hata,

$$e = y - y_m \quad (3.1)$$

ise,

$$J(\theta) = \frac{1}{2} e^2(\theta) \quad (3.2)$$

olur. Burada  $J$  toplam karesel hata ve  $\theta$  parametre vektörünün bir fonksiyonudur. Karesel hata değerini küçültmek için parametrelerin  $J$ 'nin negatif eğimi doğrultusunda ayarlanması gerekir.

$\theta$  parametresinin zamana göre türevi  $J$ 'nin negatif yöndeki değişimine eşittir. Bu durumda karesel hata fonksiyonundaki değişim Denklem (3.3)'deki gibi yazılır.

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (3.3)$$

Burada  $\frac{\partial e}{\partial \theta}$  hassasiyet türevini ifade eder. Bu ifade de hesaplandıktan sonra  $\theta$  parametresi Denklem (3.4)'deki gibi olur [44, 122].

$$\theta = -\frac{\gamma}{s} . e . y_m \quad (3.4)$$

Burada  $e$ , hatanın  $\theta$  parametresine göre kısmi türevi ve  $\theta$  parametresinin nasıl değişeceğini belirler. Kesir dereceli türev operatörü içeren tek çıkışlı bir sistem MRAC

algoritması ile kontrol edildiğinde, kapalı çevrim sistem kararlılığının arttığı ve bozucu etkilerinin azaldığı gözlemlenmiştir.

Kesir dereceli integral kullanan MRAC algoritması Denklem (3.1-3.4)'de hesaplandığı gibi;

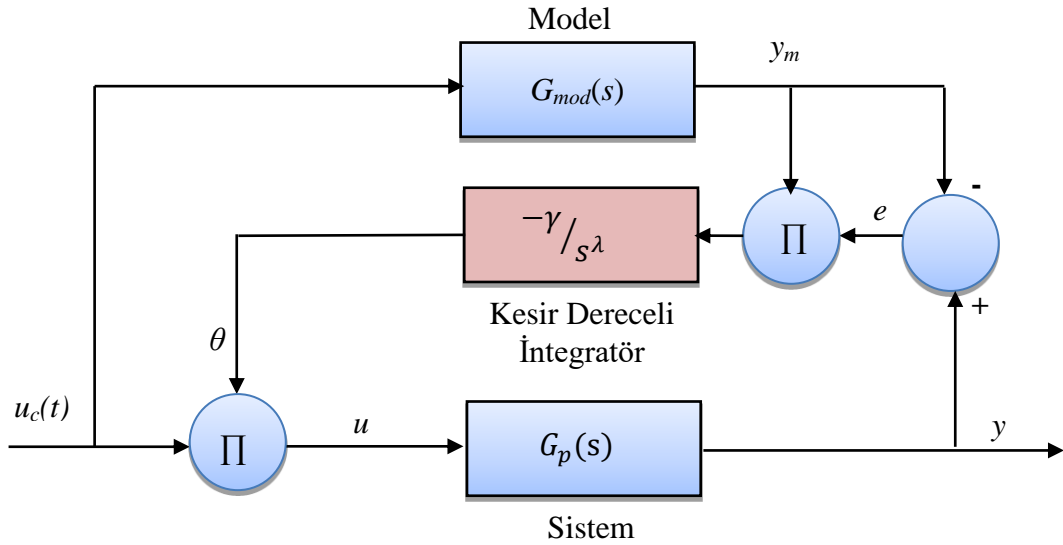
$$e = y - y_m \quad (3.5)$$

$$J(\theta) = \frac{1}{2} e^2(\theta) \quad (3.6)$$

$$\frac{d^\alpha \theta}{dt^\alpha} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma \cdot e \cdot \frac{\partial e}{\partial \theta} = -\lambda \cdot e \cdot y_m \quad (3.7)$$

$$\theta = -\frac{\gamma}{s^\alpha} \cdot e \cdot y_m \quad (3.8)$$

bulunur. Bulunan bu sonuçlara göre kesir dereceli integral içeren MRAC blok diyagramı Şekil 3.2'de gösterilmiştir.

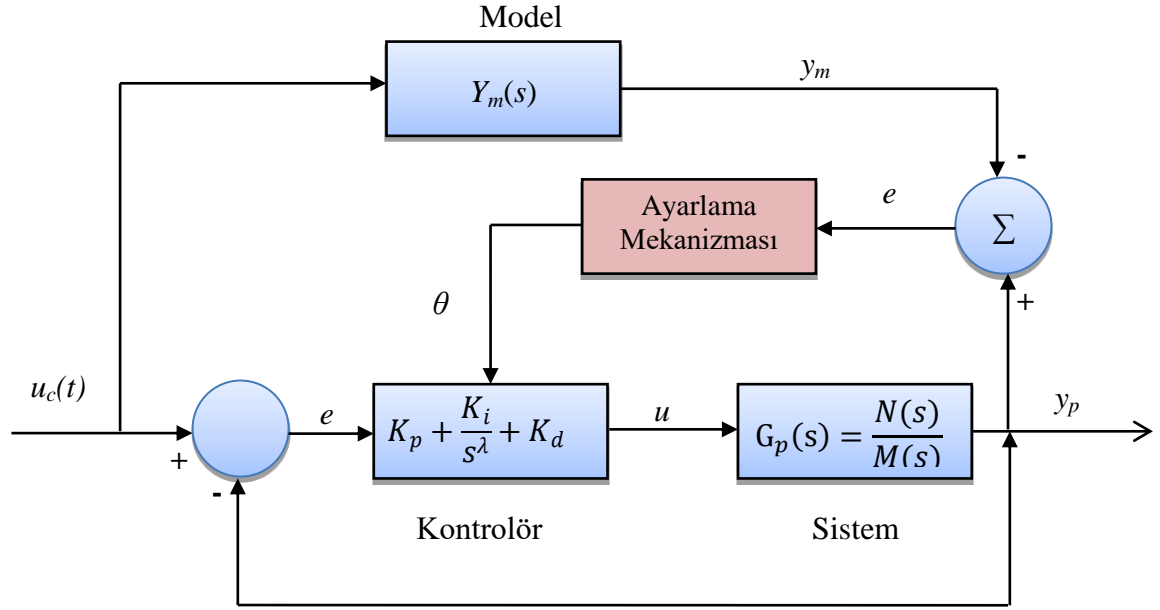


Şekil 3.2. Kesir dereceli integral kullanan MRAC algoritması

Şekil 3.2'de  $\gamma$  sabitinin değeri yüksek alınırsa sistem osilasyona ve kararsızlığa geçecektir.  $\gamma$ 'nin ayarlanması adaptasyon ve kararlılık için önemlidir. Bu nedenle  $\gamma$  sabiti uygun seçilmelidir.

### 3.1.1. Model Referans Uyarlamalı Kontrol (MRAC) ile $PI^\lambda D^\mu$ kontrollör Tasarımı (MRA- $PI^\lambda D^\mu$ )

Kesir dereceli PID kontrolörün  $K_p$ ,  $K_i$ ,  $K_d$  katsayılarının model referans uyarlama yöntemi kullanarak ayarlanmasına ilişkin sistemin blok diyagramı Şekil 3.3'de verilmiştir.



Şekil 3.3. Model referans uyarlamalı  $PI^\lambda D^\mu$  blok diyagramı [123].

Şekil 3.1'deki yüksek dereceli kontrol sistemleri için  $G_c^k(s)$  kesir dereceli PID kontrolörün,  $G_{\text{mod}}(s)$  referans modelin ve  $G_p^k(s)$  kontrol edilecek sistemin transfer fonksiyonları ise

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (3.9)$$

$$G_p(s) = \frac{b_m^p s^m + b_{m-1}^p s^{m-1} + \dots + b_1^p s + b_0^p}{a_n^p s^n + a_{n-1}^p s^{n-1} + \dots + a_1^p s + 1} = \frac{N_p(s)}{D_p(s)} \quad (3.10)$$

$$G_{\text{mod}}(s) = \frac{b_m^m s^m + b_{m-1}^m s^{m-1} + \dots + b_1^m s + b_0^m}{a_n^m s^n + a_{n-1}^m s^{n-1} + \dots + a_1^m s + 1} = \frac{N_m(s)}{D_m(s)} \quad (3.11)$$

şeklinde yazılabilir. Birim geri beslemeli bir kontrol sisteminin transfer fonksiyonu:

$$Y(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} U_c(s) \quad (3.12)$$

gibi olur. Denklem (3.9) ve Denklem (3.10) Denklem (3.12)' de yerlerine konursa Denklem (3.13) bulunur.

$$Y(s) = \frac{(K_p s^\lambda + K_d s^{\lambda+\mu} + K_i) N_p(s)}{D_p(s) s^\lambda + (K_p s^\lambda + K_d s^{\lambda+\mu} + K_i) N_p(s)} U_c(s) \quad (3.13)$$

Denklem (3.3)' ten  $\theta$ ' nın zamana göre türevi;

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma \frac{\partial J}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial \theta} \quad (3.14)$$

olur. Denklem (3.5 ve 3.6)' dan  $\frac{\partial J}{\partial e} = e$  ve  $\frac{\partial e}{\partial y} = 1$  olarak bulunur. Bu ifadeleri

Denklem (3.14)'de yerine yazarsak;

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial y}{\partial \theta} \quad (3.15)$$

olur.  $\theta = [K_p \ K_i \ K_d]$  olduğuna göre Denklem (3.15) kullanarak  $K_p$ ,  $K_i$  ve  $K_d$  parametreleri için aşağıdaki denklemler yazılabilir.

$$\frac{\partial K_p}{\partial t} = -\alpha \frac{\partial J}{\partial K_p} = -\alpha \frac{\partial J}{\partial e} \cdot \frac{\partial e}{\partial y_p} \cdot \frac{\partial y_p}{\partial K_p} \quad (3.16)$$

$$\frac{\partial K_i}{\partial t} = -\beta \frac{\partial J}{\partial K_i} = -\beta \frac{\partial J}{\partial e} \cdot \frac{\partial e}{\partial y_p} \cdot \frac{\partial y_p}{\partial K_i} \quad (3.17)$$

$$\frac{\partial K_d}{\partial t} = -\gamma \frac{\partial J}{\partial K_d} = -\gamma \frac{\partial J}{\partial e} \cdot \frac{\partial e}{\partial y_p} \cdot \frac{\partial y_p}{\partial K_d} \quad (3.18)$$

Denklem 3.13'den faydalanarak Denklem 3.19, 3.20 ve 3.21 bulunur.

$$\frac{\partial Y}{\partial K_p} = s^\lambda \frac{N_p(s)}{D_p(s)} (U_c - Y) \quad (3.19)$$

$$\frac{\partial Y}{\partial K_i} = \frac{N_p(s)}{D_p(s)} (U_c - Y) \quad (3.20)$$

$$\frac{\partial Y}{\partial K_d} = s^{\lambda+\mu} \frac{N_p(s)}{D_p(s)} (U_c - Y) \quad (3.21)$$

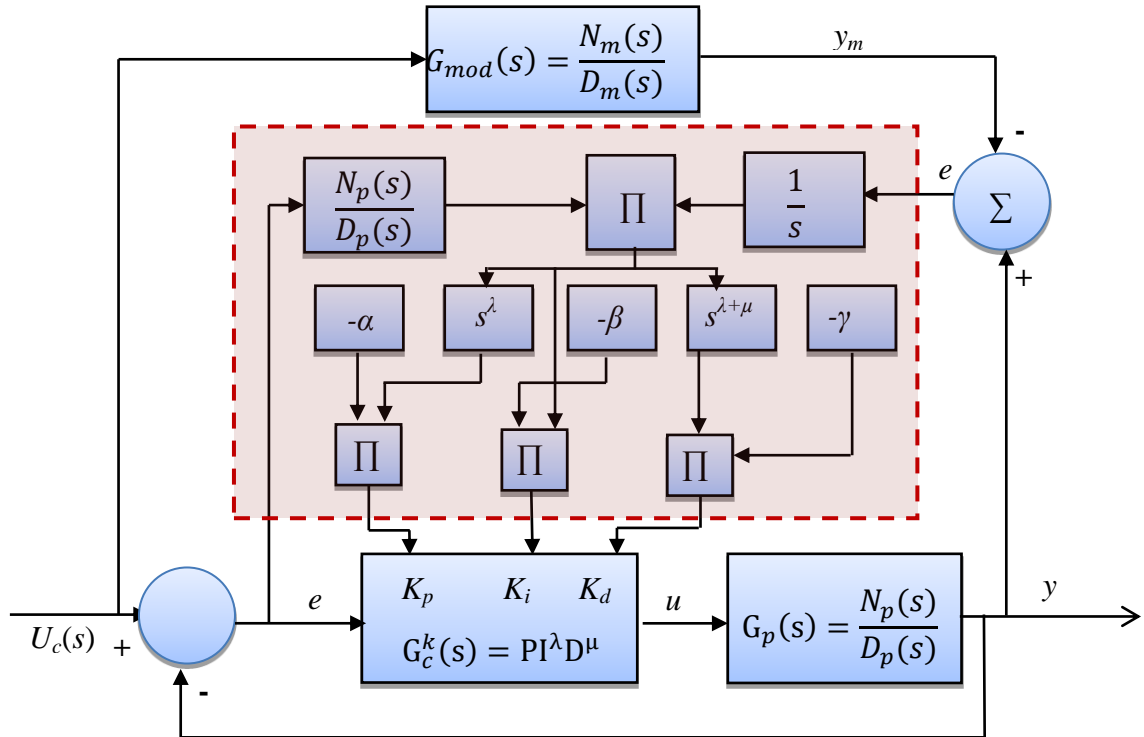
Denklem 3.19, 3.20 ve 3.21, Denklem 3.16, 3.17 ve 3.18'de yerine konulursa

$$K_p = -\alpha s^\lambda \frac{e N_p(s)}{s D_p(s)} (U_c - Y) \quad (3.22)$$

$$K_i = -\beta \frac{e N_p(s)}{s D_p(s)} (U_c - Y) \quad (3.23)$$

$$K_d = -\gamma s^{\lambda+\mu} \frac{e N_p(s)}{s D_p(s)} (U_c - Y) \quad (3.24)$$

elde edilir. Böylece uyarlamalı yöntem ile ayarlanacak  $PI^\lambda D^\mu$  kontrolörünün katsayıları bulunmuş olur. Denklem (3.22), (3.23) ve (3.24) kullanarak kesirli dereceli MRA- $PI^\lambda D^\mu$  kontrolörünün kontrol sisteminin blok diyagramı Şekil 3.4'deki gibi olur [123].



Şekil 3.4. Kesirli dereceli MRA- $PI^\lambda D^\mu$  kontrolörünün blok diyagramı [123].

### 3.2. Uyarlamalı Sinirsel Bulanık Çıkarım Sistemi (ANFIS) ile $PI^\lambda D^\mu$ kontrollör (ANFIS- $PI^\lambda D^\mu$ ) Tasarımı

Sinirsel-Bulanık sistemler, bir sistemi modellemek veya kontrol etmek için sinir ağları ve bulanık mantıktan türetilmiş bir öğrenme algoritması kullanan yapıdır. Bu yapı kontrol işlemini daha etkin duruma getirebilmek için bulanık mantık teorisi ve sinir ağlarının aşağıda verilen avantajlarını birlikte kullanır [124-127].

- Sayısal, sözel, mantıksal vb. işlem ve değerleri kullanır.
- Kesin olmayan, kısmi, belirsiz veya eksik bilgileri kullanabilir.
- Çelişkileri çözebilme yeteneğine sahiptir.
- Kendi kendine öğrenme, organize olma ve ayarlama kabiliyeti vardır.
- Veri ilişkilerinin ön bilgilerine ihtiyaç yoktur.
- Karar alma sürecinde insanın karar vermesini taklit eder.
- Bulanık mantık operatörleri (min, max gibi) kullanarak hızlı işlem yapar.

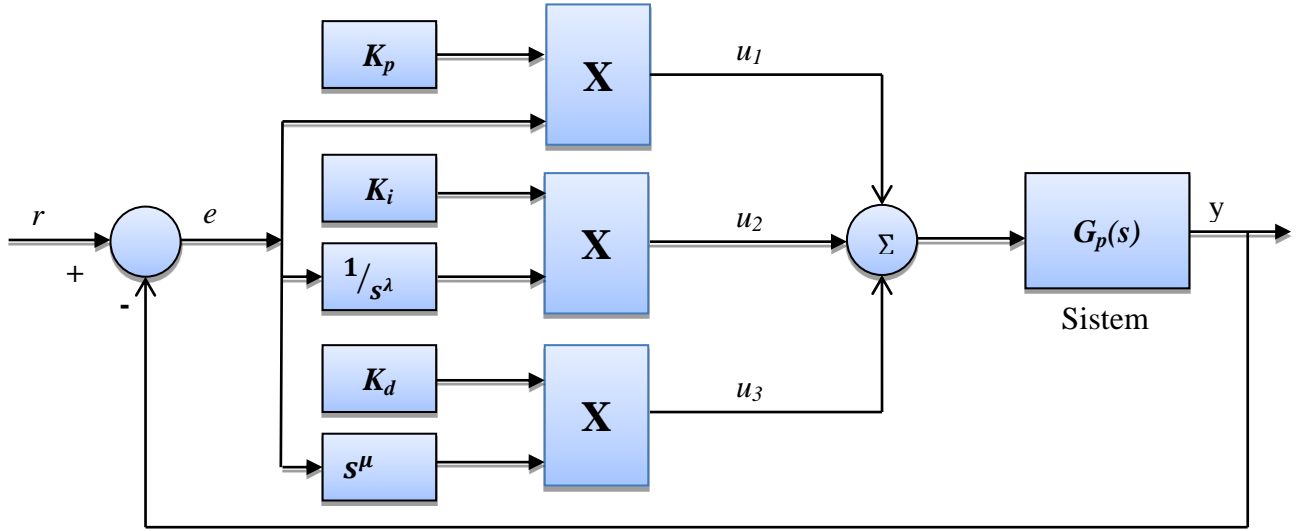
Sinirsel-bulanık sistemler, genellikle çok katmanlı ileri beslemeli özel sinir ağları şeklindedirler. Farklı yaklaşımlar olmakla birlikte sinirsel-bulanık sistem terimi aşağıdaki özellikleri içerir:

- I. Bir sinirsel-bulanık sistemi, sinir ağları teorisinden gelen bir öğrenme algoritması ile eğitilen bir bulanık sistemini esas alır.
- II. 3 katmanlı ileri beslemeli bir sinir ağı olarak düşünülür (bazı durumlarda 5 katmanlı mimaride kullanılır). İlk katman giriş değişkenlerini temsil eder, orta katman (gizli katman da denir) bulanık kurallarını ve üçüncü katman çıkış değişkenlerini temsil eder. Bulanık kümeler bağlantılı ağırlıklar olarak kodlanır.
- III. Genellikle öğrenme öncesi, öğrenme sırasında ve öğrenme sonrası bir bulanık kurallar sistemi olarak yorumlanır.

Denklem 3.25'de  $PI^\lambda D^\mu$  kontrolörün transfer fonksiyonu verilmiştir.

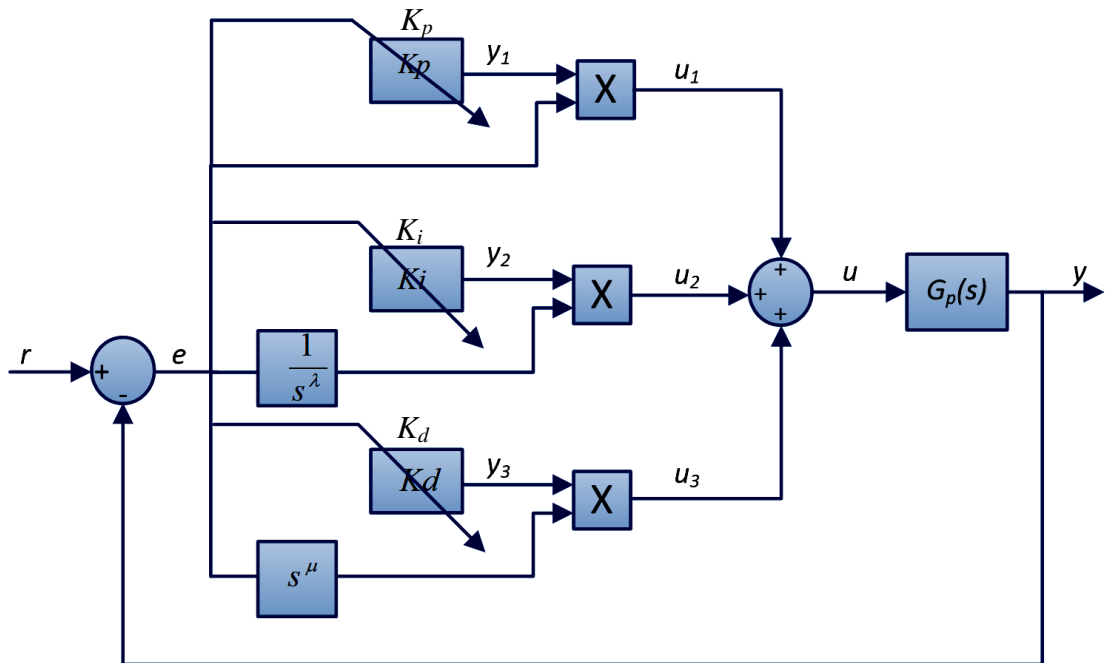
$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) = u_1 + u_2 + u_3 \quad (3.25)$$

Denklem 3.25'e göre  $PI^\lambda D^\mu$  kontrolör içeren sistemin blok diyagramı Şekil 3.5'deki gibi gösterilir:



Şekil 3.5.  $PI^\lambda D^\mu$  kontrolör içeren sistemin blok diyagramı [130]

$e$  (hata) değişkeni kullanarak  $K_p$ ,  $K_i$  ve  $K_d$  katsayıları ayarlanan  $PI^\lambda D^\mu$  denetleyici sisteminin blok diyagramı Şekil 3.6'da verilmiştir. Burada  $e$ 'nin değişimine göre  $K_p$ ,  $K_i$  ve  $K_d$  katsayıları ayarlanmaktadır. Bu katsayılar hata ile çarpılarak  $PI^\lambda D^\mu$  denetleyicinin oransal, integral ve türev operatörlerini oluşturmaktadır.



Şekil 3.6.  $K_p$ ,  $K_i$  ve  $K_d$  katsayıları çevrimiçi ayarlanabilen  $PI^\lambda D^\mu$  kontrollör sisteminin blok diyagramı [130]



Bir bulanık model veya bulanık çıkarım sistemi, IF ... THEN... öncül ve sonuç olarak çalışan bulanık kurallarını kullanır [128, 129]. İki girişli, tek çıkışlı bir kontrolörün kuralları aşağıdaki gibidir:

$$\text{Kural 1: If } x_1 = A_1 \text{ and } x_2 = B_1 \text{ then } z = C_1 \quad (3.26)$$

$$\text{Kural 2: If } x_1 = A_2 \text{ and } x_2 = B_2 \text{ then } z = C_2$$

Burada  $A_1, B_1, A_2$  ve  $B_2$  girişler için üyelik fonksiyonları (öncül),  $C_1$  ve  $C_2$  çıkış için üyelik fonksiyonlarıdır (sonuç). Bulanık IF - THEN kurallarının bir başka gösterimi Takagi ve Sugeno tarafından önerilmiştir [105]. Bu kural yapısında girdiler yine üyelik fonksiyonlarıyla gösterilir. Ancak çıktılar matematiksel olarak ifade edilir. İki girişli tek çıkışlı ve iki kurallı bir sistem için Takagi - Sugeno tipi kural yapısı aşağıdaki gibi tanımlanabilir.

$$\text{Kural 1: If } x_1 \text{ is } A_1(x_1) \text{ and } x_2 \text{ is } B_1(x_2) \text{ then } f_1(x_1, x_2) = p_1x_1 + q_1x_2 + z_1 \quad (3.27)$$

$$\text{Kural 2: If } x_1 \text{ is } A_2(x_1) \text{ and } x_2 \text{ is } B_2(x_2) \text{ then } f_2(x_1, x_2) = p_2x_1 + q_2x_2 + z_2$$

$n$  tane kural  $p_i = q_i = 0$  ( $i = 1, \dots, n$ ) ve tek girişli ( $e$ ) sistem için aşağıdaki gibi yazılabilir.

$$\text{Kural 1: if error} = A_1(e) \text{ then } f_1(e) = z_1$$

$$\text{Kural 2: if error} = A_2(e) \text{ then } f_2(e) = z_2$$

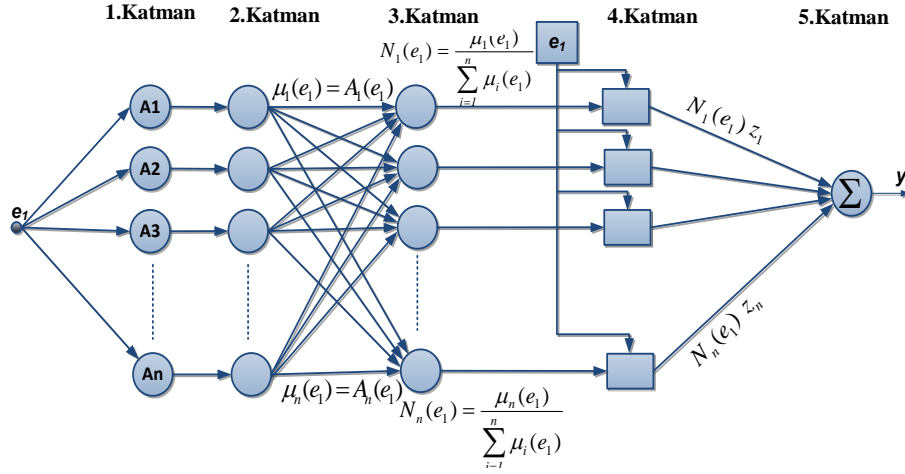
$$\text{Kural 3: if error} = A_3(e) \text{ then } f_3(e) = z_3$$

$$(3.28)$$

.....  
 .....

$$\text{Kural } n: \text{ if error} = A_n(e) \text{ then } f_n(e) = z_n$$

$n$  tane kurala sahip, ( $A_1, A_2, \dots, A_n$ ) giriş üyelik ve ( $f_1, f_2, \dots, f_n$ ) çıkış fonksiyonlarına dayalı ANFIS yapısı Şekil 3.7'de gösterilmiştir [106].



Şekil 3.7. Tek Giriş-Tek Çıkış ANFIS yapısı [130]

Bu ANFIS yapısı beş katmana sahiptir. Burada  $A_1, A_2, \dots, A_n$  sözel etiketlerdir ve  $e$  evrensel küme içinde üyelik fonksiyonları olarak dağılımı yapılmıştır.  $z_1, \dots, z_n$  sistemin en iyi performansla kontrolüne yönelik eğitim sonucunda belirlenmiş çıkış fonksiyonlarının parametreleridir. Burada;

1. katmandaki her düğümün çıkışı, kullanılan giriş ve çıkış değerlerine bağlı üyelik derecesinden oluşur. Her düğüm,  $A_i(e)$  gibi bulanık üyelik fonksiyonlarını kullanır.

2. katmanın her düğümünde, kurallara göre elde edilen ağırlıklar hesaplanır.

$$\mu_{A_1} = A_1(e_1), \mu_{A_2} = A_2(e_1), \mu_{A_3} = A_3(e_1), \dots, \mu_{A_n} = A_n(e_1) \quad (3.29)$$

3. katmanda her düğüm bir önceki katmanın ağırlığının önceki katmanların toplam ağırlığına oranını belirler ve ağırlıkları normalize eder.

$$\text{Normalize 1: } N_1(e_1) = \frac{\mu_{A_1}}{\mu_{A_1} + \mu_{A_2} + \mu_{A_3} + \dots + \mu_{A_n}} = \frac{\mu_{A_1}}{\mu_{SUM}}$$

$$\text{Normalize 2: } N_2(e_1) = \frac{\mu_{A_2}}{\mu_{A_1} + \mu_{A_2} + \mu_{A_3} + \dots + \mu_{A_n}} = \frac{\mu_{A_2}}{\mu_{SUM}}$$

(3.30)

$$\text{Normalize 3: } N_3(e_1) = \frac{\mu_{A_3}}{\mu_{A_1} + \mu_{A_2} + \mu_{A_3} + \dots + \mu_{A_n}} = \frac{\mu_{A_3}}{\mu_{SUM}}$$

.....  
 .....

$$\text{Normalize n: } N_n(e_1) = \frac{\mu_{A_n}}{\mu_{A_1} + \mu_{A_2} + \mu_{A_3} + \dots + \mu_{A_n}} = \frac{\mu_{A_n}}{\mu_{SUM}}$$

4. katman, her uyarlamalı  $i$  düğümü için Denklem (3.28)' de verilen fonksiyonlara sahiptir.

$$f_1(e) = z_1, f_2(e) = z_2, f_3(e) = z_3, \dots, f_n(e) = z_n \quad (3.31)$$

5. katman, 4. katmandaki düğümlerden gelen tüm sinyallerin toplamı olan çıktı katmanıdır.

$$y(e_1) = N_1(e_1).z_1 + N_2(e_1).z_2 + N_3(e_1).z_3 + \dots + N_n(e_1).z_n \quad (3.32)$$

$$= N_1(e_1).f_1(e_1) + N_2(e_1).f_2(e_1) + N_3(e_1).f_3(e_1) + \dots + N_n(e_1).f_n(e_1)$$

$$y(e_1) = \sum_{i=1}^n N_i(e_1).f_i(e_1) = \frac{\sum_{i=1}^n \mu_i(e_1).f_i(e_1)}{\sum_{i=1}^n \mu_i(e_1)} = \frac{\sum_{i=1}^n A_i(e_1).f_i(e_1)}{\sum_{i=1}^n A_i(e_1)} \quad (3.33)$$

$PI^\lambda D^\mu$  kontrollörün girişindeki hata,

$$e(t) = r(t) - y(t) \quad (3.34)$$

şeklinde yazılır.

$PI^\lambda D^\mu$  kontrolörünün transfer fonksiyonu Denklem (3.33) kullanılarak

$$u(t) = \frac{\sum_{i=1}^n A_i(e_1).f_i(e_1)}{\sum_{i=1}^n A_i(e_1)} e(t) + \frac{\sum_{j=1}^m B_j(e_1).f_j(e_1)}{\sum_{j=1}^m B_j(e_1)} D^{-\lambda} e(t) + \frac{\sum_{k=1}^v C_k(e_1).f_k(e_1)}{\sum_{k=1}^v C_k(e_1)} D^\mu e(t) \quad (3.35)$$

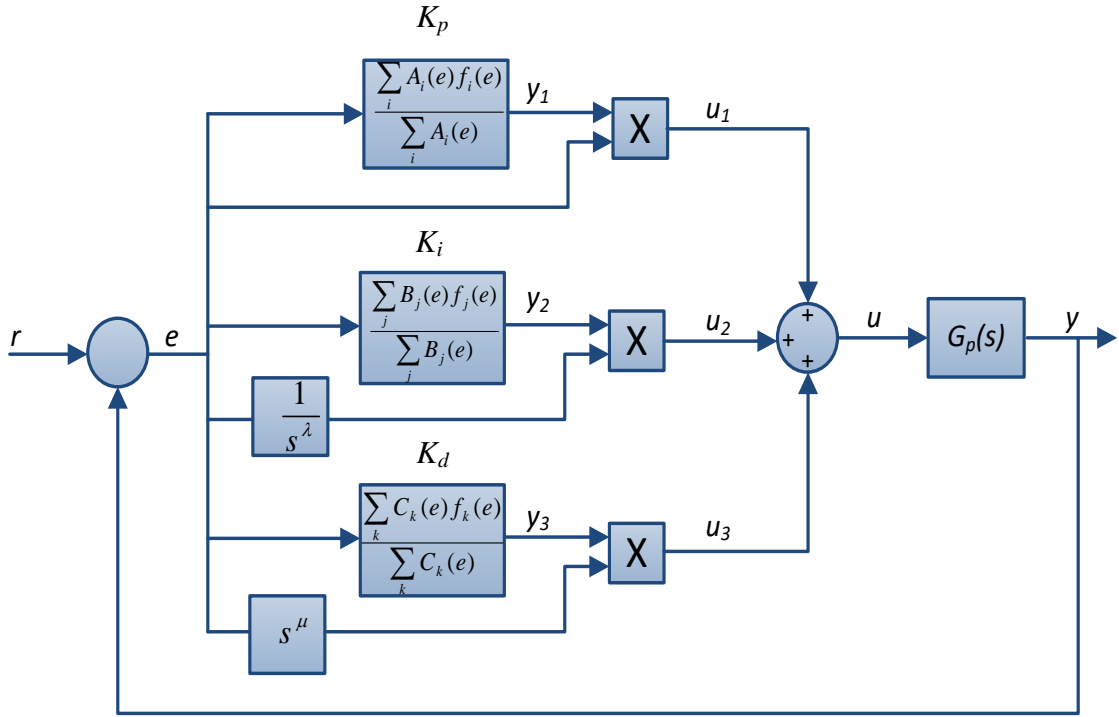
gibi tanımlanabilir.  $PI^\lambda D^\mu$  kontrolörün katsayıları

$$K_p(e) = \frac{\sum_{i=1}^n A_i(e).f_i(e)}{\sum_{i=1}^n A_i(e)}, K_i(e) = \frac{\sum_{j=1}^m B_j(e).f_j(e)}{\sum_{j=1}^m B_j(e)}, K_d(e) = \frac{\sum_{k=1}^v C_k(e).f_k(e)}{\sum_{k=1}^v C_k(e)} \quad (3.36)$$

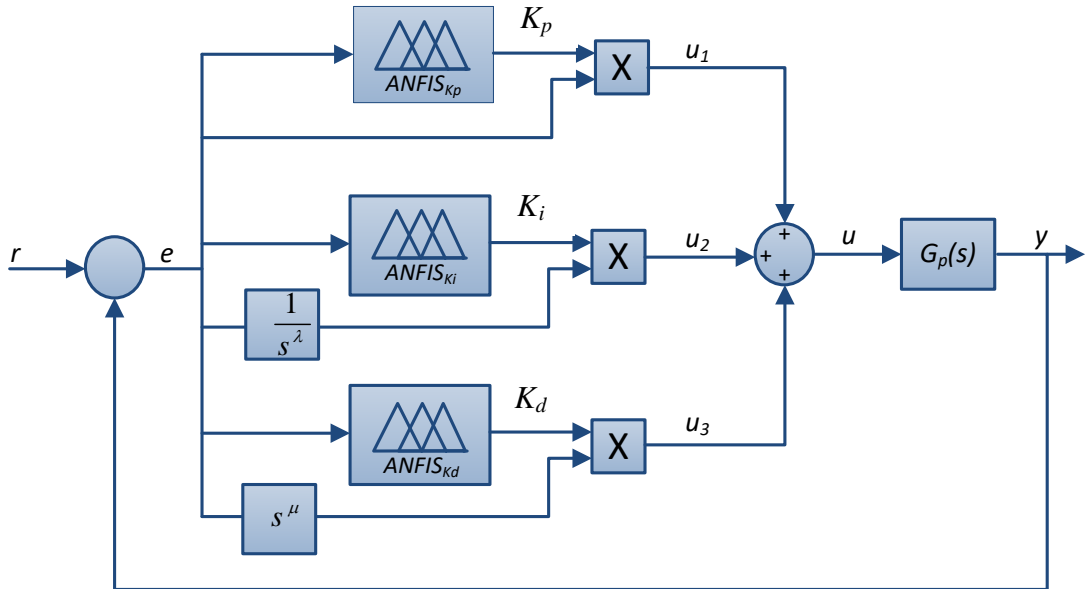
olur. Burada  $[A_i(e), f_i(e)]$ ,  $[B_j(e), f_j(e)]$  ve  $[C_k(e), f_k(e)]$   $K_p$ ,  $K_i$  ve  $K_d$  katsayıları için sırasıyla üyelik ve çıkış fonksiyonlarıdır.

Denklem (3.35) ve (3.36) denklemleri kullanarak  $PI^\lambda D^\mu$  kontrollör sisteminin blok diyagramı Şekil 3.8 ve 3.9'da verilmiştir. Burada  $f_i(e)$ ,  $f_j(e)$  ve  $f_k(e)$  Sugeno tipi çıkış

fonksiyonlarını temsil etmektedir. Bu kontrolörlerin çıkış katsayıları  $K_p$ ,  $K_i$  ve  $K_d$ , ANFIS algoritması ile tanımlanabilir. Öğrenme yöntemiyle katsayıları belirlemek için gerekli veriler Şekil 3.10’da verilen kontrolörden elde edilir [130].

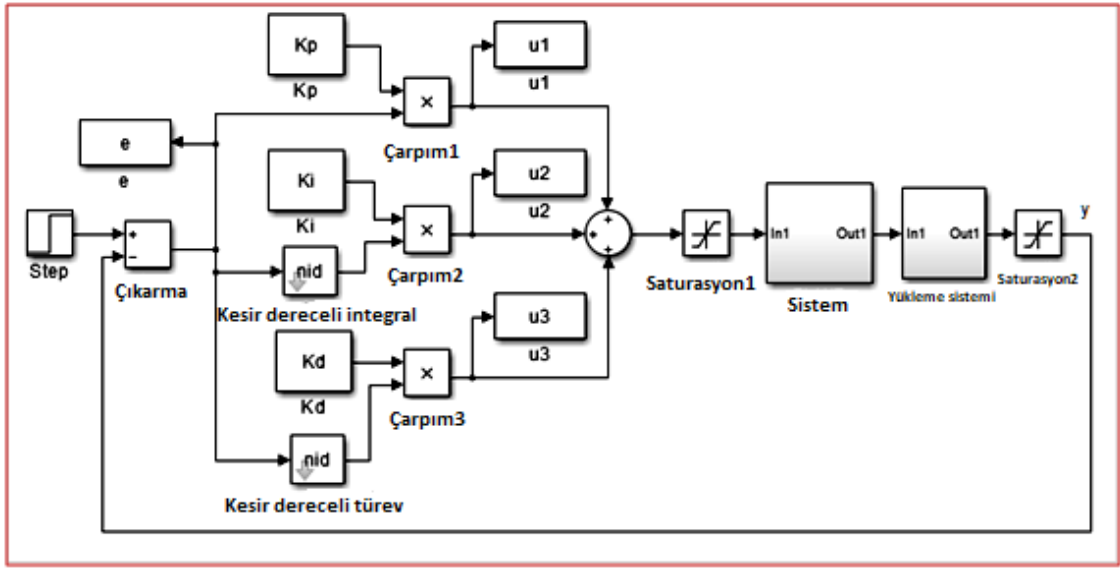


Şekil 3.8. ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolör tasarımı [130]



Şekil 3.9. ANFIS-PI<sup>λ</sup>D<sup>μ</sup>’nin  $K_p$ ,  $K_i$  ve  $K_d$  parametrelerinin ayarlanması [130]

Şekil 3.8’deki  $y_1$ ,  $y_2$  ve  $y_3$  yerine, Şekil 3.9’da gösterildiği gibi Sugeno tipi bulanık denetleyiciler kullanılmıştır.



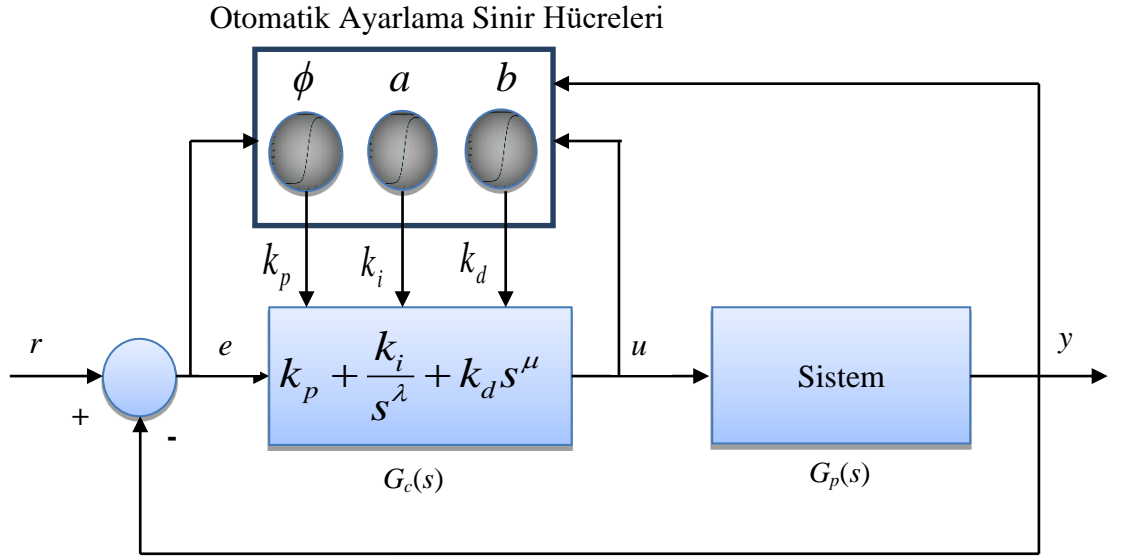
Şekil 3.10. Öğrenme verilerini elde etmek için kullanılan  $PI^\lambda D^\mu$  kontrollör sisteminin blok diyagramı [130]

ANFIS- $PI^\lambda D^\mu$  kontrollör tasarımı  $PI^\lambda D^\mu$  kontrollörün  $K_p$ ,  $K_i$  ve  $K_d$  katsayıları genetik algoritma ile bulunan kontrol sisteminden eğitim verileri alınmış ve her katsayı için alınan bu eğitim verilerinden birer ANFIS modeli oluşturulmuştur. Şekil 3.10'daki sistemde öğrenme için gerekli veriler  $(e, u_1)$ ,  $(e, u_2)$ ,  $(e, u_3)$  olarak alınmıştır. Burada,  $e$  ANFIS' in girişini,  $u_1$  (oransal),  $u_2$  (integral) ve  $u_3$  (türevsel) ANFIS' in çıkış öğrenme verilerini temsil etmektedir. Bu çalışmada  $(e, u_1)$ ,  $(e, u_2)$ ,  $(e, u_3)$  eğitim veri çiftleri için öğrenme süreci MATLAB - ANFIS editörü kullanılarak gerçekleştirilmiştir. Elde edilen eğitilmiş çıkarım sonuçları (.fis) bulanık çıkarım dosyası biçiminde simulink benzetim programında oluşturulan sistemde FLC (ANFIS) birimlerine transfer edilmiştir.

### 3.3. Otomatik Ayarlamalı Sinir Hücresi ile $PI^\lambda D^\mu$ Kontrolör Tasarımı

Güçlü, kararlı ve dayanıklı bir kontrolör tasarlamak için katsayıları otomatik ayarlanabilen uyarlamalı kontrol sistemleri ile mümkündür. Pratikte kontrol sistemi, malzeme yaşlanması, yapısal kusurlar ve çevresel faktörlere bağlı parametre belirsizlikleri gibi gerçek sistemlerin model bozulmalarının üstesinden gelmesi gerekir. Uyarlamalı bir kontrolörün, bu faktörler nedeni ile gerçek uygulamaların belirsiz ve dalgalanmalı koşulları altında sadece sistem kararlılığını değil aynı zamanda kontrol performansını muhafaza etmesi de beklenir.

Bu bölüm otomatik ayarlamalı sinir hücreleri ile uyarlamalı  $PI^\lambda D^\mu$  kontrolör geliştirmeyi amaçlamaktadır. Önerilen uyarlamalı kontrol sisteminin blok diyagramı Şekil 3.11'de gösterilmiştir. Sinir hücrelerinin çıkışları  $PI^\lambda D^\mu$  kontrolörünün kazanç katsayılarını ayarlar. Hata  $e = r - y$  şeklinde tanımlanır ve Şekil 3.11'deki otomatik ayarlamalı sinir hücresi ile beraber kontrolör katsayılarının optimizasyonunda kullanılır. Yinelemeli olarak hatanın karesinin toplamı yerel minimuma yaklaştığında, kontrol sisteminin çıkışı referans sinyale yaklaşır ve bu nedenle sinir hücrelerinin adaptasyonu tamamlanır ve optimal sistem tepkisi elde edilir.



Şekil 3.11. Otomatik Ayarlamalı Sinir Hücreleri ile  $PI^\lambda D^\mu$  kontrol sisteminin blok diyagramı

### 3.3.1. $PI^\lambda D^\mu$ Kontrollörün Ayrık Zamanda Gösterimi

$PI^\lambda D^\mu$  kontrolörün transfer fonksiyonu;

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) = u_1 + u_2 + u_3 \quad (3.37)$$

ise, Tustin ayrıklaştırma yöntemine göre, kesir dereceli türev teriminin ( $s^r, r > 0$ ) ayrık filtre uygulaması Denklem 3.38 ve 3.39'daki gibi yazılır [131, 132].

$$s = w(z^{-1}) \quad (3.38)$$

$$s^{\pm r} = (w(z^{-1}))^{\pm r} = \left( \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \right)^{\pm r} \quad (3.39)$$

ise

$$s^r = (w(z^{-1}))^r = \left( \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \right)^r = \left( \frac{2}{T} \right)^r \frac{P_p(z^{-1})}{Q_q(z^{-1})} = \left( \frac{2}{T} \right)^r \frac{\sum_{i=0}^m p_i z^{-i}}{\sum_{i=0}^n q_i z^{-i}} \quad (3.40)$$

$$s^{-r} = (w(z^{-1}))^{-r} = \left( \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \right)^{-r} = \left( \frac{2}{T} \right)^{-r} \frac{Q_q(z^{-1})}{P_p(z^{-1})} = \left( \frac{2}{T} \right)^{-r} \frac{\sum_{i=0}^n q_i z^{-i}}{\sum_{i=0}^m p_i z^{-i}} \quad (3.41)$$

olur.

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (3.42)$$

ise

$$G_c(w(z^{-1})) = K_p + K_i (w(z^{-1}))^{-\lambda} + K_d (w(z^{-1}))^\mu \quad (3.43)$$

$$G_c(z) = \frac{U(z)}{E(z)} = K_p + K_i (w(z^{-1}))^{-\lambda} + K_d (w(z^{-1}))^\mu \quad (3.44)$$

$$U(z) = K_p E(z) + K_i (w(z^{-1}))^{-\lambda} E(z) + K_d (w(z^{-1}))^\mu E(z)$$

(3.45)

$$u_1(z) = K_p E(z)$$

$$u_2(z) = K_i (w(z^{-1}))^{-\lambda} E(z) \quad (3.46)$$

$$u_3(z) = K_d (w(z^{-1}))^\mu E(z)$$

olur.

Çizelge 3.1' deki  $p = q$  için  $P_p(z^{-1}) = (1 - rz^{-1})$ ,  $k = 1$  ve  $Q_q(z^{-1}) = (1 + rz^{-1})$ ,  $k = 0$  olur.

Bu durumda  $u_1(t)$ ,  $u_2(t)$  ve  $u_3(t)$  ifadeleri ayrık zamanda aşağıdaki gibi yazılabilir.

$$u_1(z) = K_p E(z) \rightarrow u_1(k) = K_p e(k) \quad (3.47)$$

$$u_2(z) = K_i (w(z^{-1}))^{-\lambda} E(z) \rightarrow u_2(z) = K_i \left(\frac{2}{T}\right)^{-\lambda} \frac{Q_q(z^{-1})}{P_p(z^{-1})} E(z) \quad (3.48)$$

$$u_2(z) P_p(z^{-1}) = K_i \left(\frac{2}{T}\right)^{-\lambda} Q_q(z^{-1}) E(z) \quad (3.49)$$

$$u_2(z) (1 - \lambda z^{-1}) = K_i \left(\frac{2}{T}\right)^{-\lambda} (1 + \lambda z^{-1}) E(z) \quad (3.50)$$

$$u_2(k) = K_i \left(\frac{2}{T}\right)^{-\lambda} (e(k) + \lambda e(k-1)) + \lambda u_2(k-1) \quad (3.51)$$

$$u_3(z) = K_d \left(\frac{2}{T}\right)^\mu \frac{P_p(z^{-1})}{Q_q(z^{-1})} E(z) \rightarrow u_3(z) Q_q(z^{-1}) = K_d \left(\frac{2}{T}\right)^\mu \frac{P_p(z^{-1})}{Q_q(z^{-1})} E(z) \quad (3.52)$$

$$u_3(z) (1 + \mu z^{-1}) = K_d \left(\frac{2}{T}\right)^\mu (1 - \mu z^{-1}) E(z) \quad (3.53)$$

$$u_3(k) = K_d \left(\frac{2}{T}\right)^\mu (e(k) - \mu e(k-1)) - \mu u_3(k-1) \quad (3.54)$$

olur. Denklem 3.47, 3.51 ve 3.54' ü kullanarak  $PI^\lambda D^\mu$  kontrolör ayrık zamanda aşağıdaki gibi yazılabilir.

$$U(k) = K_p e(k) + K_i \left(\frac{2}{T}\right)^{-\lambda} (e(k) + \lambda e(k-1)) + \lambda u_2(k-1) + K_d \left(\frac{2}{T}\right)^\mu (e(k) - \mu e(k-1)) - \mu u_3(k-1) \quad (3.55)$$

**Çizelge 3.1.** Farklı  $r$  değerleri için  $D_r(z)$ ' in pay ve paydası [131]

	$p = q$	$P_p(z^{-1})(k = 1)$ ve $Q_q(z^{-1})(k = 0)$
1	$(-1)^k z^{-1} r + 1$	
3	$(-1)^k (r^3 - 4r) z^{-3} + (6r^2 - 9) z^{-2} + (-1)^k 15 z^{-1} r + 15$	
5	$(-1)^k (r^5 - 20r^3 + 64r) z^{-5} + (-195r^2 + 15r^4 + 225) z^{-4} + (-1)^k (105r^3 - 735r) z^{-3} + (420r^2 - 1050) z^{-2} + (-1)^k 945 z^{-1} r + 945$	

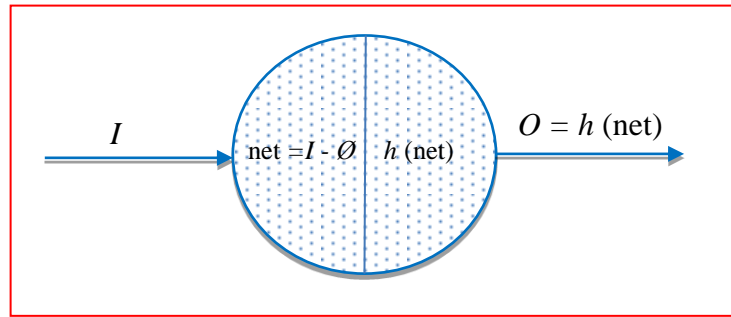


### 3.3.2. Otomatik Ayarlamalı Sinir Hücresi

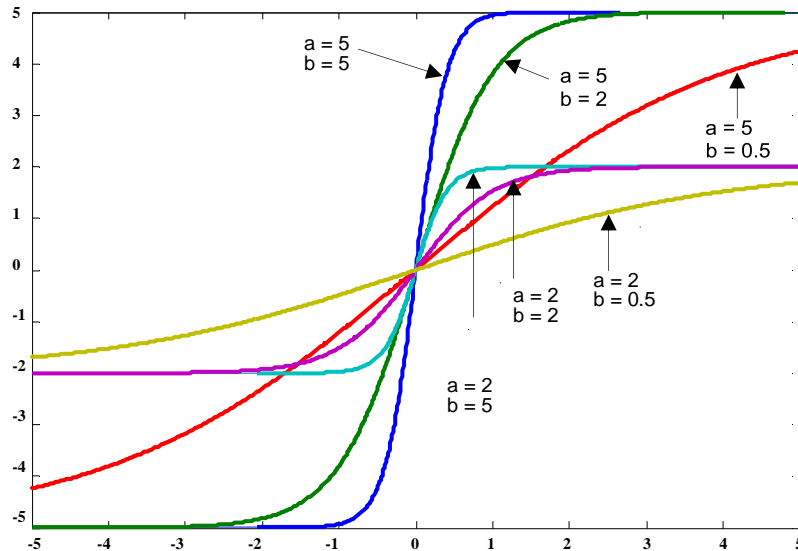
Otomatik ayarlamalı sinir hücresinin şematik yapısı Şekil 3.12'deki gibidir. Burada  $I$  sinir hücresinin girişi,  $\phi$  ayarlanabilir eşik seviyesi ve  $net = I - \phi$  ise sinir hücresinin giriş işleminden sonraki hesaplamasıdır [133]. Sinir hücresinin çıkışı;

$$O = h(net) = \frac{a * [1 - \exp(-b * net)]}{[1 + \exp(-b * net)]} \quad (3.56)$$

şeklinde verilir. Bu aktivasyon fonksiyonu,  $h(\cdot): \mathcal{R} \rightarrow \mathcal{R}$ , modifiye hiperbolik tanjant fonksiyonudur. Buradaki  $a$  fonksiyonun doyum seviyesi,  $b$  ise fonksiyonun eğimidir. Sinir hücresinin aktivasyon fonksiyonundaki çıkış aralığını ve eğrinin şeklini bu ayarlanabilir  $a$  ve  $b$  parametreleri belirler. Şekil 3.13'te verilen bu durum kontrol işlemleri sırasındaki ölçekleme ve doyum problemlerini önler. Sinir hücresine pratik uygulamalar için esnek ve uygun durum sağlar.



Şekil 3.12. Otomatik ayarlamalı sinir hücresi temel yapısı



Şekil 3.13. Farklı  $a$  ve  $b$  değerleri için hiperbolik tanjant fonksiyonu

Şekil 3.12’de ayarlanabilir parametreler vektörü  $[\phi, a, b]^T$  ise karesel hata;

$$J(k) = \frac{1}{2} [y_d(k) - y(k)]^2 \quad (3.57)$$

$$J(k) = \frac{1}{2} e^2(k) \quad (3.58)$$

olarak bulunur.

$y_d(k)$  istenen çıkış değeri,  $y(k)$  kontrol edilen sistem çıkışıdır.

$$\frac{\partial J}{\partial a_i} = \frac{\partial J}{\partial y} \frac{\partial y}{\partial u_i} \frac{\partial u_i}{\partial O_i} \frac{\partial O_i}{\partial a_i} \quad (3.59.a)$$

$$\frac{\partial J}{\partial b_i} = \frac{\partial J}{\partial y} \frac{\partial y}{\partial u_i} \frac{\partial u_i}{\partial O_i} \frac{\partial O_i}{\partial b_i} \quad (3.59.b)$$

$$\frac{\partial J}{\partial \phi_i} = \frac{\partial J}{\partial y} \frac{\partial y}{\partial u_i} \frac{\partial u_i}{\partial O_i} \frac{\partial O_i}{\partial \phi_i} \quad (3.59.c)$$

Burada  $i$  indeksi ( $i = 1$  ise  $K_p$ ,  $i = 2$  ise  $K_i$  ve  $i = 3$  ise  $K_d$  için  $PI^{\lambda}D^{\mu}$  kontrolöre bağlı sinir hücre parametrelerini ve çıkışlarını gösterir.

Denklem (3.57)’den;

$$\frac{\partial J}{\partial y} = -e$$

(3.60)

yazılır.  $\partial y / \partial u$  değerinin kontrol edilen sistemin karakterine bağlı olarak belirlenmesi gerekir. Basitleştirme ve sistemden bağımsız çözümler için, sistem modelleri  $sgn[\delta y / \delta u] = \{-1, 1\}$  biçiminde gösterilir. Bu durumda meyilli azalım optimizasyonu için, eğer sistem çıkışı, girişin artması ile artıyorsa veya girişin azalması ile azalıyorsa 1, aksi takdirde -1 alınmalıdır [133].

Denklem (3.55)’den faydalanarak Denklem (3.61) bulunur.

$$\frac{\partial u_i}{\partial O_i} = \begin{cases} e(k) & i = 1 \\ e(k) + \lambda e(k-1) & i = 2 \\ e(k) - \mu e(k-1) & i = 3 \end{cases} \quad (3.61)$$

Denklem (3.56) göz önüne alındığında, Denklem 3.62.a, 3.62.b ve 3.62.c denklemleri yazılır.

$$\frac{\partial O_i}{\partial a_i} = \frac{O_i}{a_i} \quad (3.62.a)$$

$$\frac{\partial O_i}{\partial b_i} = \frac{a_i \cdot net}{2} \left( 1 + \frac{O_i}{a_i} \right) \left( 1 - \frac{O_i}{a_i} \right) \quad (3.62.b)$$

$$\frac{\partial O_i}{\partial \phi_i} = -\frac{a_i \cdot b_i}{2} \left( 1 + \frac{O_i}{a_i} \right) \left( 1 - \frac{O_i}{a_i} \right) \quad (3.62.c)$$

Denklem (3.59) – (3.62) dikkate alınarak, eşik seviyesi ( $\emptyset$ ), doyma seviyesi ( $a$ ) ve eğim ( $b$ ) parametreleri Denklem (3.63.a, 3.63.b, 3.63.c)'deki gibi elde edilir.

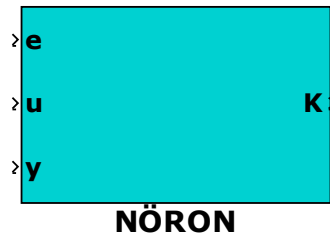
$$a_i(k+1) = a_i(k) + \eta_i \cdot e \cdot \text{sgn} \left[ \frac{\partial y}{\partial u_i} \right] \cdot \frac{\partial u_i}{\partial O_i} \cdot \frac{O_i}{a_i} \quad (3.63.a)$$

$$b_i(k+1) = b_i(k) + \eta_i \cdot e \cdot \text{sgn} \left[ \frac{\partial y}{\partial u_i} \right] \cdot \frac{\partial u_i}{\partial O_i} \cdot \frac{a_i \cdot net_i}{2} \left( 1 + \frac{O_i}{a_i} \right) \left( 1 - \frac{O_i}{a_i} \right) \quad (3.63.b)$$

$$\phi_i(k+1) = \phi_i(k) - \eta_i \cdot e \cdot \text{sgn} \left[ \frac{\partial y}{\partial u_i} \right] \cdot \frac{\partial u_i}{\partial O_i} \cdot \frac{a_i \cdot b_i}{2} \left( 1 + \frac{O_i}{a_i} \right) \left( 1 - \frac{O_i}{a_i} \right) \quad (3.63.c)$$

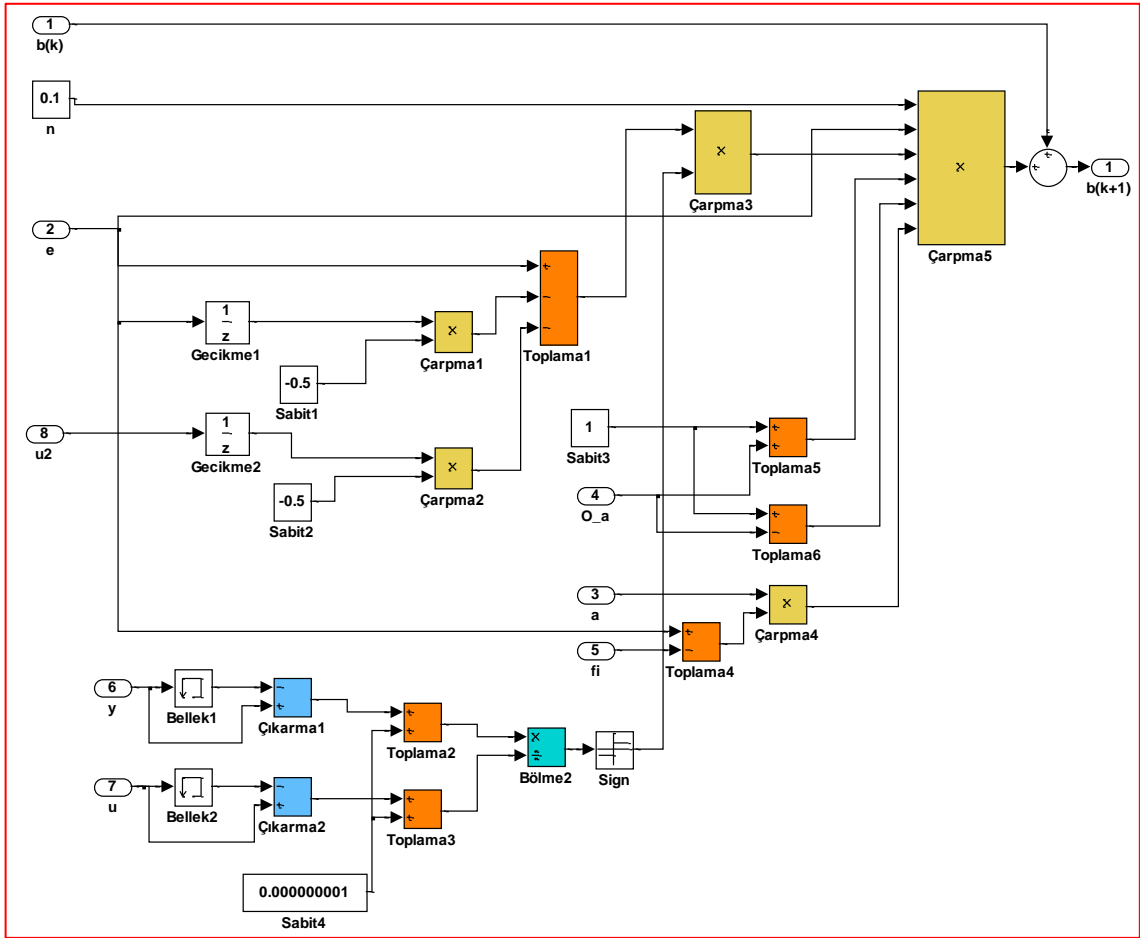
PI<sup>λ</sup>D<sup>μ</sup> kontrolörün katsayıları ve kesirli dereceleri, ZN yöntemine göre belirlenir ve kontrolörün optimizasyon esnasında karmaşıklığını ve sistem kararsızlığı riskini azaltmak için kesir dereceleri sabit tutulur.

Denklem (3.63.a, 3.63.b, 3.63.c)'e göre MATLAB–Simulink ortamında  $K_p$  için Otomatik Ayarlamalı Sinir Hücre Bloğu Şekil 3.14' de gösterilmiştir.

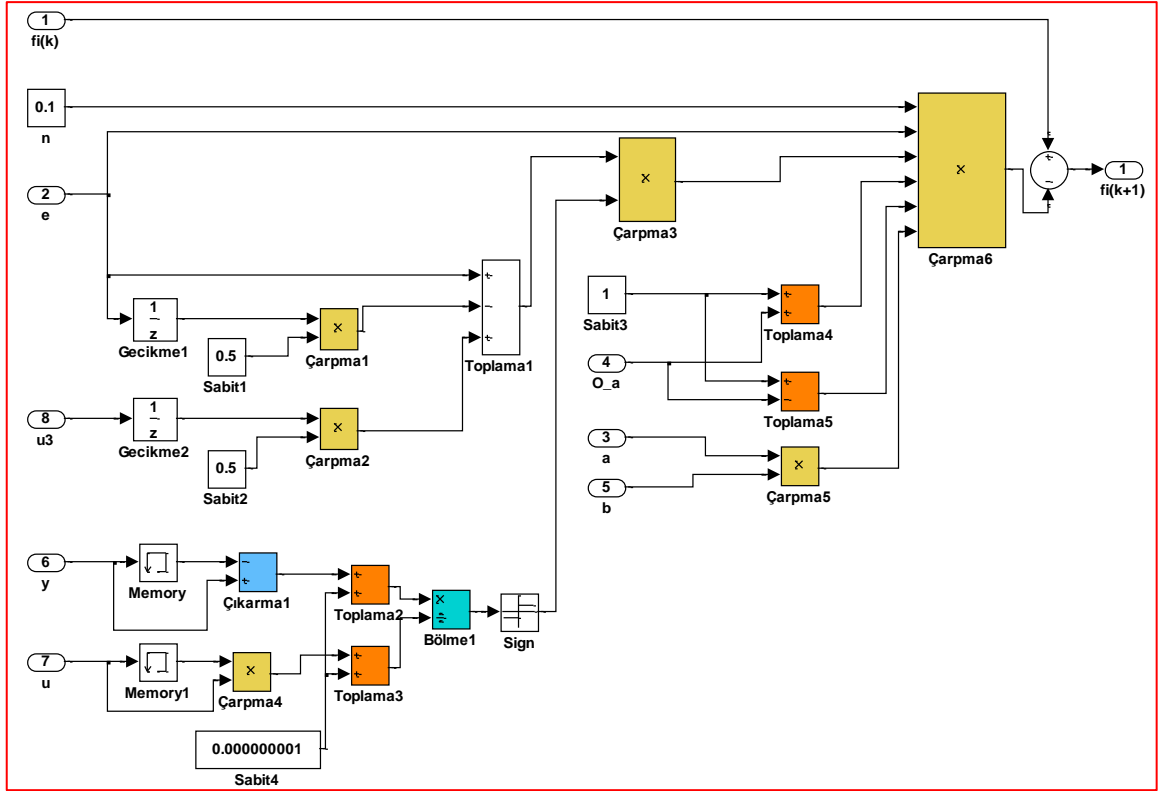


Şekil 3.14. Otomatik Ayarlamalı Sinir Hücresi





Şekil 3.17. Denklem (3.63.b)'ye göre  $b$  parametresinin MATLAB – Simulink gösterimi



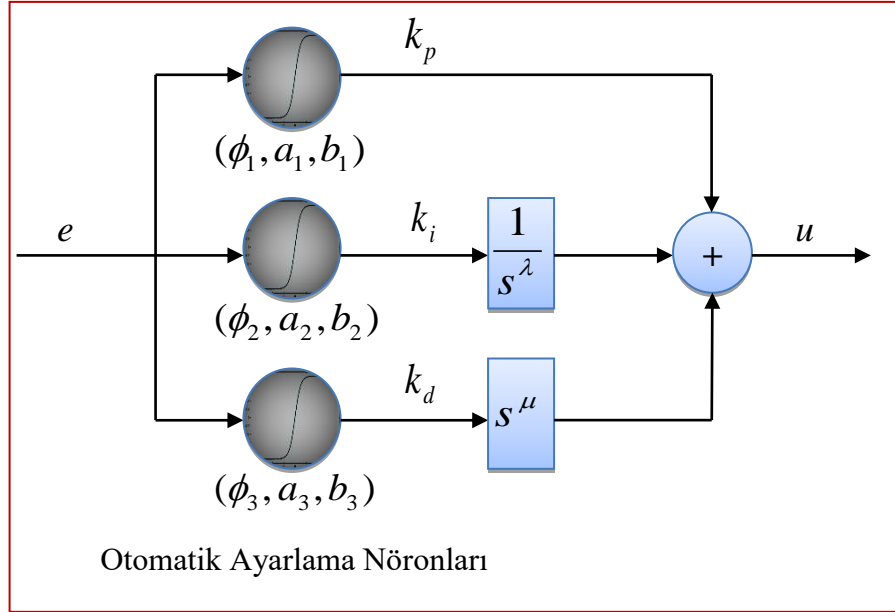
**Şekil 3.18.** Denklem (3.63.c)'ye göre  $\phi$  parametresinin MATLAB– Simulink gösterimi

Otomatik ayarlanmalı sinir hücresi eşik seviyesi ( $\phi$ ), doyma seviyesi ( $a$ ) ve eğim ( $b$ ) aktivasyon fonksiyonunun parametreleri optimize ederek eğitilir. Doğrusal olmayan aktivasyon fonksiyonunun şekli ise  $a$  ve  $b$  değerleri ile kontrol edilir. Kontrol sisteminin karesel hatasının toplamını en aza indirmek için eğimli azalım (gradient descent) optimizasyon metodu kullanılarak gerçekleştirilir ( $J = \int e^2 dt$ ).

$$J_{opt} = \min_{\{\phi, a, b\}} (J) \quad (4.74)$$

Önerilen yöntemin sistem modelinden bağımsız olmasını sağlamak için, sistem çıkışının sistem girişine bağımlılığı kesinlikle artan veya kesinlikle azalan fonksiyonlar olarak kabul edilir. Bu nedenle sistem modeli optimizasyon sürecinde  $\text{sgn}[\partial y / \partial u] = \{-1, 1\}$  şeklinde temsil edilir. Bu da eğer sistemin çıkışı, giriş ile artarsa pozitif eğimlidir ve +1 değerini verir. Aksi halde negatif eğimlidir ve optimizasyon işlemi için -1 değerini verir.

Bu otomatik ayarlamalı sinir hücresi kullanılarak kontrolörlerin istenilen parametreleri hiçbir müdahale veya ayarlamaya gereksinim olmadan ayarlanabilir. Bu teori doğrultusunda P,  $PI^\lambda$ ,  $PD^\mu$  ve  $PI^\lambda D^\mu$  kontrolör uygulamaları gerçekleştirilmiştir [134].



**Şekil 3.19.** Otomatik ayarlamalı sinir hücreleri ile  $PI^\lambda D^\mu$  kontrolör tasarımı

Şekil 3.11'de gösterilen tek girişli-tek çıkışlı bir sistemde,  $G_p$  tamsayı veya kesir dereceli sistemin transfer fonksiyonunu temsil etmektedir.  $G_c$  ise otomatik ayarlamalı sinir hücreleri ile ayarlanacak olan  $PI^\lambda D^\mu$  kontrolördür. Bu yöntemde hedef, sistemi kararlı hale getiren  $PI^\lambda D^\mu$  kontrolörün  $K_p$ ,  $K_i$  ve  $K_d$  katsayılarını minimum karesel hata ( $J_{opt}$ ) ile çevrimiçi ayarlanmasıdır.

#### 4. ARAŞTIRMA BULGULARI

Bu bölümde model referans uyarlamalı kontrol, ANFIS ve otomatik ayarlamalı sinir hücresi ile tasarlanan kesir dereceli PID kontrolörlerle yapılan benzetim uygulamalarından elde edilen sonuçlar verilmiştir ve bu sonuçların fuzzy ve PID kontrolör sonuçları ile karşılaştırılması yapılmıştır.

##### 4.1. Model Referans Uyarlamalı Kesir Dereceli PID (MRA-PI<sup>λ</sup>D<sup>μ</sup>) Kontrolör Tasarım Uygulamaları

Elde edilen MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolör iki farklı sistem üzerinden denenmiştir. İkinci ve üçüncü dereceden iki farklı sisteme uygulanan bu çalışmanın sonuçları tam dereceli MRA-PID kontrolörün sonuçları ile karşılaştırılmıştır. Benzetim programında kullanılan kesir dereceli türev hesaplamalarında CRONE yaklaşımı kullanılmıştır.

##### Örnek 1.

Bu örnekte sistemin transfer fonksiyonu Denklem (4.1)'deki gibi alınmıştır. S cevabına dayalı ayarlama kuralları kullanarak PID ve PI<sup>λ</sup>D<sup>μ</sup> kontrolörlerinin transfer fonksiyonları sırasıyla Denklem (4.2) ve Denklem (4.3)'deki gibi bulunmuştur [70].

$$\frac{N(s)}{M(s)} = \frac{K}{4.32s^2 + 19.1801s + 1} \quad (4.1)$$

$$G(s) = 120 + \frac{300}{s} + 12s \quad (4.2)$$

$$G(s) = 6.9928 + \frac{12.4044}{s^{0.6}} + 4.1066s^{0.7805} \quad (4.3)$$

Denklem (3.13), (4.1) ve (4.3) kullanılarak MRA-PI<sup>λ</sup>D<sup>μ</sup> için model transfer fonksiyonu Denklem (4.4)'deki gibi belirlenir.

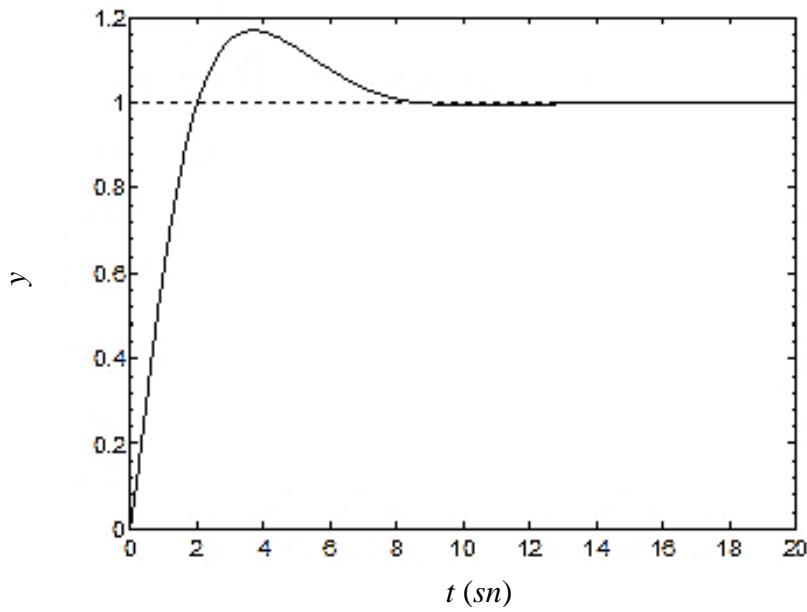
$$\frac{Z_m(s)}{D_m(s)} = \frac{6.9928s^{0.6} + 4.1066s^{1.3805} + 12.4044}{4.32s^{2.6} + 19.1801s^{1.6} + 4.1066s^{1.3805} + 7.9928s^{0.6} + 12.4044} \quad (4.4)$$



Oustaloup filtresini ve yüksek derece yaklaşımı kullanılarak Denklem (4.4), Denklem (4.5)'deki gibi tam dereceli transfer fonksiyonuna indirgenebilir [135, 136].

$$\frac{Z_m(s)}{D_m(s)} = \frac{12s^2 + 120s + 300}{4.32s^3 + 31.1801s^2 + 121s + 300} \quad (4.5)$$

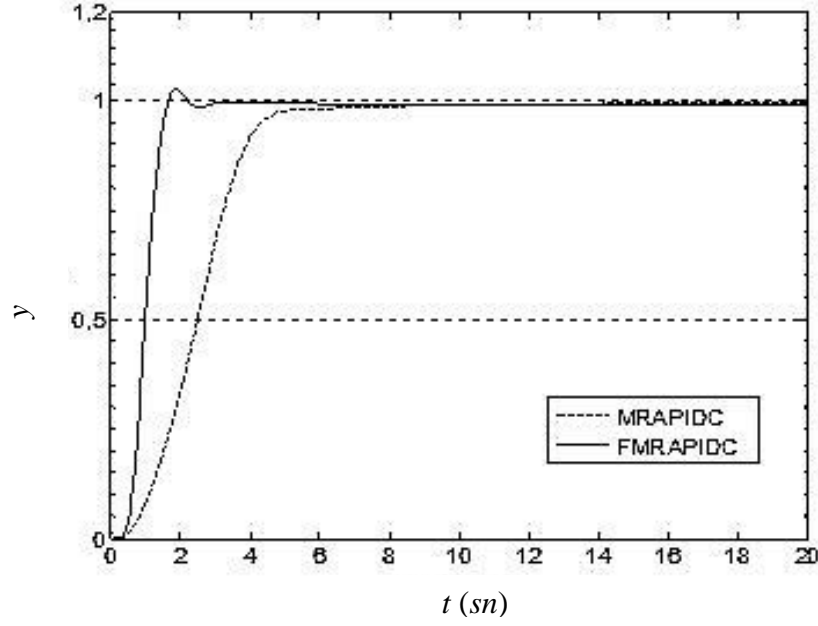
Denklem (4.3)'deki  $PI^\lambda D^\mu$  kontrolör ve Denklem (4.1)'deki sistem transfer fonksiyonunda  $K = 1$  alınarak sadece  $PI^\lambda D^\mu$  kontrolör için birim basamak çıkış cevabı Şekil 4.1'de gösterilmiştir.



Şekil 4.1.  $PI^\lambda D^\mu$  kontrolör için sistemin birim basamak cevabı

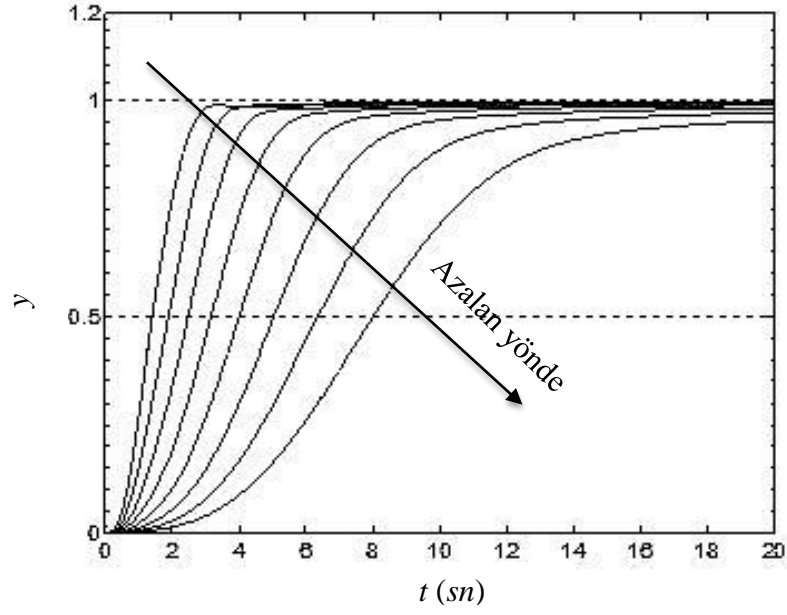
Klasik MRA-PID kontrolör için Denklem (3.22), (3.23) ve (3.24)'de  $\lambda = \mu = 1$  alındıktan sonra Denklem (4.1) ve Denklem (4.2) kullanılarak Şekil 3.4'deki sistem girişine birim basamak işareti uygulanarak sistem çıkış cevabı Şekil 4.2'deki gibi bulunmuştur. MRA-PID kontrolör ile sistemden en iyi cevabı almak için, parametreler sırasıyla  $\alpha = 2000$ ,  $\beta = 15$  ve  $\gamma = 1000$  olarak belirlenmiştir. Benzer şekilde, Denklem (4.1) ve Denklem (4.3)'deki değerlere göre MRA- $PI^\lambda D^\mu$  kontrolör için Denklem (3.22), (3.23) ve (3.24) kullanarak Şekil 3.4'deki sistem girişine birim basamak uygulanarak sistem çıkış cevabı Şekil 4.2'deki gibi klasik MRA-PID kontrolör çıkış cevabı ile birlikte çizilmiştir. MRA- $PI^\lambda D^\mu$  kontrolör ile en iyi cevabı almak için parametreler  $\alpha =$

100,  $\beta = 500$  ve  $\gamma = 600$  olarak belirlenmiş ve her iki kontrolör için yapılan uygulamada sistem transfer fonksiyon kazancı  $K = 1$  olarak alınmıştır.

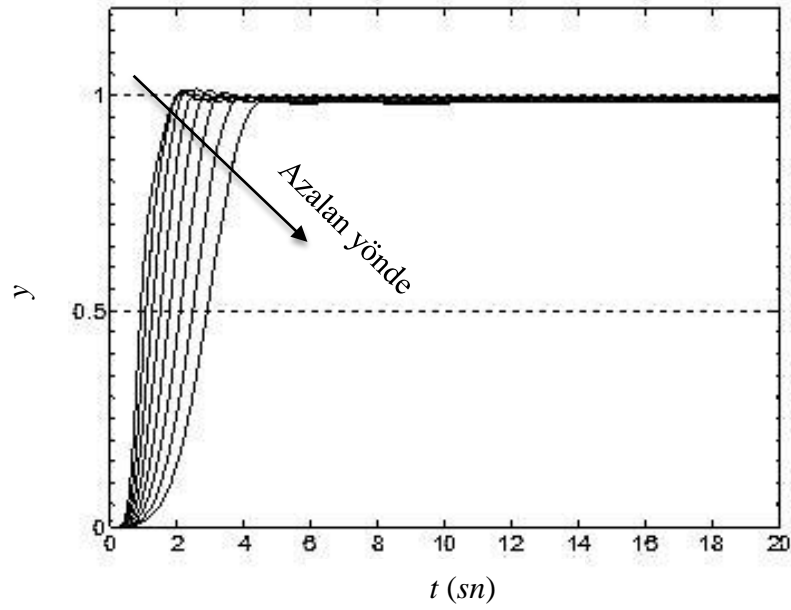


**Şekil 4.2.** MRA-PID ve MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolörler için birim basamak çıkış cevapları

Şekil 4.2'de MRA-PID ve MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolörlerinin sistem çıkış cevaplarından MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolör cevabının çok kısa sürede, yaklaşık 3 sn, istenilen konuma geldiği fakat MRA-PID kontrolörde ise sistem ancak 9 sn'de istenen konuma ulaşmıştır. MRA-PID ve MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolörlerin dayanıklılığını karşılaştırmak için sistemin kazanç parametresi sırayla  $K = 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32$  olarak azaltılarak her iki denetleyici için birim basamak çıkış cevapları Şekil 4.3 ve Şekil 4.4'de gösterilmiştir.



**Şekil 4.3.** Azalan  $K$  değerleri için MRA-PID kontrolörün sistem birim basamak çıkış cevapları



**Şekil 4.4.** Azalan  $K$  değerleri için MRA-PI $^{\lambda}$ D $^{\mu}$  kontrolörün sistem birim basamak çıkış cevapları

Azalan  $K$  değerleri grafiklerinden, Şekil 4.3 ve Şekil 4.4, MRA-PID kontrolör ile 4–20 sn’leri ve üzeri bir zaman aralığında sistem istenen konuma ulaşırken, MRA-PI $^{\lambda}$ D $^{\mu}$  kontrolör ile 2-6 sn’leri arasında gerçekleşmektedir. Sonuç olarak MRA-PI $^{\lambda}$ D $^{\mu}$  ile gerçekleştirilen kontrol uygulamasının çok daha hızlı olduğu görülmüştür.

## Örnek 2.

Sistemin transfer fonksiyonu Denklem (4.6)'daki gibi alınmıştır. Bu sisteme kritik kazanca dayalı ayarlama kuralı uygulanarak PID ve  $PI^\lambda D^\mu$  kontrolörlerin katsayıları alınmış ve fonksiyonları Denklem (4.7) ve Denklem (4.8)'de verilmiştir [70].

$$\frac{N(s)}{M(s)} = \frac{K}{s^3 + 2.539 s^2 + 62.15 s} \quad (4.6)$$

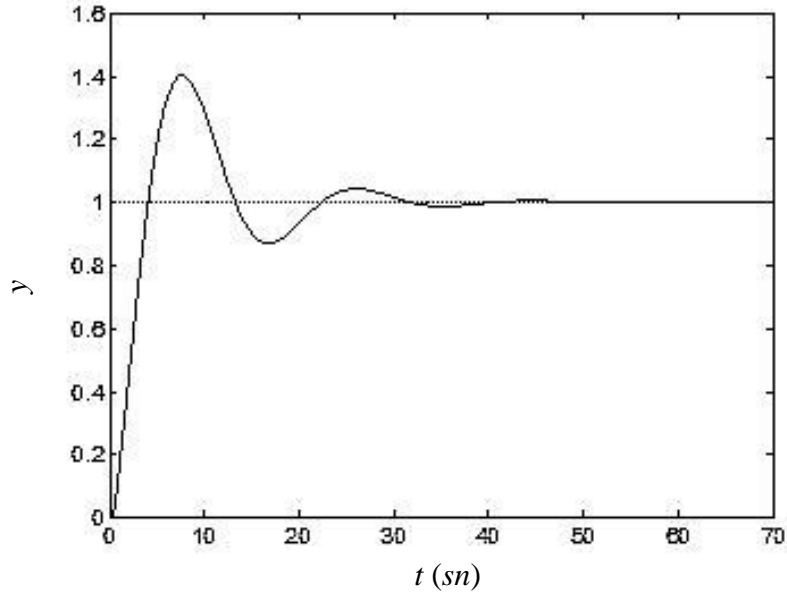
$$G(s) = 94.68 + \frac{237.591}{s} + 9.4325 s \quad (4.7)$$

$$G(s) = 0.8271 + \frac{14.3683}{s^{0.5588}} - 1.6866 s^{1.2328} \quad (4.8)$$

$$\frac{Z_m(s)}{D_m(s)} = \frac{9.4325 s^2 + 94.68 s + 237.591}{s^4 + 2.539 s^3 + 71.5825 s^2 + 94.68 s + 237.591} \quad (4.9)$$

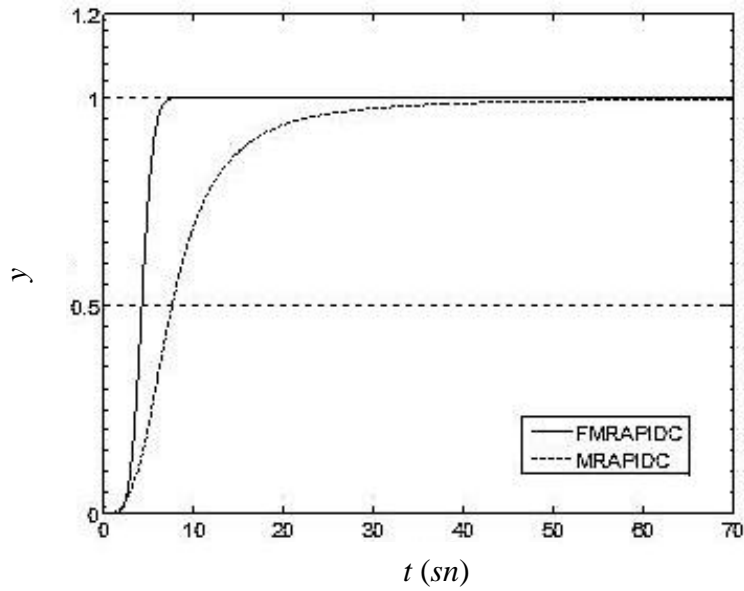
$$\frac{Z_m(s)}{D_m(s)} = \frac{-1.6866 s^{1.7916} + 0.8271 s^{0.5588} + 14.3683}{s^{3.5588} + 2.539 s^{2.5588} - 1.6866 s^{1.7916} + 62.15 s^{1.5588} + 0.8271 s^{0.5588} + 14.3683} \quad (4.10)$$

Önceki örnekte olduğu gibi, Denklem 4.6 ve 4.7 kullanarak MRA-PID için model transfer fonksiyonu Denklem 4.9'da ve Denklem 4.8 ve 4.10 kullanarak MRA-  $PI^\lambda D^\mu$  için model transfer fonksiyonu Denklem 4.10'da verilmiştir. Sistem transfer fonksiyonunda  $K = 1$  olarak alınmış ve  $PI^\lambda D^\mu$  kontrolörün birim basamak cevabı Şekil 4.5'de gösterilmiştir.



Şekil 4.5.  $PI^{\lambda}D^{\mu}$  kontrolör için sistemin birim basamak cevabı

MRA-PID kontrolör için  $\alpha = 100$ ,  $\beta = 500$  ve  $\gamma = 600$  ve MRA- $PI^{\lambda}D^{\mu}$  kontrolör içinde  $\alpha = 10$ ,  $\beta = 390$  ve  $\gamma = 10$  alınarak iki kontrol sisteminin birim basamak çıkış cevapları karşılaştırmalı olarak Şekil 4.6'da verilmiştir.

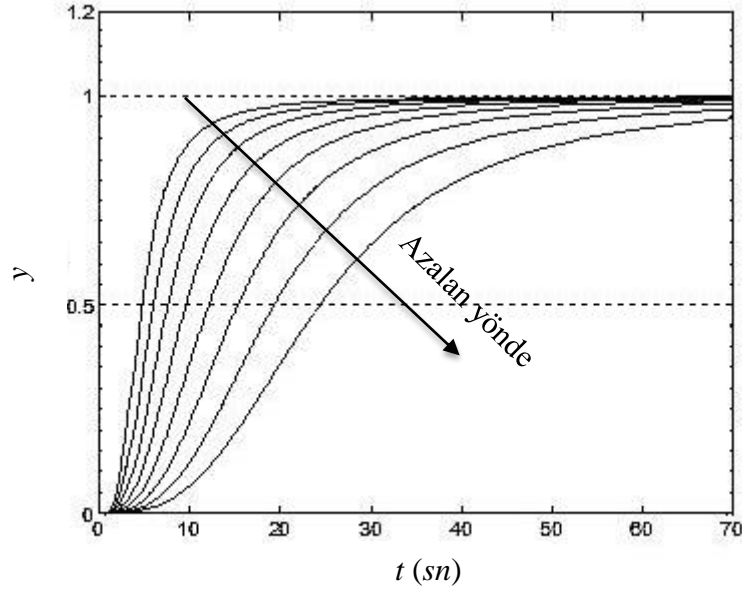


Şekil 4.6. MRA-PID ve MRA- $PI^{\lambda}D^{\mu}$  kontrolörler için birim basamak çıkış cevapları ( $K = 1$ )

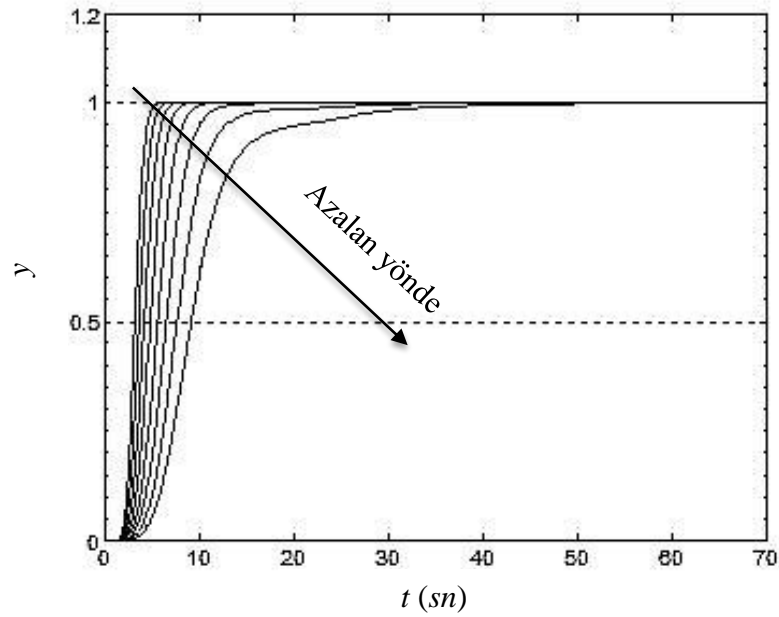
Şekil 4.6' da MRA-PID ve MRA- $PI^{\lambda}D^{\mu}$  kontrolörlerinin sistem çıkış cevapları karşılaştırıldığında MRA- $PI^{\lambda}D^{\mu}$  kontrolör ile sistem 8 sn. de istenilen değere ulaşırken

MRA-PID kontrolör ise yaklaşık 60 sn. de istenilen değere ulaşmıştır. Buradan MRA- $PI^{\lambda}D^{\mu}$  kontrolörün daha hızlı cevap verdiği anlaşılmaktadır.

Her iki kontrol sisteminin dayanıklılığını karşılaştırmak için, sistem kazanç ( $K$ ) parametresi azaltılarak sırasıyla  $K = 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32$  olarak alınmış ve MRA-PID ve MRA- $PI^{\lambda}D^{\mu}$  kontrolörlerin sistem birim basamak cevapları sırasıyla Şekil 4.7' de ve Şekil 4.8'de gösterilmiştir.



Şekil 4.7. Azalan  $K$  değerleri için MRA-PID kontrolörün sistem birim basamak çıkış cevapları



Şekil 4.8. Azalan  $K$  değerleri için MRA- $PI^{\lambda}D^{\mu}$  kontrolörün sistem birim basamak çıkış cevapları

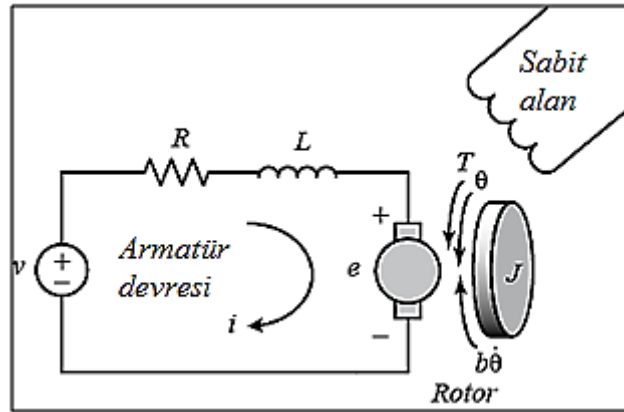
Azalan  $K$  değerleri grafiklerinden, Şekil 4.7 ve Şekil 4.8, MRA-PID kontrolör ile 25sn ve üzerindeki zamanlarda sistem istenen konuma ulaşırken, MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolör de ise bu durum 5. saniyeden sonra gerçekleşmektedir.

#### 4.2. Uyarlamalı Sinirsel Bulanık Çıkarım Sistemi (ANFIS) ile Kesir Dereceli PID (ANFIS-PI<sup>λ</sup>D<sup>μ</sup>) Kontrolör Tasarım Uygulamaları

Bu bölümde önerilen yöntemin performansı aşağıdaki iki örnekle gösterilmiş ve elde edilen sonuçlar tartışılmıştır.

**Örnek 1.** Bu örnekte hız kontrolü gerçekleştirilecek olan bir DC motorun transfer fonksiyonu Denklem 4.11'de verilmiştir [137].

$$G_p(s) = \frac{K}{(Js + b)(Ls + R) + K^2} \quad (4.11)$$



Şekil 4.9. DC motorun eşdeğer devresi

Şekil 4.9'da motorun armatürüne  $V$  gerilimi uygulanmasına karşılık şaftın konumu  $\theta$  ve dönme hızı  $d\theta/dt$ 'dir. Ayrıca şaft açısız hızıyla orantılı bir sürtünme torkunun olduğu varsayılır.

Bu örnek için fiziksel parametreler:

$(J)$	rotorun atalet momenti	→	0.01 kg.m <sup>2</sup>
$(b)$	motor viskoz sürtünme sabiti	→	0.1 N.m.s
$(K_e)$	elektromotor kuvvet sabiti	→	0.01 V/rad/s
$(K_t)$	motor tork sabiti	→	0.01 N.m/Amp
$(R)$	armatür direnci	→	1 Ohm
$(L)$	armatür indüktansı	→	0.5 H

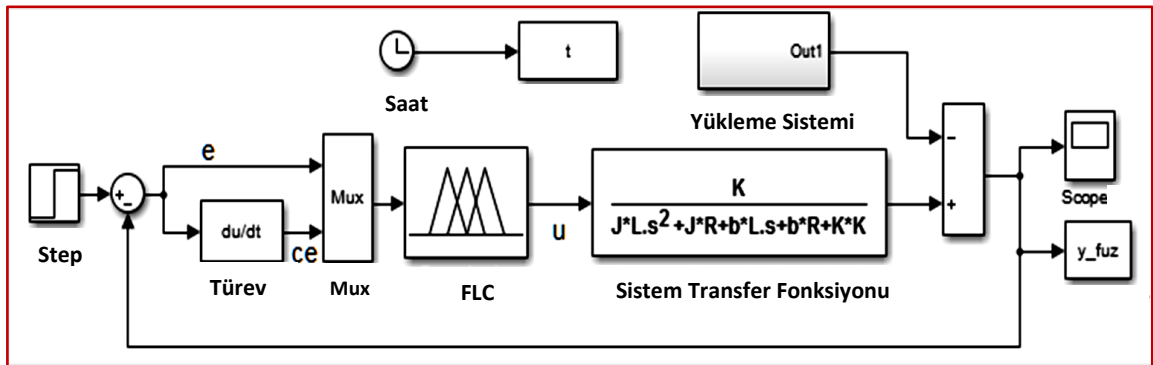
Bu parametreler Denklem 4.11’de yerine konulduğunda motorun açık çevrim transfer fonksiyonu

$$G_p(s) = \frac{0.01}{0.005s^2 + 0.06s + 0.1001} \quad (4.12)$$

gibi olur. Genetik algoritma kullanarak bu sistem için  $PI^\lambda D^\mu$  kontrolörün başlangıçtaki katsayıları aşağıdaki gibi bulunur [28].

$$PI^\lambda D^\mu: K_p = 110.560, K_i = 212.541, K_d = 5.752, \lambda = -0.9103, \mu = 0.8706 \quad (4.13)$$

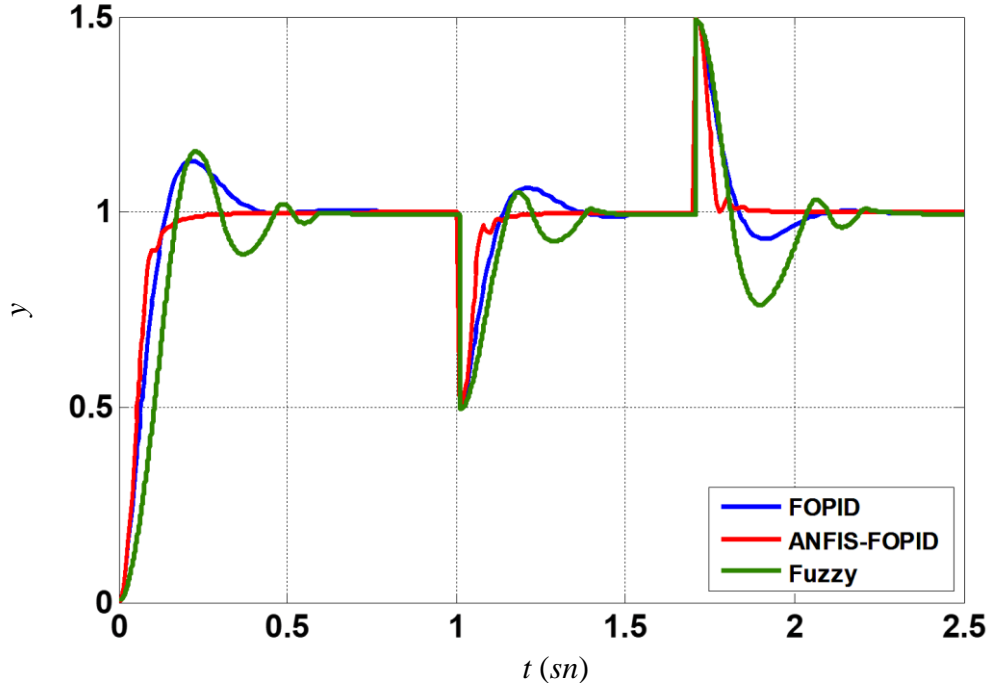
Şekil 3.10’da blok diyagramı gösterilen ve Denklem 4.13’deki katsayılarla sahip  $PI^\lambda D^\mu$  kontrolör sistemi ANFIS eğitiminde verileri almak için kullanılmıştır. Eğitim sürecinden sonra Şekil 3.9’da verildiği gibi  $FLC_1$  (ANFIS $_{K_p}$ ),  $FLC_2$  (ANFIS $_{K_i}$ ) ve  $FLC_3$  (ANFIS $_{K_d}$ ) için bulanık mantık kuralları ve üyelik fonksiyonları veritabanı oluşturulmuştur. ANFIS yapılarında, Gauss üyelik fonksiyonları ve beş tane kural kullanılmıştır. Şekil 4.10’da, önerilen sistem ile karşılaştırmak için kullanılan FLC kontrol sistemi blok diyagramı gösterilmiştir. Hata ( $e$ ) ve hatanın değişimi ( $ce$ ) FLC girişlerine ve çıkışta alınan  $u$  kontrol sinyali DC motora giriş işareti olarak uygulanmıştır. Üyelik fonksiyonlarını, kuralları ve çıkarım yöntemini içeren bulanık çıkarım dosyası (.fis) MATLAB fuzzy toolbox ile FLC ünitesine yüklenmiştir [138].



Şekil 4.10. FLC ile DC motor hız kontrol blok diyagramı [140]

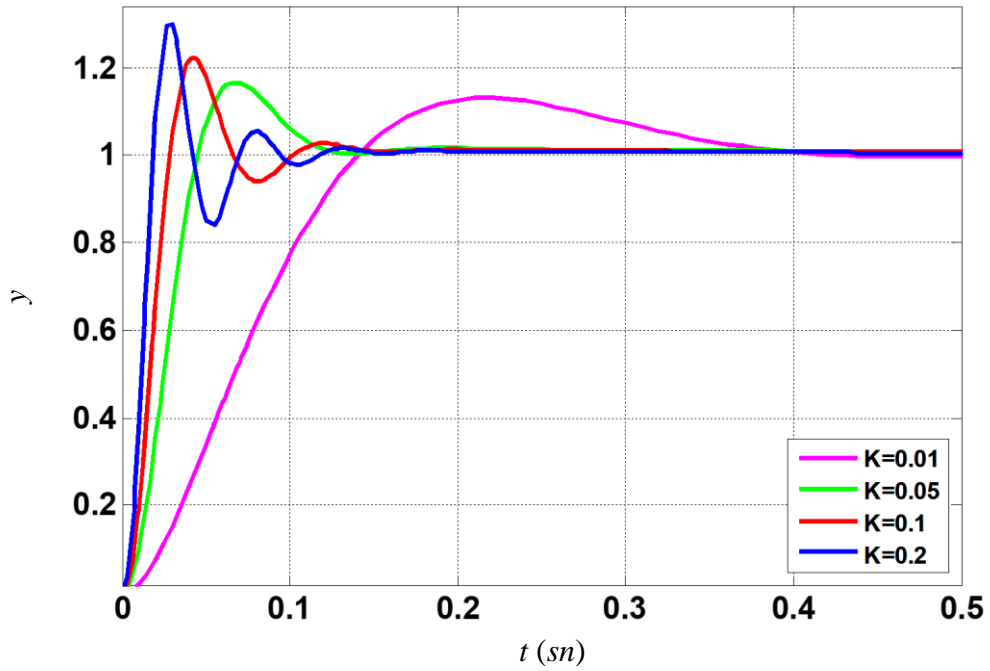
Daha sonra  $K$ ,  $J$ ,  $L$  ve  $b'$  nin çeşitli değerleri için  $PI^\lambda D^\mu$ , FLC ve ANFIS- $PI^\lambda D^\mu$  kontrolörlerinin birim basamak cevapları Şekil 4.11 - 4.23’a kadar olan grafiklerde gösterilmiş ve analizleri yapılmıştır. (FOPID:  $PI^\lambda D^\mu$ , ANFIS-FOPID: ANFIS- $PI^\lambda D^\mu$ , Fuzzy Lojik Kontrolör : FLC).



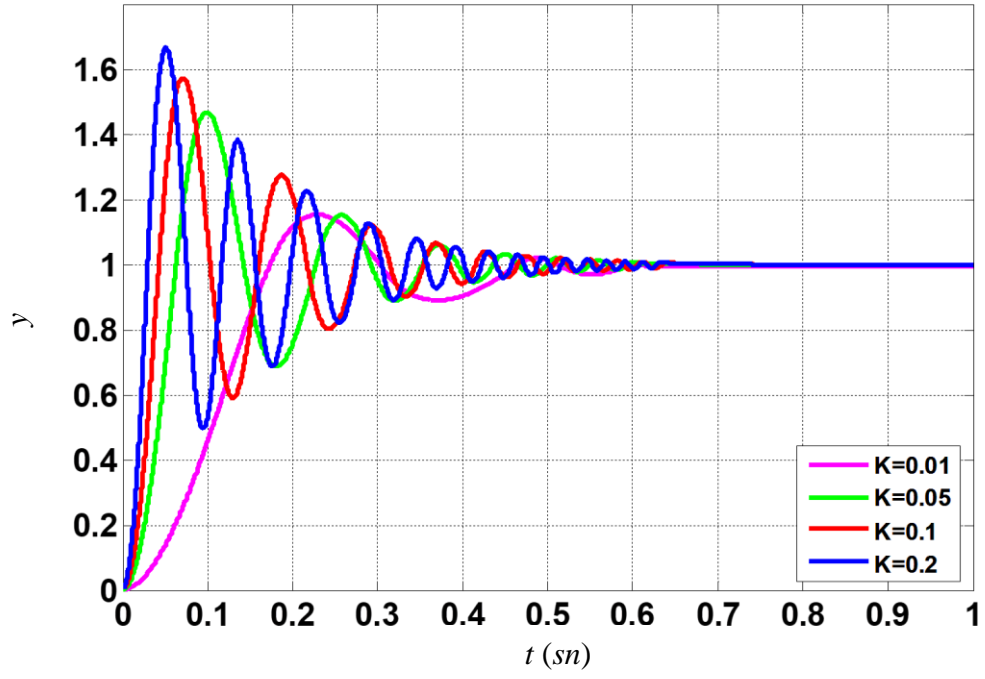


Şekil 4.11.  $K (0.01)$ ,  $J (0.01)$ ,  $L (0.5)$  ve  $b (0.1)$  için  $PI^\lambda D^\mu$ , FLC ve ANFIS- $PI^\lambda D^\mu$  kontrolör sistemlerinin birim basamak cevapları

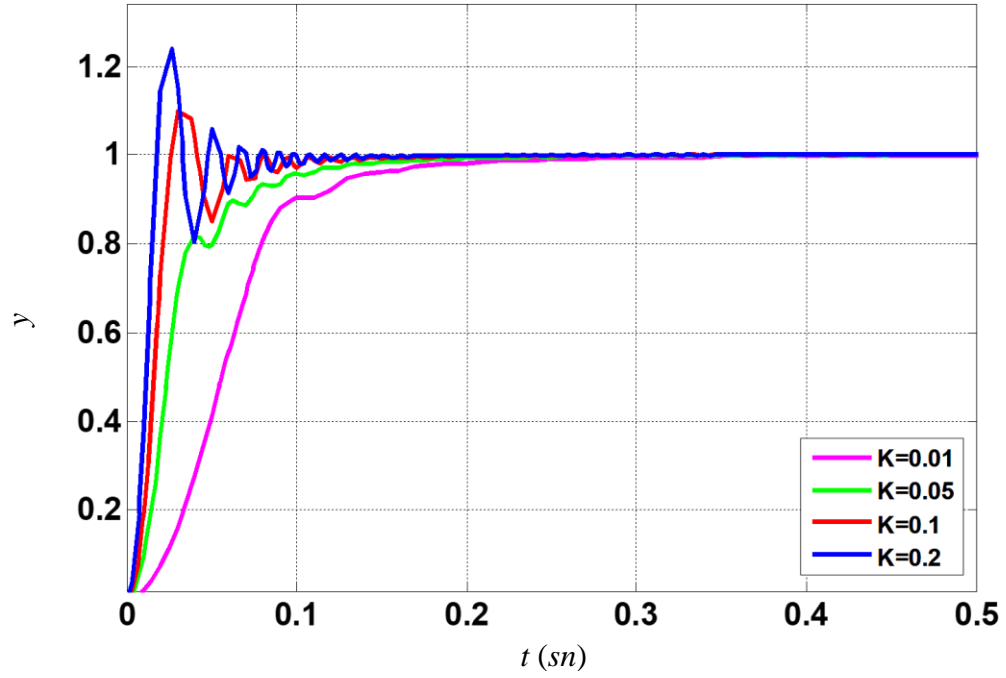
Şekil 4.11' de gösterildiği gibi,  $PI^\lambda D^\mu$ , FLC ve ANFIS- $PI^\lambda D^\mu$  denetleyicilerinin karşılaştırmalı birim basamak cevaplarından ANFIS- $PI^\lambda D^\mu$  kontrolörün daha az sönümlü, daha az salınımlı ve daha hızlı tepki verdiği görülmektedir. Çevrim içi ayarlama yapabilen önceden eğitilmiş ANFIS'ler sayesinde bu sonuçlar alınmıştır.



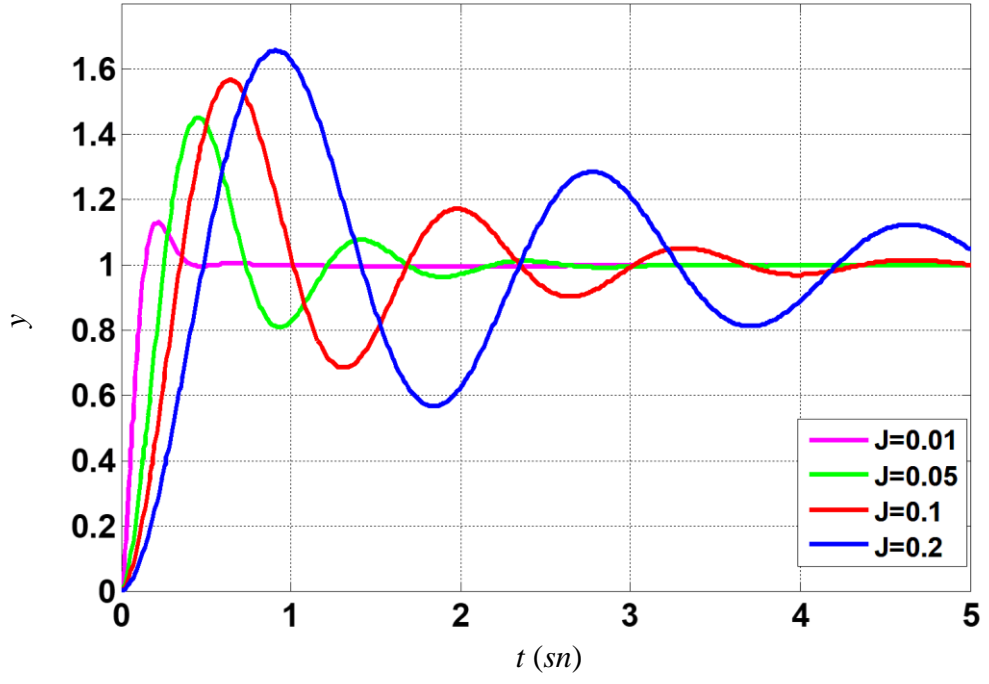
Şekil 4.12. Artan  $K$  değerleri için  $PI^\lambda D^\mu$  kontrolör birim basamak cevapları



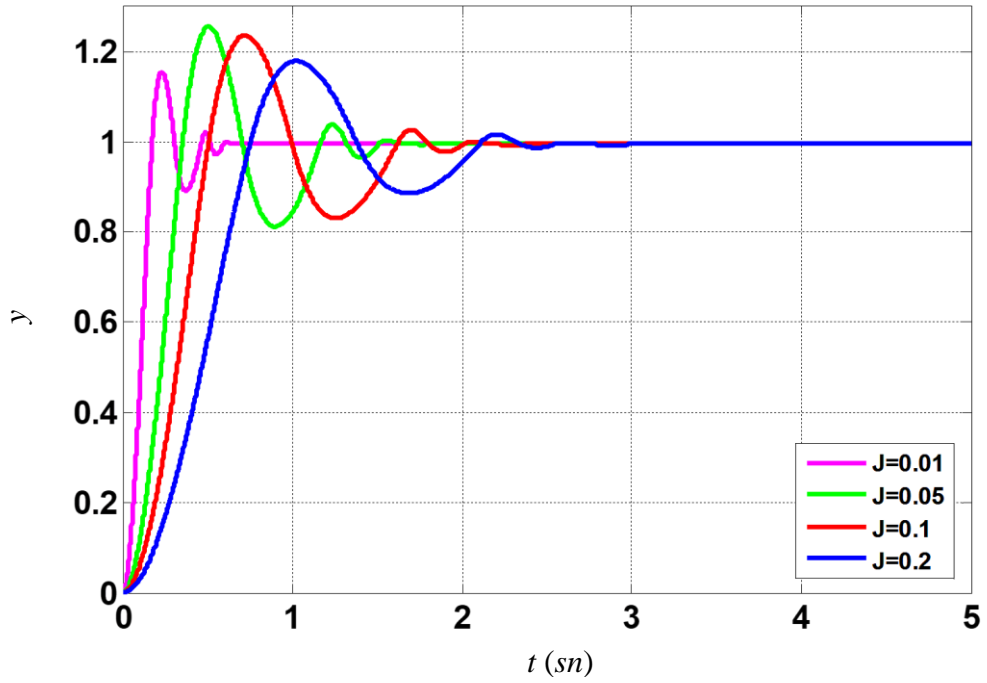
Şekil 4.13. Artan  $K$  değerleri için FLC birim basamak cevapları



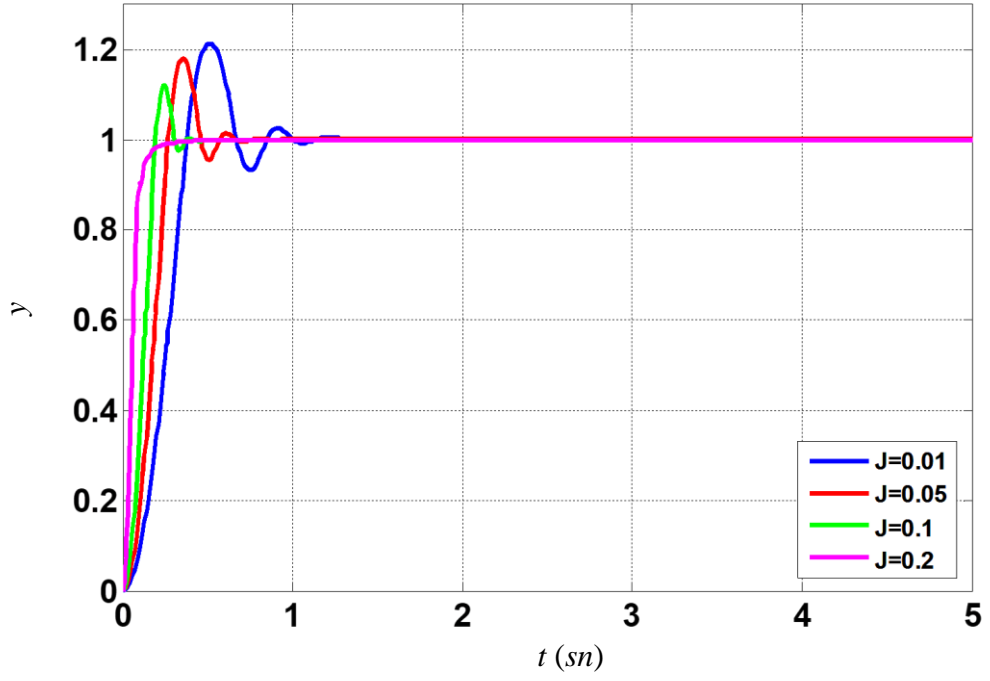
Şekil 4.14. Artan  $K$  değerleri için ANFIS-PI<sup>3</sup>D<sup>m</sup> kontrolör birim basamak cevapları



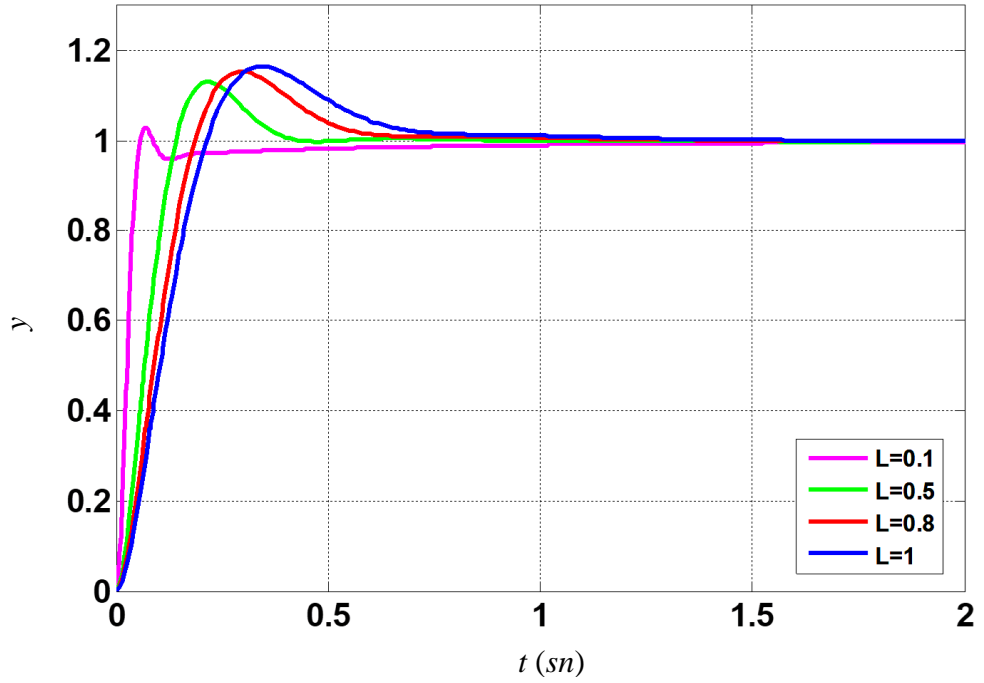
Şekil 4.15. Artan  $J$  değerleri için  $PI^\lambda D^\mu$  kontrolör birim basamak cevapları



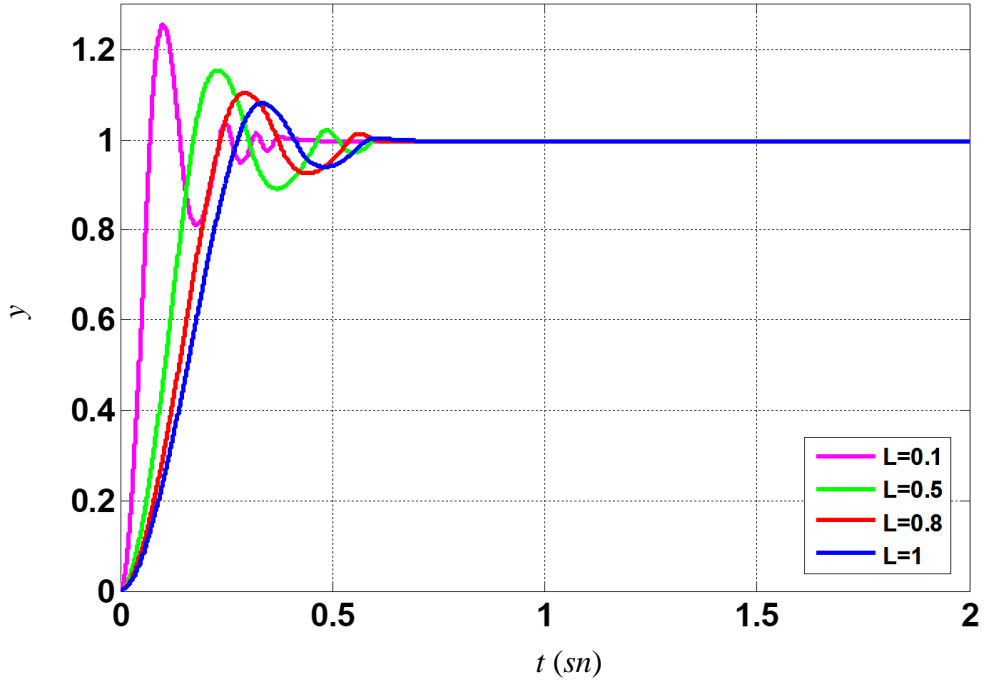
Şekil 4.16. Artan  $J$  değerleri için FLC birim basamak cevapları



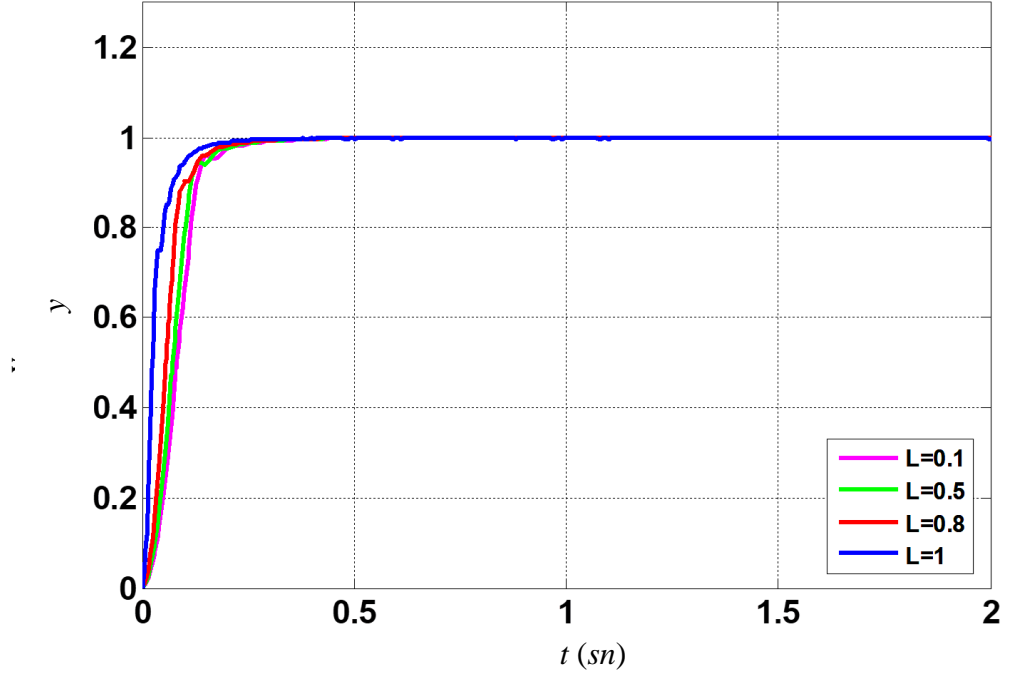
Şekil 4.17. Artan  $J$  değerleri için ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolör birim basamak cevapları



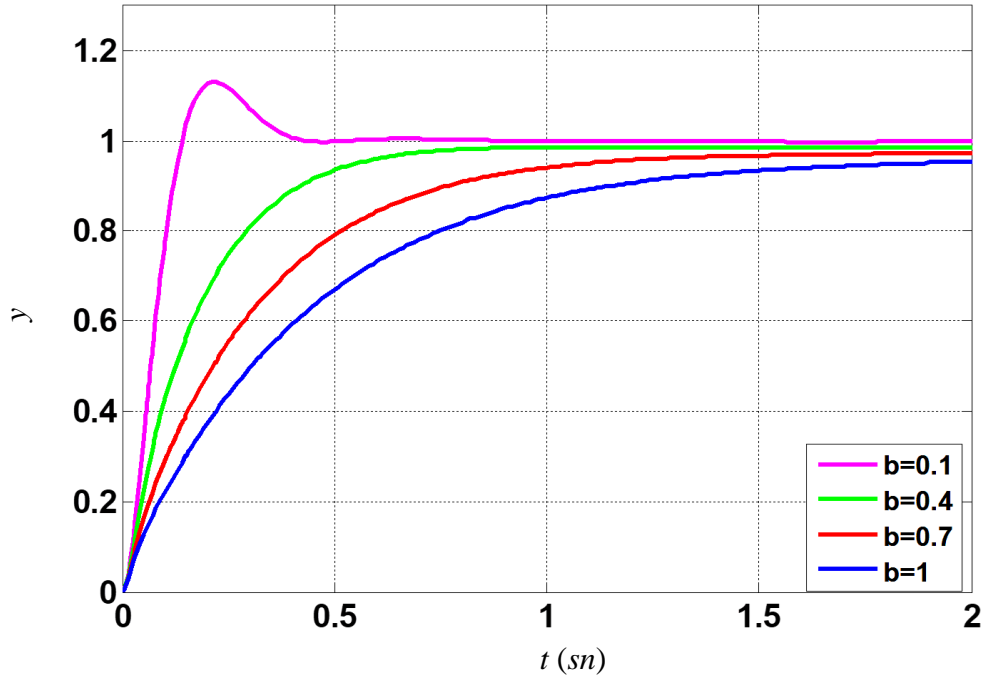
Şekil 4.18. Artan  $L$  değerleri için PI<sup>λ</sup>D<sup>μ</sup> kontrolör birim basamak cevapları



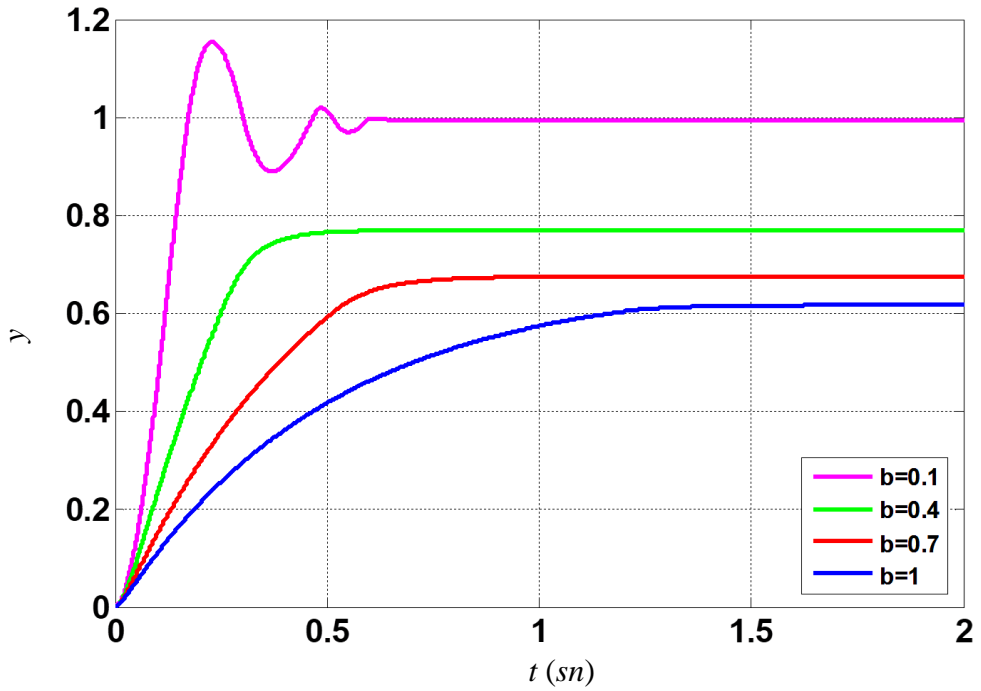
Şekil 4.19. Artan  $L$  değerleri için FLC birim basamak cevapları



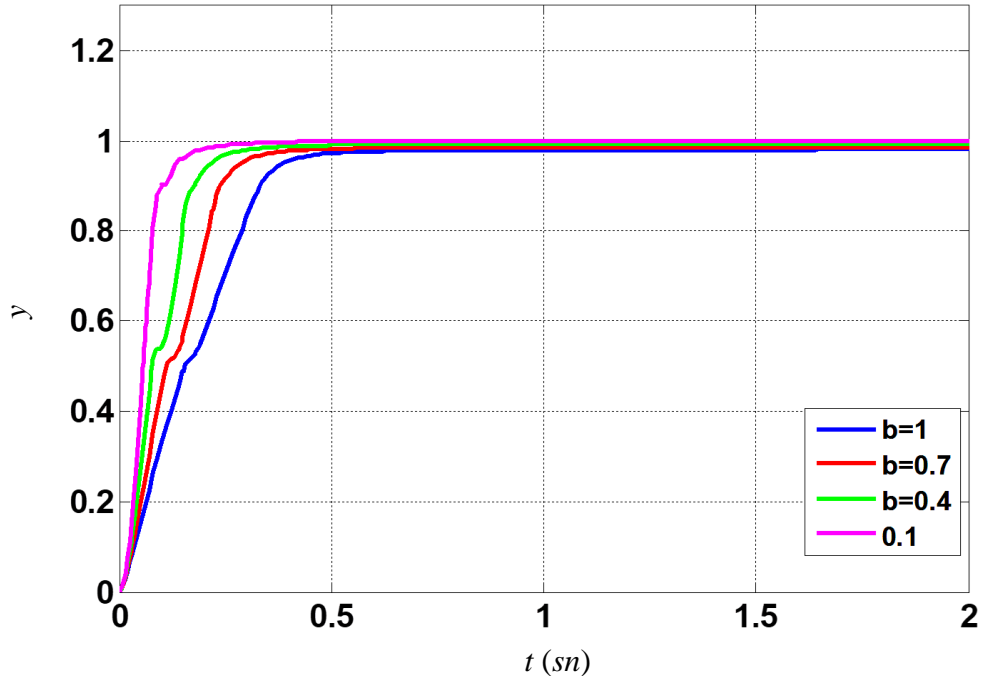
Şekil 4.20. Artan  $L$  değerleri için ANFIS-PI <sup>$\lambda$</sup> D <sup>$\mu$</sup>  kontrolör birim basamak cevapları



Şekil 4.21. Artan  $b$  değerleri için  $PI^\lambda D^\mu$  kontrolör birim basamak cevapları



Şekil 4.22. Artan  $b$  değerleri için FLC birim basamak cevapları



**Şekil 4.23.** Artan  $b$  değerleri için ANFIS-PI $^{\lambda}$ D $^{\mu}$  kontrolör birim basamak cevapları

Şekil 4.11 - 4.23'a kadar artan  $K$ ,  $J$ ,  $L$  ve  $b$  değerleri için PI $^{\lambda}$ D $^{\mu}$ , FLC ve ANFIS-PI $^{\lambda}$ D $^{\mu}$  kontrolörlerinin birim basamak çıkış cevapları karşılaştırılmıştır. Bu grafiklerden de görüldüğü gibi aynı motor parametreleri değişiminde ANFIS-PI $^{\lambda}$ D $^{\mu}$  kontrolörün birim basamak cevabı diğer kontrolörlere göre daha hızlı, daha az sönümlenmeli ve daha az aşımli cevaplar verdiği görülmektedir. Buna göre ANFIS, öğrenme mekanizması yoluyla daha iyi ayarlanmış PI $^{\lambda}$ D $^{\mu}$  denetleyici katsayıları sağlamıştır. Bu nedenle, ANFIS-PI $^{\lambda}$ D $^{\mu}$  kontrolörlerin, sistemin kontrolü üzerinde olumsuz etkileri olan çevresel koşullara karşı daha dayanıklı olduğu söylenebilir.

**Örnek 2.** Bu örnekte zaman gecikmeli birinci dereceden bir ısı değiştiricinin (heat exchanger) transfer fonksiyonu için ANFIS-PI $^{\lambda}$ D $^{\mu}$  yöntemi uygulanmıştır [139]. Isı değiştiricinin transfer fonksiyonu

$$G(s) = \frac{k}{as+1} e^{-Ts} = \frac{0.068}{23s+1} e^{-19.4s} \quad (4.14)$$

şeklinde ifade edilir.

Burada sırasıyla  $k$ ,  $a$  ve  $T$ , ısı deęiřtiricinin DC kazancı, zaman sabiti ve zaman gecikmesidir. Birinci dereceden Pade yaklaşımını kullanarak ısı deęiřtiricinin transfer fonksiyonu;

$$G(s) = \frac{0.068}{23s+1} e^{-19.3s} = \frac{0.068}{23s+1} \frac{-9.7s+1}{9.7s+1} \quad (4.15)$$

$$G(s) = -\frac{0.068 - 0.65s}{223.1s^2 + 32.7s + 1} \quad (4.16)$$

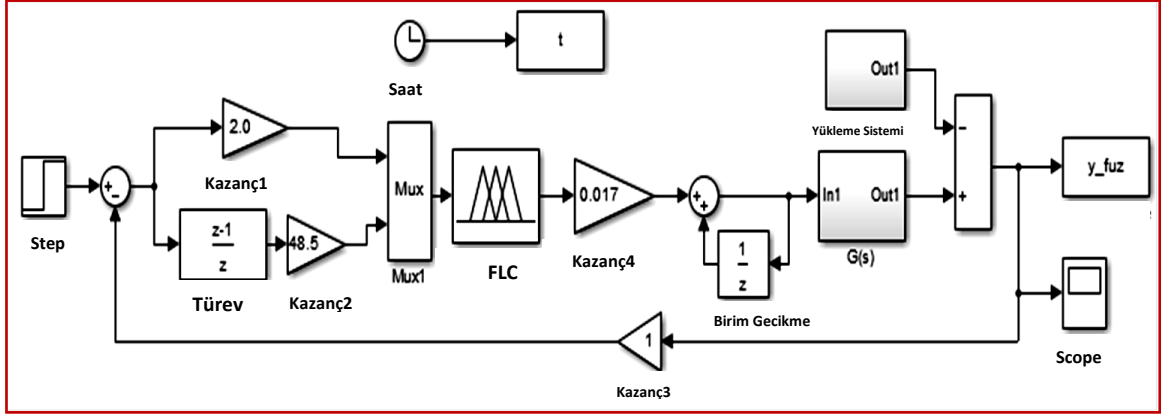
řeklinde yazılabilir. ANFIS' i eęitmek için kullanılan  $PI^\lambda D^\mu$  denetleyicisinin katsayıları genetik algoritma kullanarak ařaęıdaki gibi bulunmuřtur.

$$PI^\lambda D^\mu: K_p = 20.50, K_i = 1.22, K_d = 80, \lambda = -0.85, \mu = 0.80 \quad (4.17)$$

$K_p$ ,  $K_i$  ve  $K_d$  için ANFIS eęitimleri için beř tane Gauss üyelik fonksiyonu ve beř tane kural kullanılmıřtır.

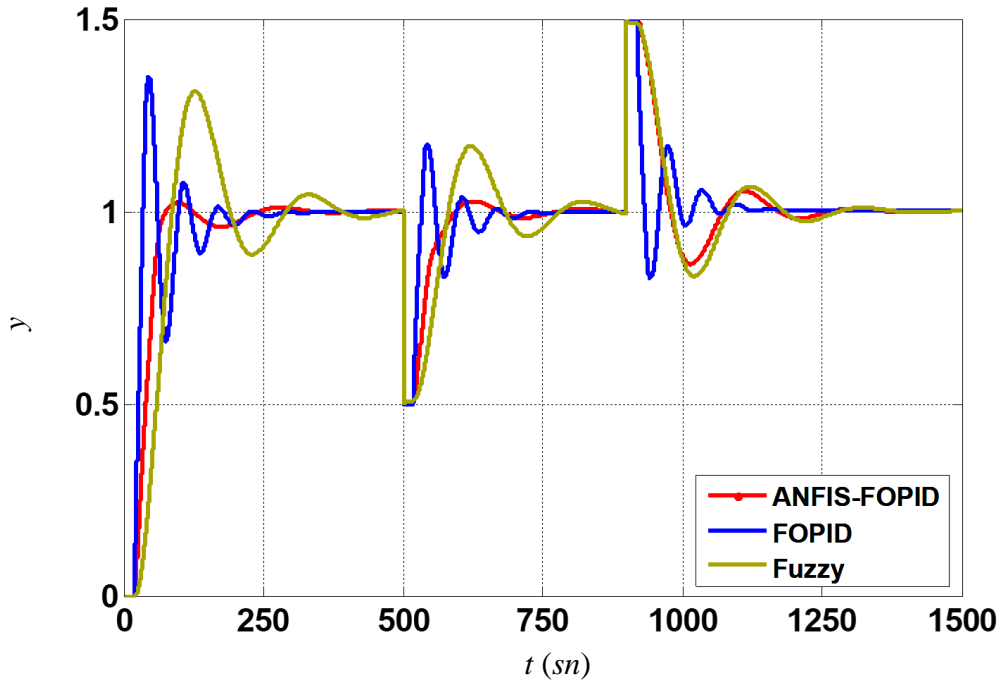
řekil 4.24'de ise FLC ile kontrol edilen ısı deęiřtirici sisteminin blok diyagramı gösterilmiřtir. Hata ( $e$ ) ve hatadaki deęiřim ( $ce$ ) giriř deęiřkenleri için 5 üyelik fonksiyonu ve dolayısıyla 25 tane kural kullanılmıřtır [140]. Üyelik fonksiyonu ve kurallar MATLAB Fuzzy Toolbox programında yazılmıř ve fis dosyası olarak Simulink' te tasarlanan kontrol sistemindeki FLC ünitesine aktarılmıřtır. řekil 4.24' de, Kazanç4 doęal olarak sistemin çıkıř cevabının hızını etkilemektedir. Eęer bu deęer (Kazanç4) artarsa çıkıř cevabının hızı artar, aksi durumda çıkıř cevabının hızı azalır. Kontrolörlerin çıkıř cevaplarını eřit řartlarda karřılařtırmak için, Kazanç4, FLC sisteminin çıkıř cevabının hızı,  $PI^\lambda D^\mu$  ve ANFIS- $PI^\lambda D^\mu$  kontrolörlerinin çıkıř cevaplarının hızına yakın olabilecek řekilde belirlenmiřtir.





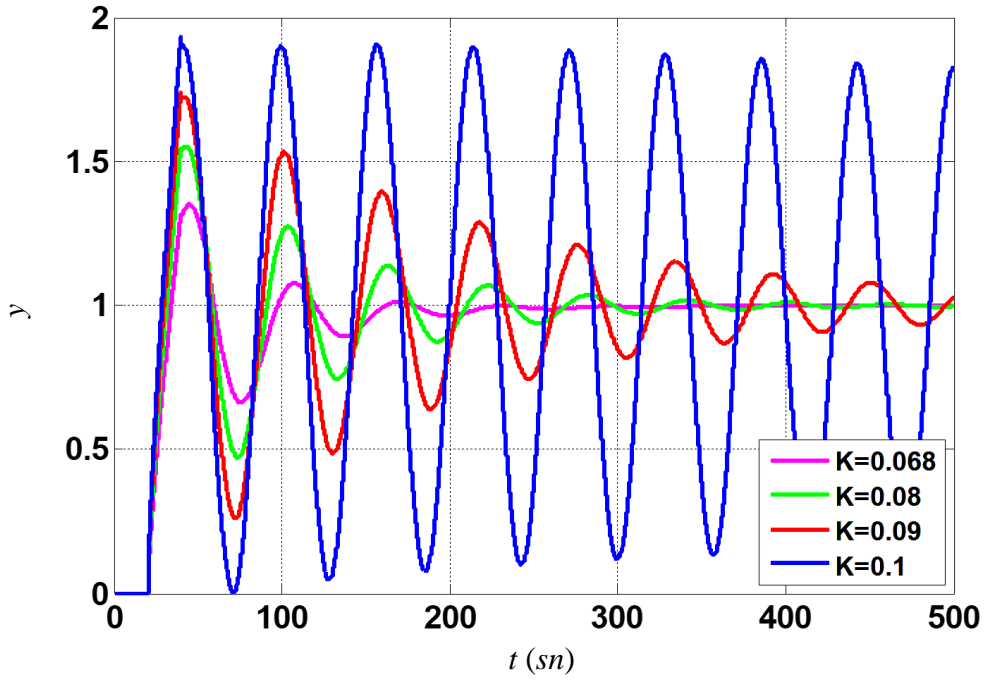
Şekil 4.24. FLC içeren ısı değiştirici sistemin kontrol blok diyagramı [140]

Bu durumda farklı DC kazanç değerleri ( $k$ ), zaman sabiti ( $a$ ) ve zaman gecikmesi ( $T$ ) için,  $PI^\lambda D^\mu$ , FLC ve ANFIS- $PI^\lambda D^\mu$  kontrolörlerinin birim basamak cevapları Şekil 4.25 – 4.34'e kadar olan grafiklerde gösterilmiştir.

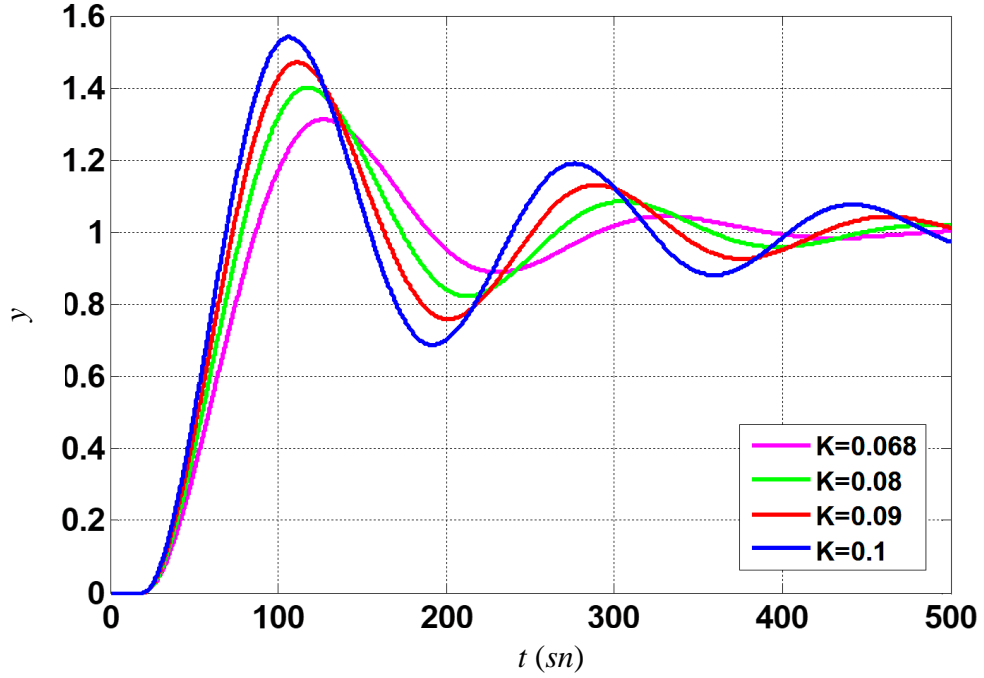


Şekil 4.25. Aynı DC kazancı için  $k = 0.068$ , zaman sabiti  $a = 23$  ve zaman gecikmesi  $T = 19.4$  için  $PI^\lambda D^\mu$ , FLC ve ANFIS- $PI^\lambda D^\mu$  kontrolörlerin birim basamak cevapları

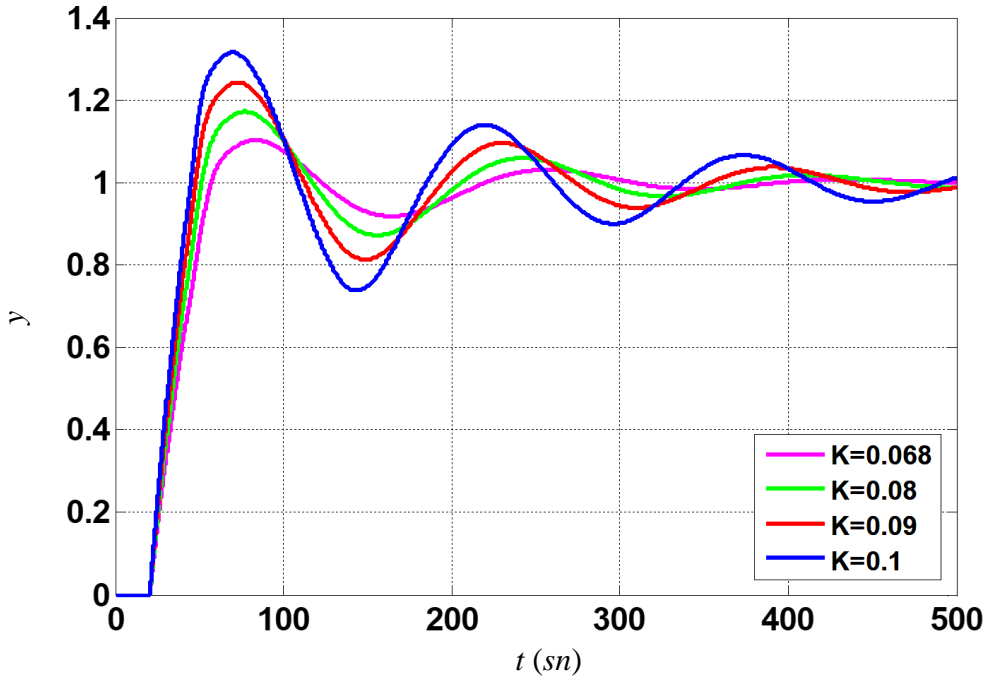
Şekil 4.25'de gösterildiği gibi  $PI^\lambda D^\mu$ , FLC ve ANFIS- $PI^\lambda D^\mu$  kontrolörlerin karşılaştırmalı birim basamak cevaplarından ANFIS- $PI^\lambda D^\mu$  kontrolör ile daha az aşım, kısa sönümlü ve daha az salınımlı cevaplar alındığı görülür.



Şekil 4.26. Artan DC kazanç ( $k$ ) değerleri için  $PI^\lambda D^\mu$  kontrolörünün birim basamak cevapları ( $K_p = 20.50$ ,  $K_i = 1.22$ ,  $K_d = 80$ ,  $\lambda = -0.85$ ,  $\mu = 0.80$ )

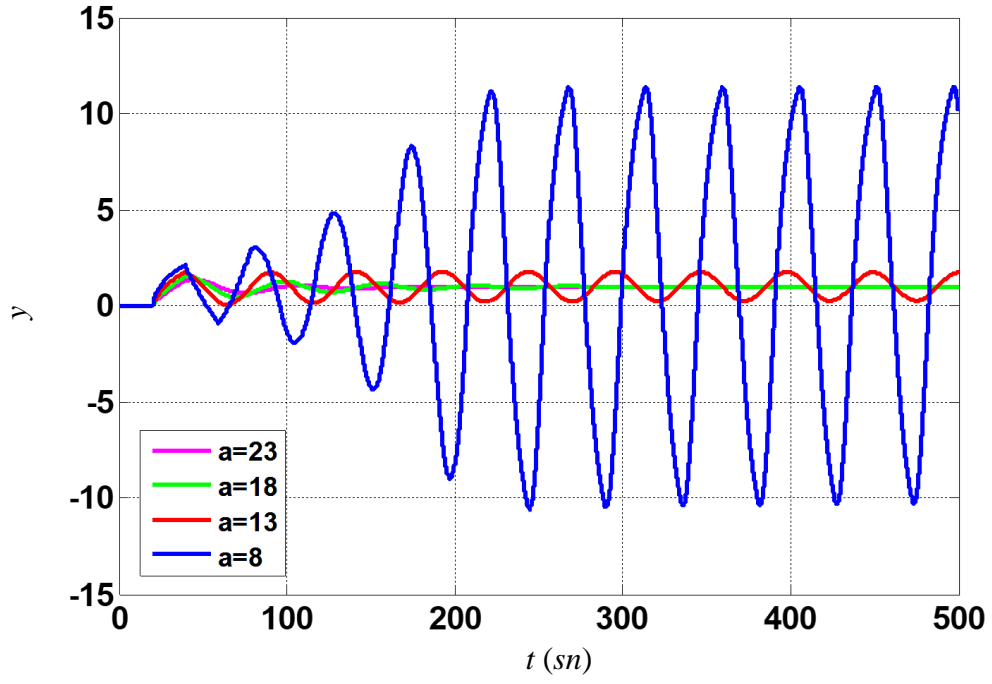


Şekil 4.27. Artan DC kazanç ( $k$ ) değerleri için FLC'nin birim basamak cevapları

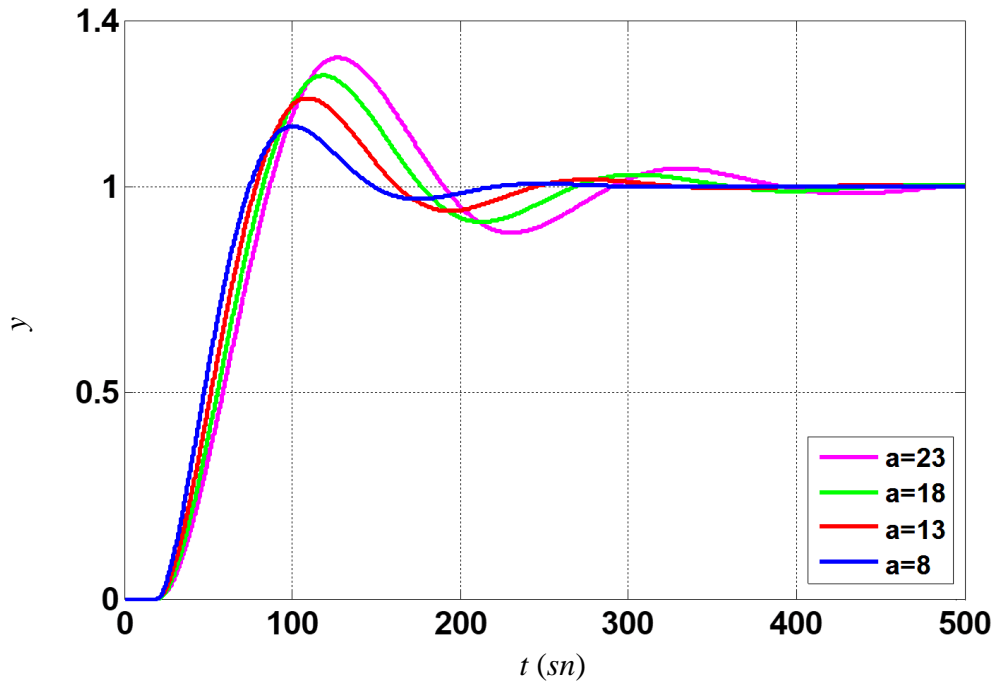


**Şekil 4.28.** Artan DC kazanç ( $k$ ) değerleri için ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolörünün birim basamak cevapları

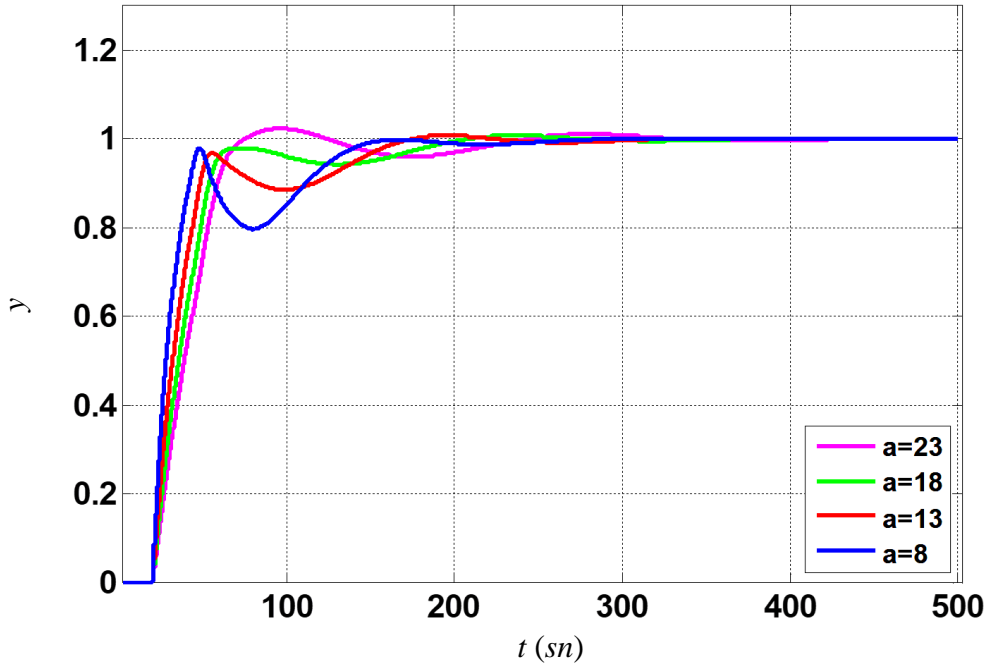
Şekil 4.26, 4.27 ve 4.28’de görüldüğü gibi,  $k$  değeri arttıkça ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolörün birim basamak cevabının PI<sup>λ</sup>D<sup>μ</sup> ve FLC kontrolörlerinin birim basamak cevaplarına göre daha hızlı, daha az sönümlenmeli ve daha az aşım oranına sahip cevaplar verdiği görülmektedir. Bunun nedeni, ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolöründeki  $K_p$ ,  $K_i$  ve  $K_d$  parametrelerinin geniş bir varyasyon aralığı için eğitilmiş bir ANFIS yapısını kullanılmasıdır. Bu durumda ısı değıştricinin transfer fonksiyonundaki DC kazancı ( $k$ ) değışirse önerilen yöntem daha esnek ve dayanıklı kontrol sağlayabilir.



Şekil 4.29. Artan zaman sabitleri ( $a$ ) için  $PI^1D^1$  kontrolörünün birim basamak cevapları

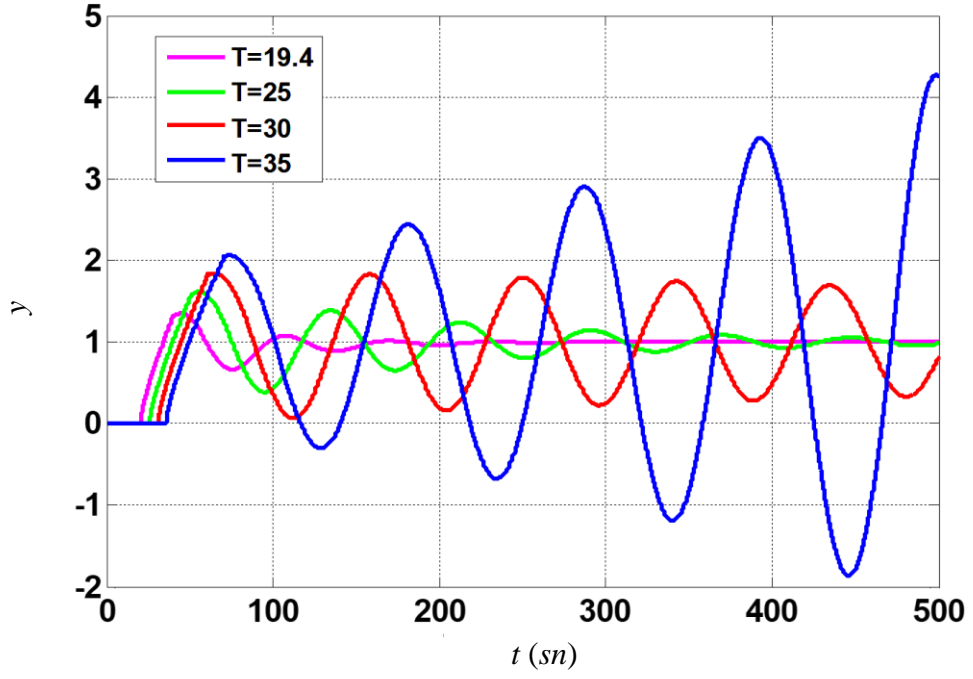


Şekil 4.30. Artan zaman sabitleri ( $a$ ) için FLC'nin birim basamak cevapları

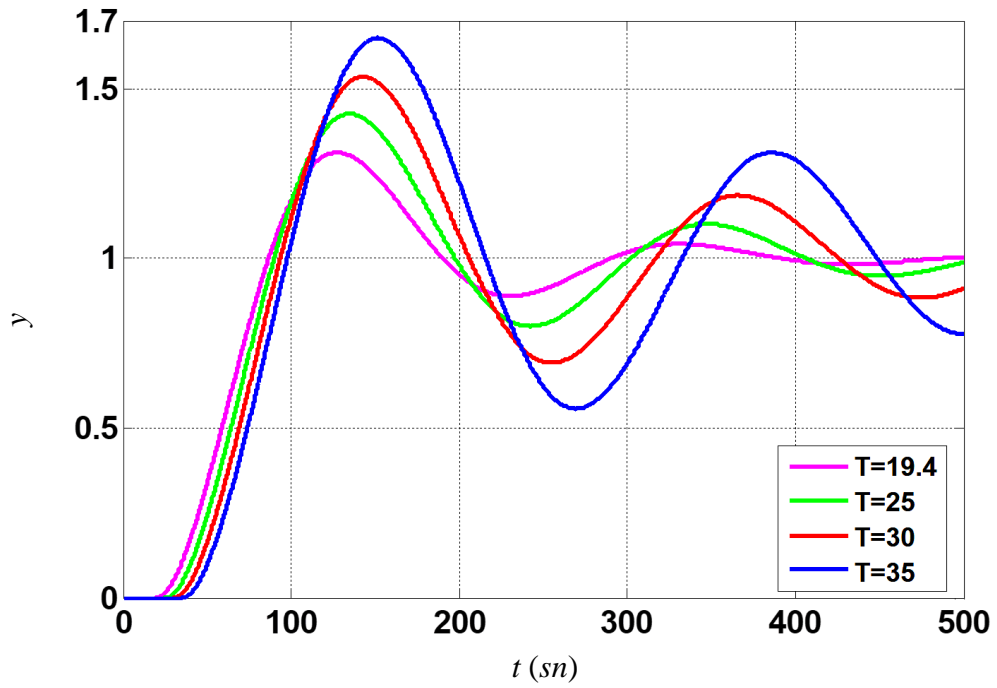


Şekil 4.31. Artan zaman sabitleri ( $a$ ) için ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolörünün birim basamak cevapları

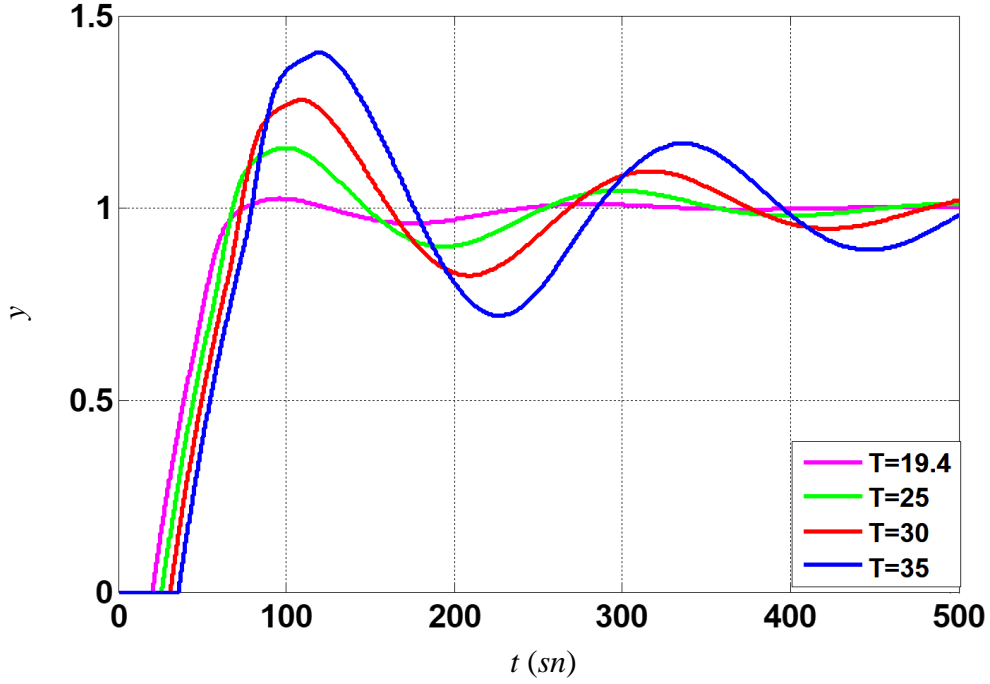
Şekil 4.29, 4.30 ve 4.31’de görüldüğü gibi, zaman sabiti ( $a$ ) değerleri arttırıldığında PI<sup>λ</sup>D<sup>μ</sup> kontrolörün cevaplarında olumsuz yönde çok büyük değişiklikler görülür. FLC’ nin cevaplarındaki genlik salınımları, PI<sup>λ</sup>D<sup>μ</sup> kontrolörüne göre daha azdır. Fakat zaman sabiti ( $a$ ) değişimine karşılık diğer kontrolörlere göre en az salınım zamanı ve en az aşım oranı ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolörün cevaplarında görülmüş, dayanıklı ve hızlı cevaplar alınmıştır.



Şekil 4.32. Artan zaman gecikme sabitleri ( $T$ ) için  $PI^\lambda D^\mu$  kontrolörünün birim basamak cevapları



Şekil 4.33. Artan zaman gecikme sabitleri ( $T$ ) için FLC'nin birim basamak cevapları



Şekil 4.34. Artan zaman gecikme sabitleri ( $T$ ) için ANFIS- $PI^{\lambda}D^{\mu}$  kontrolörünün birim basamak cevapları

Şekil 4.32, 4.33 ve 4.34'de görüldüğü gibi,  $PI^{\lambda}D^{\mu}$  kontrolörün cevaplarında salınım  $T = 30$  değerinde başlar ve  $T = 35$  değerinde kararsız olur. FLC' nin birim basamak cevabının aşım oranı daha yüksektir ve çıkış cevabı ANFIS- $PI^{\lambda}D^{\mu}$  kontrolöre göre daha yavaştır. ANFIS- $PI^{\lambda}D^{\mu}$  kontrolör  $T = 30$  ve  $T = 35$  değerleri için kararludur ve sadece sönüm genliği ve aşım oranı az miktarda artar. Zaman gecikmesinin ( $T$ ) farklı değerleri için, ANFIS- $PI^{\lambda}D^{\mu}$  kontrolörün,  $PI^{\lambda}D^{\mu}$  ve FLC kontrolörlerine göre daha dayanıklı ve hızlı olduğu görülür.

Bu kısımda, bir sistemin kontrolünü iyileştirmek için kesir dereceli  $PI^{\lambda}D^{\mu}$  kontrolörünün katsayılarının ( $K_p$ ,  $K_d$  ve  $K_i$ ) ayarlanmasında ANFIS'in kullanılması incelenmiş ve sonuçlar diğer kontrolörler ile karşılaştırılmıştır. Bu örneklerdeki benzetim çalışmaları ile çevresel etkenlerden herhangi bir değişiklik meydana gelmesi durumunda,  $PI^{\lambda}D^{\mu}$  kontrolör ile gerçekleştirilen kontrol uygulamalarında taşmaların arttığı ve sönümlemenin uzadığı görülür ve bu sebeple kontrol sistemi kararlılık ve dayanıklılık açısından daha az duyarlıdır. Çevresel koşullar nedeni ile kontrol edilen sistemin katsayıları değiştiğinde FLC ile yapılan kontrolde daha kararlı ve dayanıklı birim basamak cevapları alınmakla birlikte FLC tek başına kesir dereceli  $PI^{\lambda}D^{\mu}$  kontrolörüne göre daha yavaş cevap verdiği görülür. FLC'nin hızı arttıkça, çıkış

cevabındaki osilasyon ve aşım oranı artar ve bu nedenle kontrol sistemi olumsuz olarak etkilenir.

Sistem parametreleri geniş bir aralıkta değiştiğinde  $PI^\lambda D^\mu$  denetleyicisinin parametreleri dinamik olarak ayarlandığı için önerilen ANFIS- $PI^\lambda D^\mu$  kontrolör ile sistem, daha küçük aşım oranı ve daha kısa süreli sönümlenme ile çok daha dayanıklı bir şekilde kontrol edilebilmektedir. Bu nedenle, karşılaştırmalı benzetim eğrilerinden de görüldüğü üzere önerilen yöntem ile tasarlanan kontrolörün  $PI^\lambda D^\mu$  ve FLC kontrolörlerinden daha dayanıklı, kararlı ve hızlı olduğu görülür.

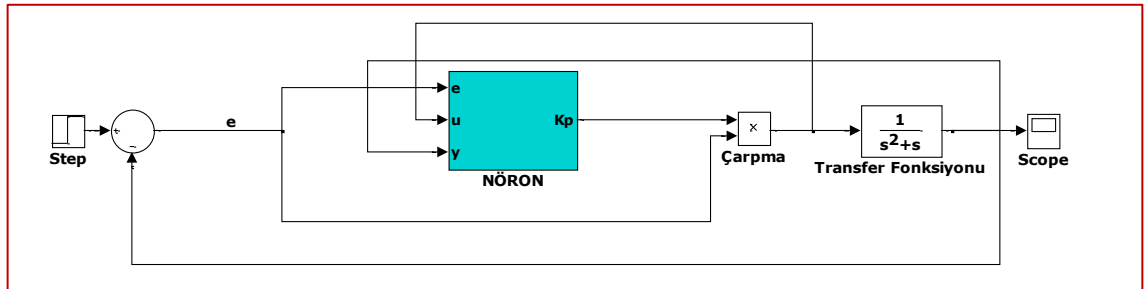
### 4.3. Otomatik Ayarlamalı Sinir Hücresi İle $PI^\lambda D^\mu$ Kontrolör Tasarım Uygulamaları

Bu bölümde uyarlamalı P,  $PI^\lambda$ ,  $PD^\mu$  ve  $PI^\lambda D^\mu$  kontrolör yapıları için MATLAB-Simulink tasarımları ve benzetim sonuçları gösterilmiştir. Birinci örnek ile uyarlamalı P,  $PI^\lambda$ ,  $PD^\mu$  ve  $PI^\lambda D^\mu$  kontrolör tasarımları ve benzetim sonuçları verilmiştir. İkinci örnekte ise uyarlamalı  $PI^\lambda D^\mu$  kontrolör kullanarak zaman gecikmeli sistem modeli için elde edilen birim basamak cevapları alınmış ve bu sonuçlar ZN yöntemi ile ayarlanan  $PI^\lambda D^\mu$  kontrolörün sonucu ile karşılaştırılmıştır.

#### Örnek 1.

##### 4.3.1. P Kontrolör

Birinci tasarımdaki hedef, girişleri hata ( $e$ ), kontrolör çıkış sinyali ( $u$ ) ve sistem çıkış sinyali ( $y$ ) olan bir kontrol sisteminde  $K_p$  katsayısı sinir hücresi ile ayarlanarak sistemin kontrol edilmesi istenmektedir. Tasarımın MATLAB-Simulink blok diyagramı Şekil 4.35’de gösterilmiştir.

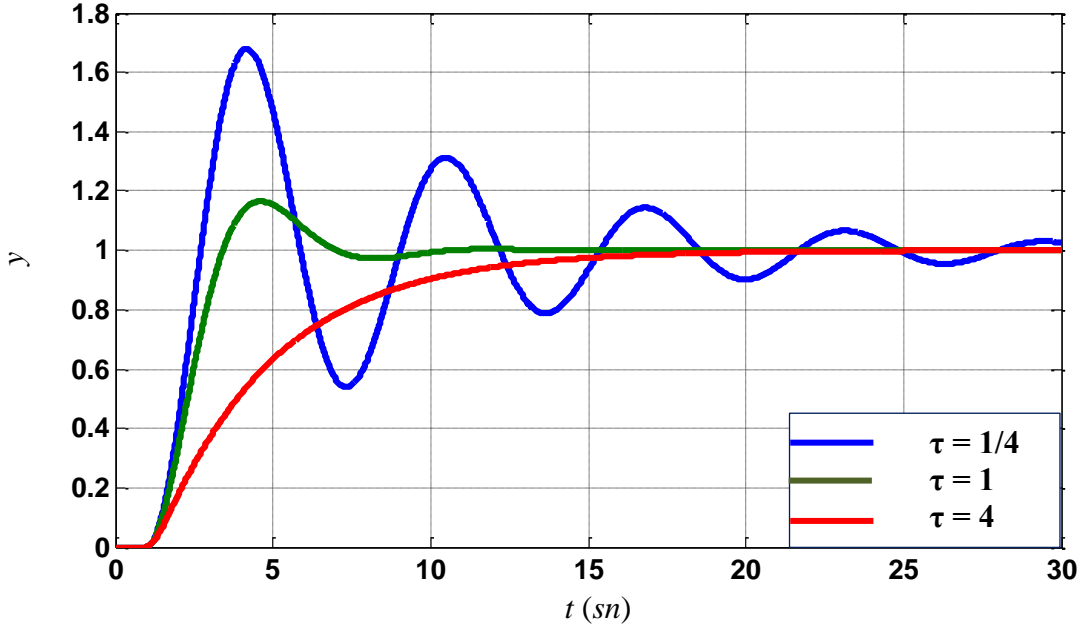


Şekil 4.35. MATLAB – Simulink’te otomatik ayarlamalı sinir hücresi ile P kontrolör blok diyagramı



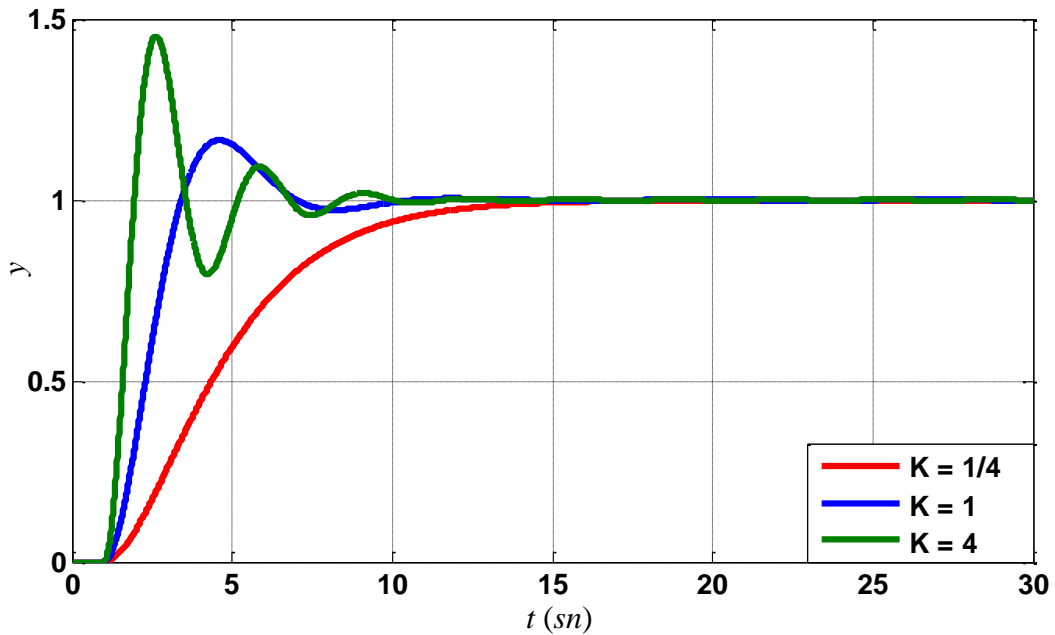
Bu tasarım ile Denklem (4.17)'deki sistem için artan  $K$  ve  $\tau$  değerleri için birim basamak çıkış cevapları alınmış ve grafikler Şekil 4.36 ve Şekil 4.37'de gösterilmiştir.

$$G_c(s) = \frac{K}{s(s+\tau)} \quad (4.17)$$



Şekil 4.36. Artan  $\tau$  değerleri için kontrol sisteminin birim basamak cevapları

Şekil 4.36'da zaman sabiti  $\tau$  arttıkça sistem çıkışı daha kararlı duruma gelmektedir. Fakat doğal olarak sistem çıkış cevabı yavaşlamaktadır.

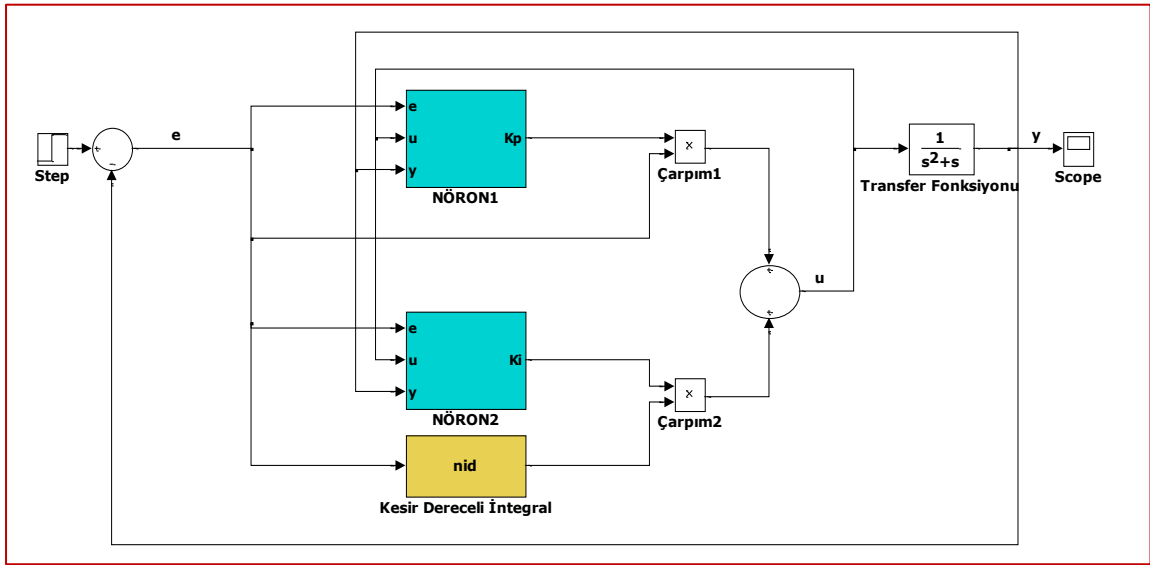


Şekil 4.37. Artan  $K$  değerleri için kontrol sisteminin birim basamak cevapları

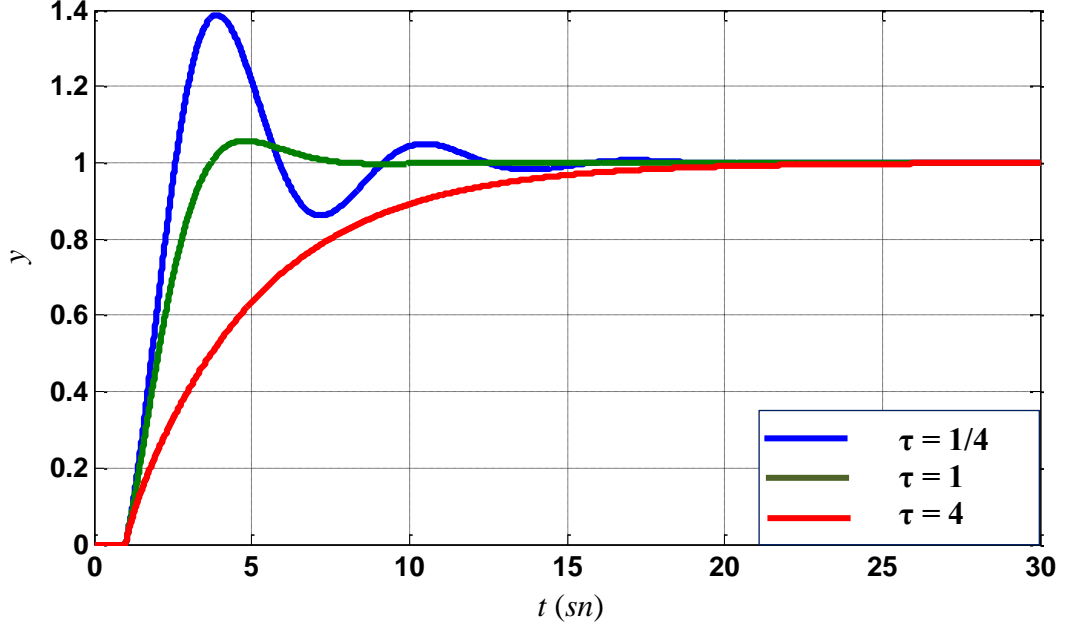
Şekil 4.37’den görüleceği gibi  $K$  değeri arttıkça sistem cevabı hızlanmakta fakat aşım ve kısa süreli bir salınım oluşmaktadır.  $K$  değeri azaldıkça sistem cevabı yavaşlamakta aşım ve salınım oluşmamaktadır.

#### 4.3.2. $PI^\lambda$ Kontrolör

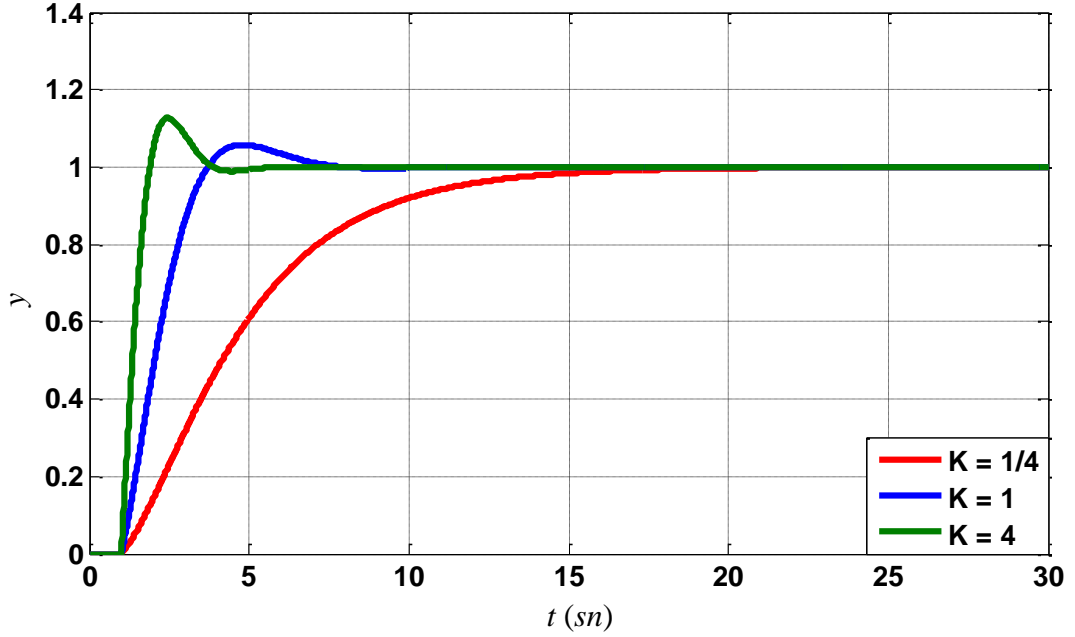
Kullanılan  $PI^\lambda D^\mu$  kontrolörün  $K_d$  katsayısı 0 olarak alınmıştır. Denklem (4.17)’daki sistem transfer fonksiyonu kullanılmıştır.  $PI^\lambda$  kontrolör sistemi Şekil 4.38’de gösterilmiştir. Sistem transfer fonksiyonunda artan  $K$  ve  $\tau$  değerleri için birim basamak çıkış cevapları Şekil 4.39 ve Şekil 4.40’da gösterilmiştir.



Şekil 4.38. MATLAB – Simulink’te otomatik ayarlamalı sinir hücresi ile  $PI^\lambda$  kontrolör blok diyagramı



Şekil 4.39. Artan  $\tau$  değerleri için kontrol sisteminin birim basamak cevapları

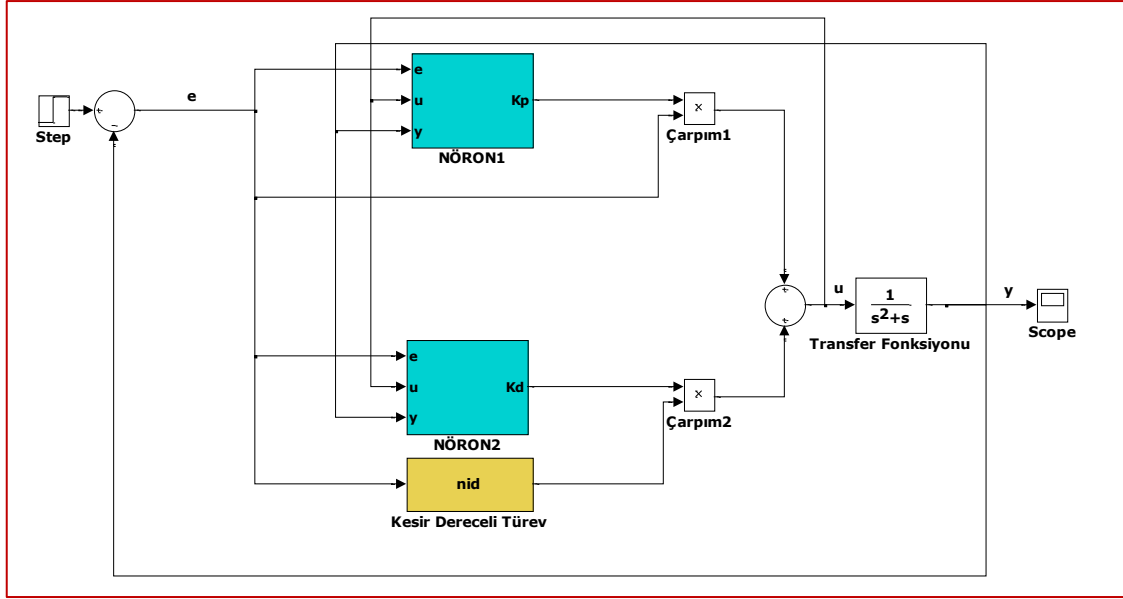


Şekil 4.40. Artan  $K$  değerleri için kontrol sisteminin birim basamak cevapları

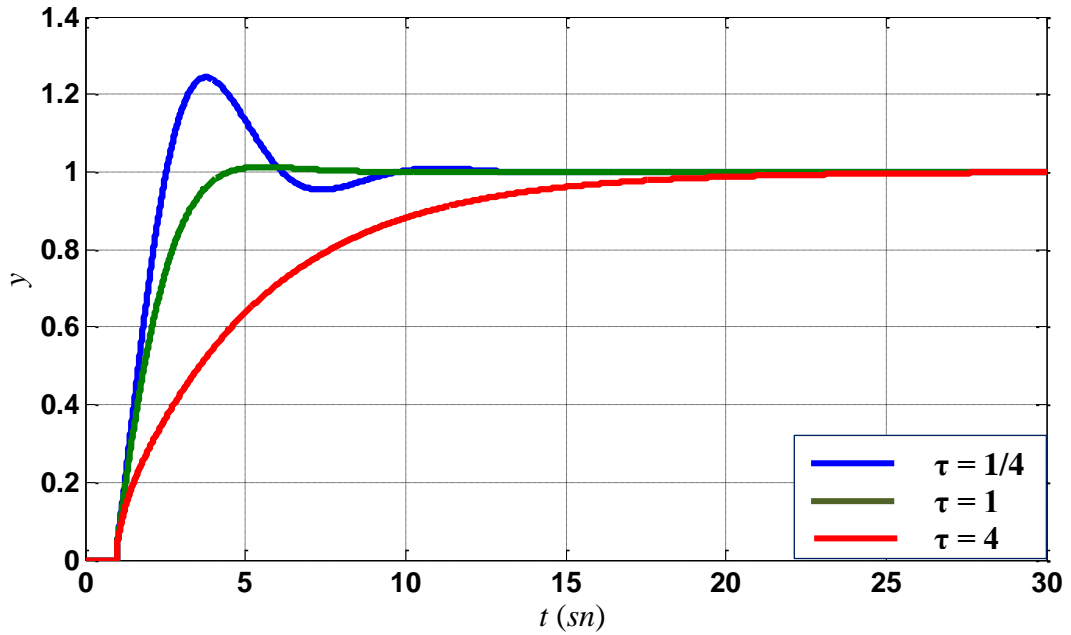
Şekil 4.36 - 4.37 ve Şekil 4.39 – 4.40 incelendiğinde P kontrolöre göre aynı şartlar altında daha hızlı cevap verdiği görülmektedir. Buna karşılık  $PI^\lambda$  kontrolörün çıkış cevabında salınımlar artmaktadır.

### 4.3.3. PD<sup>μ</sup> Kontrolör

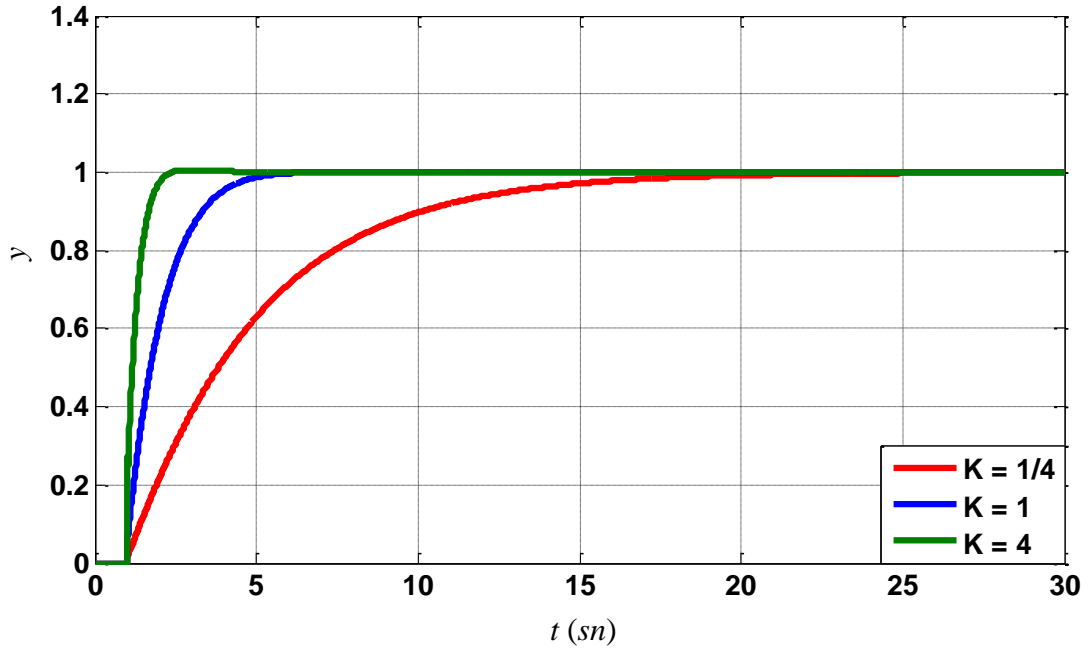
Kullanılan PI<sup>λ</sup>D<sup>μ</sup> kontrolörün  $K_i$  katsayısı 0 olarak alınmıştır. Denklem (4.17)'deki sistem transfer fonksiyonu kullanılmıştır. PD<sup>μ</sup> kontrolör sistemi Şekil 4.41'de gösterilmiştir. Sistem transfer fonksiyonunda artan  $K$  ve  $\tau$  değerleri için birim basamak çıkış cevapları Şekil 4.42 ve Şekil 4.43'de gösterilmiştir.



Şekil 4.41. MATLAB – Simulink'te otomatik ayarlamalı sinir hücresi ile PD<sup>μ</sup> kontrolör blok diyagramı



Şekil 4.42. Artan  $\tau$  değerleri için kontrol sistemin birim basamak cevapları

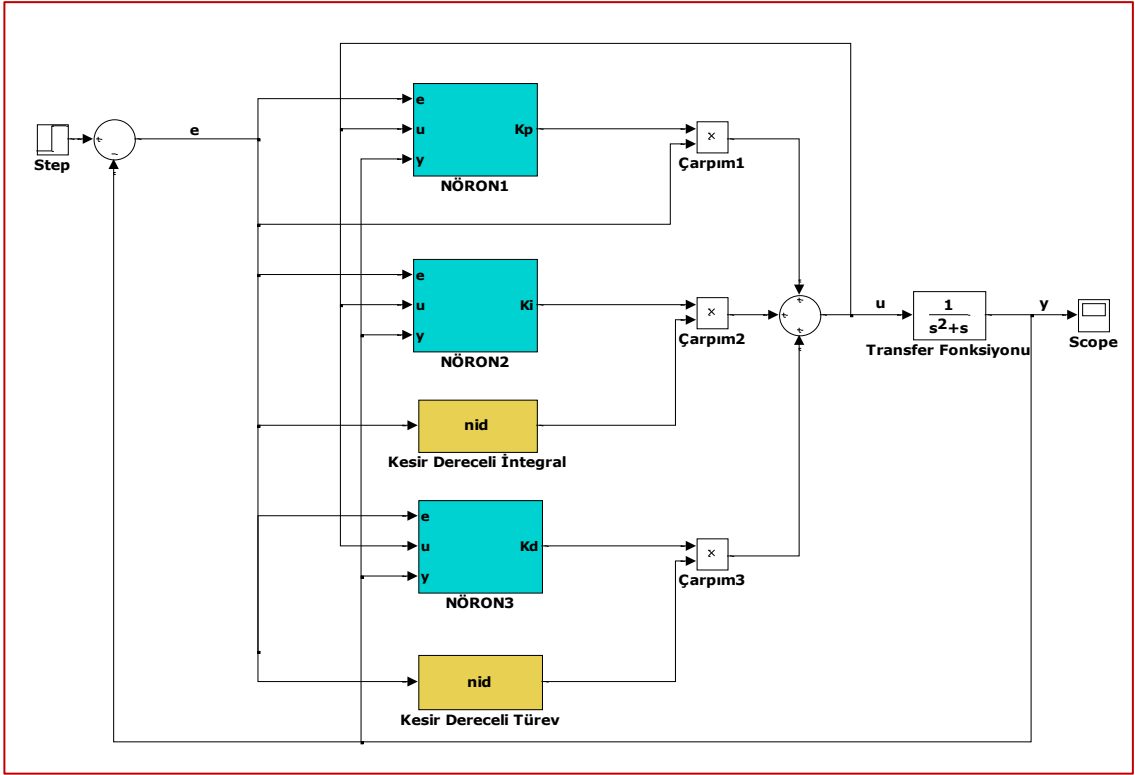


Şekil 4.43. Artan K değerleri için kontrol sistemin birim basamak cevapları

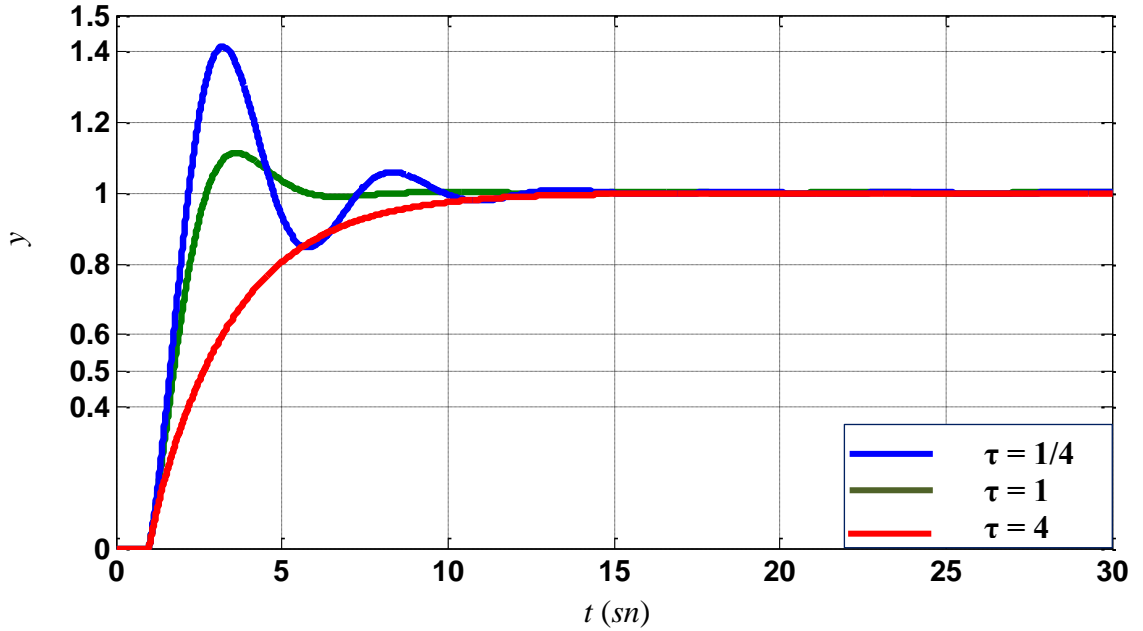
Şekil 4.42 ve Şekil 4.43 incelendiğinde bu sistem transfer fonksiyonu için  $PD^\mu$  kontrolörün, P ve  $PI^\lambda$  kontrolörlere göre daha hızlı ve dayanıklı olduğu görülmektedir.

#### 4.3.4. $PI^\lambda D^\mu$ Kontrolör

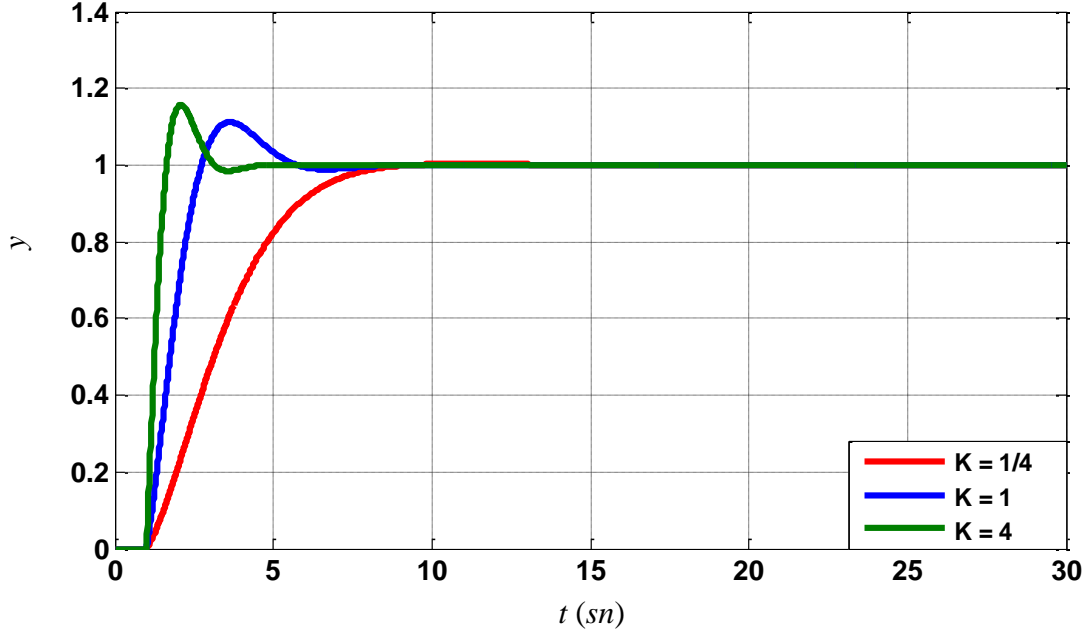
Aynı şekilde Denklem (4.17)'deki sistem transfer fonksiyonu kullanılarak  $PI^\lambda D^\mu$  kontrolör sistemi Şekil 4.44'de, artan  $K$  ve  $\tau$  değerleri için birim basamak çıkış cevapları ise Şekil 4.45 ve Şekil 4.46'da gösterilmiştir.



Şekil 4.44. MATLAB – Simulink’te otomatik ayarlamalı sinir hücresi ile  $PI^{\lambda}D^{\mu}$  kontrolör sistemin blok diyagramı



Şekil 4.45. Artan  $\tau$  değerleri için kontrol sistemin birim basamak cevapları



Şekil 4.46. Artan K değerleri için kontrol sistemin birim basamak cevapları

Şekil 4.45 ve Şekil 4.46'da görüldüğü gibi  $PI^\lambda D^\mu$  kontrolör, P,  $PI^\lambda$  ve  $PD^\mu$  kontrolörlere göre daha kararlı ve hızlı cevaplar vermiştir.

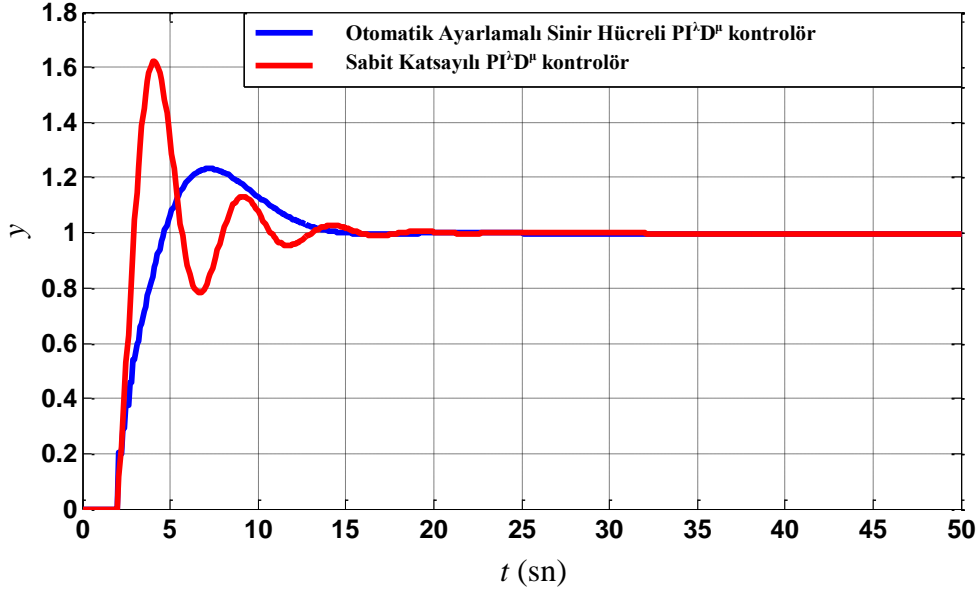
Benzetim sonuçlarından uyarlamalı  $PI^\lambda D^\mu$  kontrolörün,  $PI^\lambda$  ve  $PD^\mu$  kontrolörlerine göre daha az salınımlı, aşım genliği küçük ve ayrıca daha hızlı cevaplar verebildiği görülmektedir. Şekil 4.44'de, önerilen otomatik ayarlamalı sinir hücresi ile tasarlanan  $PI^\lambda D^\mu$  kontrolörün istenen set değerini yaklaşık 10 saniyede sıfır hata ile yakalamıştır.  $PI^\lambda$ ,  $PD^\mu$  ve  $PI^\lambda D^\mu$  kontrolörlerinin kazanç katsayıları, kontrolör hatası sıfıra giderken otomatik ayarlama sinir hücreleri vasıtasıyla sürekli ayarlanır. Sinir hücreleri benzetim çalışması sırasında kontrolörün kazanç katsayılarının değişmesine neden olur ve  $PI^\lambda D^\mu$  kontrolörüne değişken kazanç katsayılarının avantajını sağlar, böylece  $PI^\lambda D^\mu$  kontrolörün kontrol performansı artar. Adaptasyon becerisi ile ayrıca kontrol edilen sistemin çevresel etkiler nedeni ile parametrelerinde birtakım değişimler gerçekleşirse kontrolörün kazanç katsayılarını çevrimiçi ayarlamasını sağlar, bu durum  $PI^\lambda D^\mu$  kontrolörün hızını ve dayanıklılığını artırır.

## Örnek 2.

Farklı ve daha karmaşık sistemlerde tezde önerilen yöntemin nasıl cevap verdiğini görmek için Denklem (4.18)'de verilen birinci derece zaman gecikmeli bir sistem (gerçek şeker kamışı ham suyu nötralizasyon işleminin dinamik pH modeli)

kullanılmıştır [141]. Otomatik ayarlamalı sinir hücresi ile tasarlanan ve ZN yöntemine göre katsayıları bulunan  $PI^{\lambda}D^{\mu}$  kontrolörlerin verdikleri cevaplar Şekil 4.47’de gösterilmiştir.

$$G(s) = \frac{k}{\tau s + 1} e^{-Ls} = \frac{0.55}{62s + 1} e^{-1s} \quad (4.18)$$



**Şekil 4.47.** Sabit katsayılı ve otomatik ayarlamalı sinir hücreli  $PI^{\lambda}D^{\mu}$  kontrolör sisteminin birim basamak cevapları

Şekil 4.47’de görüldüğü gibi  $PI^{\lambda}D^{\mu}$  kontrolörün birim basamak çıkış cevabı 30 sn’de kalıcı duruma ulaşmaktadır ve aşım genliği fazladır. Fakat otomatik ayarlamalı sinir hücreli  $PI^{\lambda}D^{\mu}$  kontrolörün birim basamak cevabı ise yaklaşık 15 sn’de istenen set değerine ulaşmaktadır. Aşım genliği ve salınımlar diğerine göre çok daha azdır.

Bu kısımda,  $PI^{\lambda}D^{\mu}$  kontrolörünün otomatik ayarlamalı sinir hücreleri ile uygulaması gerçekleştirilmiştir. Bu sinir hücreleri eğimli azalım (gradient descent) optimizasyon yöntemini kullanarak kazanç katsayılarını ( $K_p$ ,  $K_i$ ,  $K_d$ ) çevrimiçi optimize ederek ayarlandı. Bu durum, değişken kazanç katsayılı  $PI^{\lambda}D^{\mu}$  kontrolör yapısı oluşturduğundan;  $PI^{\lambda}D^{\mu}$  kontrolörün birim basamak cevaplarını geliştirir.  $PI^{\lambda}D^{\mu}$  kontrolörünün birim basamak cevaplarının gelişmesine ek olarak, kazanç katsayılarının çevrimiçi olarak ayarlanıyor olmaları,  $PI^{\lambda}D^{\mu}$  kontrolörüne adaptasyon becerisi de kazandırır. Adaptasyon becerisi gerçek uygulamalarda  $PI^{\lambda}D^{\mu}$  kontrolörünün kontrol performansını ve dayanıklılığını artırır. Otomatik ayarlamalı sinir hücreli  $PI^{\lambda}D^{\mu}$  kontrolörlerinin tasarımı



aşamasında  $K_p$ ,  $K_i$  ve  $K_d$  katsayılarının önceden belirlenmesine gerek yoktur.  $PI^\lambda D^\mu$  kontrolörünün tasarımı sadece kesir derecelerinin ( $\lambda$  ve  $\mu$ ) bulunmasına indirildiği için basitleşmiştir. Geliştirilen yöntemde  $PI^\lambda D^\mu$  kontrolörünün performansını göstermek için P,  $PI^\lambda$ ,  $PD^\mu$  ve  $PI^\lambda D^\mu$  kontrolör yapıları için iki örneğin MATLAB-Simulink tasarımı ve benzetim sonuçları verilmiştir. Zaman gecikmeli sistemler için önerilen  $PI^\lambda D^\mu$ 'nin kontrol performansı, ZN yöntemi ile ayarlanan  $PI^\lambda D^\mu$ 'nin performansıyla karşılaştırılmıştır.

## 5. TARTIŞMA ve SONUÇ

### 5.1. Tartışma

- I. MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolör ile farklı iki örnek için elde edilen sonuç eğrileri (Şekil 4.3, 4.4 ve Şekil 4.7, 4.8) incelendiğinde görülmektedir ki sistem parametresindeki değişimlerine karşı (artan  $K$  değerleri için) daha dayanıklı ve kararlı cevaplar verdiği görülmüştür. Şekil 4.2 ve Şekil 4.6 grafiklerinden yapılan karşılaştırmalar sonucundan anlaşılacağı üzere; MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolörün MRA-PID kontrolörden daha hızlı bir yapıya sahip olduğu görülür. Fakat bu tasarımda parametre sayısı azaltılmaya çalışılmış olmakla birlikte, tasarım sırasında yeni parametrelerin çıkması ( $\alpha$ ,  $\beta$ ,  $\gamma$ ); bu modelin uygulanabilirlik ve pratik kullanım kolaylığını azaltan bir etken olarak değerlendirilebilir.
- II. Şekil 4.12 - 4.23 ve Şekil 4.26 - 4.34'den görüldüğü gibi ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolör ile birinci ve ikinci örneklerden alınan cevaplardan görülmektedir ki son derece hızlı ve kararlı cevaplar vermektedir. Parametrelerdeki değişimlere ( $K$ ,  $J$ ,  $L$ ,  $b$  ve  $k$ ,  $a$ ,  $T$ ) karşı daha dayanıklı davranıp sistemi kararlılıkla kontrol edebilmiştir. Şekil 4.11 ve Şekil 4.25'de görüldüğü gibi sabit katsayılı PI<sup>λ</sup>D<sup>μ</sup> kontrolör ile yapılan karşılaştırmalı benzetim sonuçları göstermektedir ki ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolör daha hızlı, başlangıçta küçük bir taşma olmakla birlikte ilerleyen süreçte daha az salınımlı cevap vermektedir. Ayrıca bu tasarımlarla kazanılan diğer bir avantaj  $K_p$ ,  $K_i$  ve  $K_d$  parametreleri ANFIS blokları ile ayarlanmakta ve kullanıcının ayarlamak zorunda kalacağı parametre sayısı sadece iki tanedir ( $\lambda$  ve  $\mu$ ).
- III. Otomatik ayarlamalı sinir hücresi ile gerçekleştirilen dört farklı tasarım sonuçları sırası ile;  
**P** kontrolör ile sistem ( $K = 1$  ve  $\tau = 1$  için) yaklaşık 10 sn'lik bir sürede kararlı konuma gelmiştir (Şekil 4.36 - 4.37). Fakat sadece oransal ( $P$ ) kontrol olduğu için parametrelerdeki değişimlere karşı dayanıklılık gösterememiştir.  
**PI<sup>λ</sup>** kontrolör ile alınan cevap eğrilerinde ise sistem yine yaklaşık 10 sn'de kararlı hale gelmiştir. Bu durumda kontrolörün sistem çıkış cevabının istenen değere ulaşma süresi azalmış fakat başlangıçta aşım meydana gelmiştir. Ayrıca sistemin zaman sabiti  $\tau$ 'nun değişimine karşı duyarlıdır (Şekil 4.39 – 4.40).  
**PD<sup>μ</sup>** kontrolör ile sistem yine 10 sn.' de kararlı hale gelmiştir ve başlangıçtaki taşmalarda oluşmamıştır. Sistemin birim basamak cevabındaki yükselme zamanı kısadır (Şekil 4.42 – 4.43).

$PI^{\lambda}D^{\mu}$  kontrolör sonuçlarına göre sistem yine yaklaşık aynı zamanda kararlı hale gelmiştir. Parametrelerdeki değişimlere karşı gösterilen direnç artmıştır. Sistem transfer fonksiyonunda parametre değiştiğinde sistem birim basamak çıkış cevabında yükselme zamanı çok fazla etkilenmemektedir. Daha geniş bir parametre aralığında kararlılıkla ve bozulmalar olmadan kontrol edilebildiği benzetim sonuçları ile gözlemlenmiştir (Şekil 4.45 – 4.46).

Otomatik ayarlamalı sinir hücresi ile gerçekleştirilen tasarımlarla da kesir dereceli bir PID ( $PI^{\lambda}D^{\mu}$ ) kontrolörün üç parametresinin ayarlanması sağlanmıştır. Buradaki tasarımın diğer tasarımlardan üstünlüğü, parametreyi ayarlayan sinir hücresinin önceden bir eğitime gereksinim duymaması ve modifiye edilmiş geçiş fonksiyonu olan tanjant hiperbolik fonksiyonunu kullanmasıdır. Sinir hücresi geri beslemelerle gerekli en iyi parametre değeri için ayarlamayı kendisi yapmaktadır.

## 5.2. Sonuç

Sistemleri kontrol etmek için kullanılan klasik kontrolörlerin (PID) yerine, kesir dereceli kontrolörlerin ( $PI^{\lambda}D^{\mu}$ ) kullanılması giderek yaygınlaşmaktadır. Klasik PID kontrolörlerde  $K_p$ ,  $K_i$  ve  $K_d$  şeklinde sadece üç tane katsayının hesaplanması ya da ayarlanması, tasarımı ve kullanımı kolaylaştırmaktadır.

$PI^{\lambda}D^{\mu}$  kontrolörde ise  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  gibi beş parametrenin ayarlanması gerekmektedir. Bu parametre sayısındaki artış tasarımcı için zorlaştırıcı bir etkidir. Fakat kesir dereceli sistemler gerçek sistemleri daha iyi modelleyebildikleri için  $PI^{\lambda}D^{\mu}$  kontrolör tasarımları daha iyi çıkış cevabı vermektedir. Bu tez çalışmasında bu zorluk ve sıkıntılar dikkate alınarak önce kesir dereceli kontrol ve kontrolör yapıları incelenmiştir. Sonra bu zorlukların üstesinden gelebilmek için kesir dereceli kontrolör tasarımları için farklı metot ve modellerle farklı tasarımlar üzerinde çalışılmıştır. Bu metotlar sırası ile MRA- $PI^{\lambda}D^{\mu}$ , ANFIS-  $PI^{\lambda}D^{\mu}$  ve Otomatik Ayarlamalı Sinir Hücresi ile  $PI^{\lambda}D^{\mu}$  kontrolör tasarımlarıdır. Bu yöntemlerde hedef çok fazla parametre ile uğraşmadan dayanıklı ve hızlı kontrolörler tasarlamaktır. Bu tezde önerilen yöntemlerin sonuçları değerlendirildiğinde; pratiklik, dayanıklılık ve hız faktörleri için aşağıdaki sonuçlar elde edilir:

**Pratiklik:** Pratik tasarım açısından Auto-Tuning Neuron  $PI^{\lambda}D^{\mu}$  > ANFIS- $PI^{\lambda}D^{\mu}$  > MRA- $PI^{\lambda}D^{\mu}$  şeklinde sıralanabilir. Çünkü Otomatik Ayarlamalı Sinir hücreli tasarımda sinir hücreleri her seferinde farklı sistemler için farklı farklı tasarlanmak zorunda değildir.

Tasarlanan Otomatik Ayarlamalı Sinir hücresi her sistem için çalışabilir. Sinir hücresi geri beslemeler sayesinde parametrelerin optimizasyonunu kendi kendine yapabilmektedir. ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolörünün önceden eğitilmeleri gerekir. Bu uygulama pratiklik açısından ek bir işlem getirir. Eğitim verilerinin elde edilmesi, eğitim süreci zaman gerektiren süreçlerdir. Ayrıca her sistem için bu işlemlerin tekrar edilmesi gerekir. MRA-PI<sup>λ</sup>D<sup>μ</sup> yönteminde öncelikle bir modelin tesbit edilmesi gerekir. Ayrıca birden fazla parametrenin de ayarlanması ve her sistem için farklı model kullanmak gerekebilir.

**Dayanıklılık:** Dayanıklılık açısından ANFIS-PI<sup>λ</sup>D<sup>μ</sup> > MRA-PI<sup>λ</sup>D<sup>μ</sup> > Otomatik Ayarlamalı Sinir Hücresi ile PI<sup>λ</sup>D<sup>μ</sup> kontrolör olarak sıralanabilir. Çünkü ANFIS-PI<sup>λ</sup>D<sup>μ</sup> kontrolör kontrol işlemi esnasında hata işaretinden almış olduğu geri beslemeler sayesinde kontrol için gerekli parametre ayarlamalarını anında gerçekleştirebilmekte ve bu sayede sistemdeki değişimlerden etkilenmeden kontrol işlemi son ana kadar aynı dayanıklılıkla devam etmektedir. Sistem parametrelerindeki değişimden az etkilenen tasarım ise MRA-PI<sup>λ</sup>D<sup>μ</sup> kontrolördür. Sistem modeli iyi seçilirse çevresel etkiler nedeni ile sistemdeki katsayı değişimlerine karşılık, model çıkışı ile sistem çıkışı arasındaki hata oransal, integral ve türev katsayılarını MIT tekniğini kullanarak anında ayarlayabilmektedir.

**Hız:** Bazı benzetim çalışmalarında ANFIS yöntemi kullanan tasarımlar, Otomatik Ayarlamalı Sinir Hücresi kullanan tasarımlardan daha hızlı sonuçlar vermişlerdir. Bunun sebebi, ANFIS bloğu yapısında bulanık mantık ve sinir ağları teorisini birlikte kullanılmasıdır.

Sonuç olarak ANFIS-PI<sup>λ</sup>D<sup>μ</sup>, MRAC-PI<sup>λ</sup>D<sup>μ</sup>, Otomatik Ayarlamalı Sinir Hücresi ile PI<sup>λ</sup>D<sup>μ</sup> kontrolörlerin tam sayılı kontrolörlere göre daha hızlı, dayanıklı ve kararlı cevaplar verdikleri gözlenmiştir.

Bu tasarım çalışmaları, kesir dereceli kontrol sistemlerinin geliştirilmesine katkıda bulunmaktadır. Kontrolör katsayılarının otomatik ayarlanması, kontrol uygulamalarında PI<sup>λ</sup>D<sup>μ</sup> kontrolörünün kullanılmasını kolaylaştırır. Gelecekteki çalışmalar için PI<sup>λ</sup>D<sup>μ</sup> kontrolörün kontrol performansının ve dayanıklılığının optimizasyon algoritmaları ve sinir ağları ile daha da geliştirilebileceğini göstermektedir.

## KAYNAKLAR

- [1] J. G. Ziegler and N. B. Nichols, *Optimum Settings for Automatic Controllers*, **Transactions of the A.S.M.E**, 64 (1942) 759–768.
- [2] K.J. Aström and T. Hägglund, *PID Controllers*, The International Society for Measurement and Control, 1995.
- [3] G. Q. Wang, B. Zou, T. H. Lee, *Auto-Tuning of Multivariable PID Controllers From Decentralized Relay Feedback*, **Automatica**, 33: 3 (1997) 319-330.
- [4] R. Yusof, S. Omatu, *A Multivariable Self-Tuning PID controller*, **Internal Journal of Control**, 57:6 (1993) 1387-1403.
- [5] A. Oustaloup, P. Melchior, P. Lanusse, O. Cois and F. Dancla, *The CRONE Toolbox for Matlab*, in Proceedings of 11th IEEE International Symposium on Computer-Aided Control System Design, CACSD, Anchorage, Alaska, USA, Sept. 25-27, (2000), pp. 190-195.
- [6] I. Podlubny, *Fractional-Order Systems and  $PI^{\lambda}D^{\mu}$  Controllers*, **IEEE Trans. Automatic Control**, 44 (1999) 208–214.
- [7] I. Petras, *Control Quality Enhancement by Fractional Order Controllers*, *Acta Montanistica Slovaca*, 3:2, (1998) 143-148.
- [8] C. Ma, *Introduction of Fractional Order Control and Its Applications Motion Control*, PhD Thesis, The University of Tokyo Japan, 2004.
- [9] C. A. Monje, A. J. Calderon, B. M. Vinagre, Y. Chen and V. Feliu, *On Fractional  $PI^{\lambda}$  Controllers: Some Tuning Rules for Robustness to Plant Uncertainties*, **Nonlinear Dynamics**, 38:1-2 (2004) 369-381.
- [10] M. R. Kumar, V. Deepak, S. Ghosh, *Fractional-Order Controller Design in Frequency Domain Using an Improved Nonlinear Adaptive Seeker Optimization Algorithm*, **Turkish Journal of Electrical Engineering & Computer Sciences**, 25 (2017) 4299-4310.
- [11] H. Senberber and A. Bagis, *Fractional PID Controller Design for Fractional Order Systems Using ABC Algorithm*, *Electronics*, Palanga, June 19-21, (2017), pp. 1-7.
- [12] A. Tepljakov, *Fractional-Order PID Controller Design*, In: *Fractional-order Modeling and Control of Dynamic Systems*, Springer Theses, Springer, Cham, (2017) 47-76.

- [13] M. Al-Dhaifallah, N. Kanagaraj, and K. S. Nisar, *Fuzzy Fractional-Order PID Controller for Fractional Model of Pneumatic Pressure System*, **Mathematical Problems in Engineering**, vol. 2018, Article ID 5478781, (2018) 9 pages.
- [14] C. Ma and Y. Hori, *Fractional Order Control: Theory and Application in Motion Control*, **IEEE Industrial Electronics Magazine**, 1 (2007) 6 -16.
- [15] M.F. Silva, J.A.T. Machado and A.M. Lopes, *Fractional Order Control of a Hexapod Robot*, **Nonlinear Dynamics**, 38 (2004) 417-433.
- [16] D. Xue, C. Zhao and Y.Q. Chen, *Fractional Order PID Control of A DC-Motor with Elastic Shaft: A Case Study*, Proceedings of the 2006 American Control Conference, Minneapolis, Minnesota, USA, June 14-16, (2006) pp. 6-10.
- [17] S. Manabe, *A Suggestion of Fractional-Order Controller for Flexible Spacecraft Attitude Control*, **Nonlinear Dynamics**, 29 (2002) 251–268.
- [18] W. Xifan, C. Chengjun and Z. Zhichao, *Experiment on Fractional Frequency Transmission System*, **IEEE Transactions on Power Systems**, 21: 1 (2006) 372-377.
- [19] W. Ahmad, R. El-khazali and A.S. Elwakil, *Fractional-Order Wien-Bridge Oscillator*, **Electronics Letters**, 37:18 (2001) 1110-1112.
- [20] W. Ahmad, *Power Factor Correction Using Fractional Capacitors*, Proceedings of the 2003 International Symposium on Circuits and Systems, ISCAS '03., Thailand Bangkok, May 25-28, (2003), pp. III-5 –III-7.
- [21] J. Liang and Y.Q. Chen, *Fractional-Order Boundary Control of Fractional Wave Equation with Delayed Boundary Measurement Using Smith Predictor*, In Proceedings of the IEEE Conference on Decision and Control, Vol. 5, (2004), pp. 5088-5093.
- [22] C. A. Monje, *Design Methods of Fractional Order Controllers for Industrial Applications*, Ph. D Thesis, University of Extremadura, Badajoz, Spain, March 2006.
- [23] J. F. Leu, S. Y., Tsay, and C.Hway, *Design of Optimal Fractional Order PID Controllers*, **Journal of the Chinese Institute of Chemical Engineers**, 33: 2 (2002) 193–202.
- [24] D.Valério, J. M. G. Sá da Costa, *Tuning-Rules for Fractional PID Controllers*, FDA06 - 2<sup>nd</sup> IFAC Workshop on Fractional Differentiation and its Applications, Porto, June 2007.

- [25] R. Ranganayakulu, G. U. B. Babu, A. S. Rao, D. S. Patle, *A Comparative Study of Fractional Order  $PI^\lambda/PID^\mu$  Tuning Rules for Stable First Order Plus Time Delay Processes*, **Resource-Efficient Technologies**, 2:1 (2016) 136-152.
- [26] D. Singh and P. K. Padhy, *Design and Tuning of  $PI^\lambda D^\mu$  Controller for Higher Order Process Using PSO-NM*, 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, (2017), pp.1-5.
- [27] D. Xue and Y. Q. Chen, *A Comparative Introduction of Four Fractional Order Controllers*, Proc the 4th World Congress, Intelligent Control and Auto. 02, Vol.4, (2002), pp. 3228-3235.
- [28] J. Cao, J. Liang, and B-G. Cao, *Optimization of Fractional Order PID Controllers Based on Genetics Algorithms*, in Proceedings of the Fourth International Conference on Machine Learning and Cybernetics 9, Guangzhou, August 18 – 21, (2005), pp. 5686 – 5689.
- [29] B. Vinagre, I. Podlubny, L. Dorcak, and V. Feliu, *On Fractional PID Controllers: A Frequency Domain Approach*, in IFAC Workshop on Digital Control. Past, Present, and Future of PID Control, Terrasa, Spain, (2000), pp. 53–58.
- [30] G. Maione and P. Lino, *New Tuning Rules for Fractional  $PI^\alpha$  controllers*, **Nonlinear Dynamics**, 49 (2007) 251 - 257.
- [31] C. A. Monje, B. M. Vinagre, V. Feliu and Y.Q. Chen, *Tuning and Auto-Tuning of Fractional Order Controllers for Industry Applications*, **Control Engineering Practice**, 16 (2008) 798–812.
- [32] K. Bettou, A. Charef and F. Mesquine, *A New Design Method for Fractional  $PI^\lambda D^\mu$  Controller*, *IJ-STA*, 2 (2008) 414-429.
- [33] M. Zhuang and D. P. Atherton, *Automatic Tuning of Optimum PID Controllers*, **IEE Proc. Part D**, 140 (1993) 216-224.
- [34] Y. T. Li, *Optimizing System for Process Control*, **Instruments**, 25 (1952) 72-77, 324.
- [35] K. J. Astrom and B. Wiltenmark, *Adaptive Control*, Addison-Wesley, MA, 1995.
- [36] Y. T. Li, and W. E. Vander Velde, *Philosophy of Nonlinear Adaptive Systems*, Automatic and Remote Control (Proc. Of the First IFAC Congr.), Butterworth Scientific Publications, London, Vol II, (1961), pp. 577 -585.

- [37] K. J. Åström and B. Wittenmark, *On Self-Tuning Regulators*, **Automatica** 9, (1973) 185–199.
- [38] R. V. Monopoli, *Model Reference Adaptive Control with an Augmented Error*, **IEEE Transactions on Automatic Control**, AC-19 (1974) 474 - 484.
- [39] Y. D. Landau, *Adaptive Control: The Model Reference Approach*, Marcel Dekker, New York, 1979.
- [40] K. S. Narendra, Y. H. Lin and L. S. Valavani, *Stable Adaptive Controller Design, Part II: Proof of Stability*, **IEEE Transactions on Automatic Control**, 25:3 (1980) 440 - 448.
- [41] C. A. Canudas de Wit, *Adaptive Control for Partially Known Systems: Theory and Applications*, Elsevier Science, New York, 1988.
- [42] S. Sastry and M. Bodson, *Adaptive Control, Stability, Convergence and Robustness*, Prentice-Hall, Inc., 1989.
- [43] G. Ambrosino, G. Celentano and F. Garofalo, *Variable Structure Model Reference Adaptive Control Systems*, **Int. J. Control**, 39 (1984) 1339-1349.
- [44] S. Ladaci and A. Charef, *On Fractional Adaptive Control*, **Nonlinear Dynamics** 43 (2006) 365 – 378.
- [45] N. Aguila-Camacho, M. A. Duarte-Mermoud, *Combined Fractional Adaptive Control*, **IFAC-PapersOnLine**, 50:1 (2017) 8586-8591.
- [46] L. Zadeh, *Fuzzy Sets*, *Inf. Control* , 8, 1965, p.338–353.
- [47] E. H. Mamdani, and S. Assilian, *An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller*, **International Journal of Man-Machine Studies**, 7:1, (1975) 1-13.
- [48] W.S. McCulloch, and W.Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, **Bulletin of Mathematical Biophysics**, 5 (1943) 115-133.
- [49] D. Hebb, *The Organization of Behavior. A Neuropsychological Theory*, Wiley, New York, NY, 1949.
- [50] F. Rosenblatt, *The perceptron: a Probabilistic Model for Information Storage and Organization in the Brain*, **Psychol Rev.**, Nov, 65:6 (1958) 386-408.
- [51] B. Widrow and M. E. Hoff, *Associative Storage and Retrieval of Digital Information in Networks of Adaptive ‘Neurons’*, *Biological Prototypes and Synthetic Systems*, 1962, 1:160.
- [52] M. Minsky and S. Papert, *A Review of “Perceptrons: An Introduction to Computational Geometry*, **Information and Control**, 17 (1970) 501-522.



- [53] M. A. Cohen and S. Grossberg, *Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks*, **IEEE Trans. SMC**, 13 (1983) 815-826.
- [54] S. –I. Amari, *Characteristics of Randomly Connected Threshold-Element Networks and Network Systems*, **Proc. of the IEEE**, 59:1 (1971) 35-47.
- [55] K. Fukushima, *Neocognitron: a Hierarchical Neural Network Capable of Visual Pattern Recognition*, **Neural Networks**, 1:2 (1988) 119-130.
- [56] T. Kohonen, *Neural Networks*, Supply. 1, 303, 1988
- [57] T. Kohonen, *Self-Organization and Associative Memory*, (3rd edition 1989). Springer, Berlin, 1984.
- [58] J. G. Taylor, *The Promise of Neural Networks*, Springer-Verlag, 1992.
- [59] J. J. Hopfield and D. W. Tank, *Neural' Computation of Decisions Optimization Problems*, **Biological Cybernetics**, 52 (1985) 141-152.
- [60] D. E. Rumelhart, G. E. Hinton and R. J. Williams, *Learning Internal Representations by Error Propagation*, **Nature**, 323 (1986) 533-536.
- [61] L. O. Chua and L. Yang, *Cellular Neural Networks: Theory*, **Circuits and Systems, IEEE Transactions**, 35 (1988) 1257-1272.
- [62] I. CAN, *Yapay Sinir Ağları ve Robot Denetim Uygulamaları*, **Elektrik Mühendisliği EMO, Ankara**, 393 (1995) 19-24.
- [63] M. Amiri, M. B. Menhaj and M. J. Yazdanpah, *A Neural-Network Based Controller for a Single-Link Flexible Manipulator: Comparison of FFNN and DRNN Controllers*, In: IEEE International Joint Conference on Neural Networks, IJCNN (IEEE World Congress on Computational Intelligence), (2008), pp.1686-1691.
- [64] N. Tschichold-Gürman, *RuleNet - A New Knowledge-Based Artificial Neural Network Model with Application Examples in Robotics*, PhD Thesis, ETH Zürich Swiss, 1996.
- [65] P. Song, G. Qi and K. Li, *The Flight Control System Based on Multivariable PID Neural Network for Small-Scale Unmanned Helicopter*, In: Proceedings of the international conference on information technology and computer science, Kiev, Ukraine, (2009), pp. 538-541.
- [66] S. N. Engin, J. Kuvulmaz ve V. E. Ömürlü, *Fuzzy Control of an ANFIS Model Representing a Nonlinear Liquid-Level System*, **Neural Computing & Applications**, 13: 3 (2004) 202–210.

- [67] M. Khalid and S. Omatu, *A Neural Network Controller for Temperature Control System*, **IEEE Control System Magazine**, 12: 3 (1992) 58 – 64.
- [68] R. Hernández-Alvarado, L. G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa and F. Fonseca-Navarro, *Neural Network-Based Self-Tuning PID Control for Underwater Vehicles*, **Reindl LM, ed. Sensors (Basel, Switzerland)**, 16:9 (2016):1429 .
- [69] M. S. Ahmed, A. Mohamed, H. Shareef, R. Z. Homod and J. A. Ali, *Artificial Neural Network Based Controller for Home Energy Management Considering Demand Response Events*, International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEEES), Putrajaya, (2016), pp. 506-509.
- [70] D. Valério and J. S.Costa, *Tuning of Fractional PID Controllers with Zeigler-Nichols-Type Rules*, **Signal Processing**, 86 (2006) 2771–2784.
- [71] R. Caponetto, L. Fortuna and D. Porto, *Parameter Tuning of a Non Integer Order PID Controller*, In Proceedings of the Fifteenth International Symposium on Mathematical Theory of Networks and Systems, Notre Dame, Indiana 2002.
- [72] C. A. Monje, B. M.Vinagre, Y. Q. Chen, V. Feliu, P. Lanusse and J. Sabtier, *Optimal Tunings for Fractional  $PI^{\lambda}D^{\mu}$  Controllers*, **Fractional Differentiation and its Applications**, (2009) 675-686.
- [73] C. Zhao, D. Xue and Y. Q. Chen, *A Fractional Order PID Tuning Algorithm for A Class of Fractional Order Plants*, IEEE International Conference on Mechatronics and Automation. 1. , Niagara Falls, Canada, July 2005, pp. 216 - 221.
- [74] V. Kumar, K. P. S. Rana and P. Mishra, *Robust Speed Control of Hybrid Electric Vehicle Using Fractional Order Fuzzy PD & PI Controller in Cascade Control Loop*, **Journal of Franklin Institute**, 353:8 (2016) 1713-1741.
- [75] H. A. Malki, H. Li and G. Chen, *New Design and Stability Analysis of Fuzzy Proportional-Derivative Control Systems*, **IEEE Trans. Fuzzy Syst.**, 2:4 (1994) 245-254.
- [76] S. Das, I. Pan and S. Das, *Performance Comparison of Optimal Fractional Order Hybrid Fuzzy PID Controllers for Handling Oscillatory Fractional Order Processes with Dead Time*, *ISA Trans.*, 52 :4 (2013) 550-566.
- [77] N. Sadati, A. Ghaffarkhah and S. Ostadabbas, *A New Neural Network Based FOPID Controller*, In: IEEE International Conference on Networking, Sensing and Control, IEEE Transactions on Nuclear Science 57, (2010), pp. 1602-1612.

- [78] Y. Hou, J. Zhang, Y. Zhang, L. Chen, *Integrated Chassis Control Using ANFIS*, Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September 2008, pp. 1625-1630.
- [79] J. Carvajal, G. Chen and H. Ogma, *Fuzzy PID Controller: Design, Performance, Evaluation and Stability Analysis*, **Int. J. Inform. Sci**, 123 (2000) 249-270.
- [80] T. Emani and J. M. Watking, *Robust Stability Design of PID Controller for Arbitrary-Order Transfer Function with Uncertain Time Delay*, In: South-eastern symposium on system theory, University of Tennessee Space Institute, (2009), pp. 184-189.
- [81] T. Emani and J. M. Watking, *Robust Performance Characterization of PID Controllers in the Frequency Domain*, **WSEAS Trans J Syst. Control** 4:5 (2009) 232-242.
- [82] S.P. Bhattacharyya, H. Chapellat and L.H. Keel, *Robust Control: The Parametric Approach*, Prentice Hall, 1995.
- [83] J. Wang and Y. Li, *Frequency Domain Stability Criteria for Fractional- Order Control*, **Journal of Chongqing University**, 5:1 (2006) 30-35.
- [84] T. Amieur, A. Younsi, M. Aidoud, M. Sedraoui and O. Amieur, *Design of Robust Fractional Order PID Controller Using Fractional Weights in the Mixed Sensitivity Problem*, 14th International Multi-Conference on Systems, Signals & Devices (SSD), Marrakech, (2017), pp. 549-553.
- [85] H. Hamideh and B. Mohammad, *A Robust Fractional-Order PID Controller Design Based on Active Queue Management for TCP Network*. **International Journal of Systems Science**, 49 (2017) 1-6.
- [86] E.-H. Dulf, D. Timis, C.-I. Muresan, *Robust Fractional Order Controllers for Distributed Systems*, **Acta Polytechnica Hungarica** 14:1 (2017) 163-176.
- [87] M. Talebpour, Y.M. Roshan, S. Mohseni, *Developing Robust FFOPID Controllers Based on Fuzzy Set Point Weighting Algorithm*, **Fractional Calculus & Applied Analysis**, 12:4 (2009) 373-390.
- [88] R. Gong, Y. Yang, Y. Tian, G. Zhang, J. Sun and L. Chen, *Design and Implement of Neural Network Based Fractional Order  $PI^\alpha$  Controller*, Sixth International Conference on Natural Computation, Yantai, (2010), pp. 215-219.
- [89] W. Shan and W. Tang, *A Neural Network Fractional Order PID Controller for FOLPD Process*, 35th Chinese Control Conference (CCC), Chengdu, (2016), pp. 10459-10463.

- [90] I. Petras, L. Dorcak, P. O’Leary, B. M. Vinagre and I. Podlubny, *The Modelling and Analysis of Fractional-Order Control Systems in Frequency Domain*, In Proceeding of ICC’2000 (High Tatras, Slovak Republic), May 23-26, (2000), pp. 261-264.
- [91] R. Hilfer, *Applications of Fractional Calculus in Physics*, World Scientific Publishing Company, Singapore , 2000, 87 – 130.
- [92] F. Nonnenmacher ve W.G. Glöckle, *A fractional model for mechanical stress Relaxation*, **Philosophical Magazine Lett.**, 64:2 (1991) 89-93.
- [93] C. T. Chen, W. D. Chang, *A feedforward neural network with function shape Autotuning*, **Neural Networks**, 9:4 (1996) 627-641.
- [94] Z. Hu, J. Zhou, D. Huang, *A multivariable nonlinear self-tuning PID controller*, **Control Theory and Application**, 13:2 (1996) 268-271.
- [95] Z. Li, D. Xiao, S. He, *A fuzzy adaptive PID controller based on neural networks*, **Control and Decision**, 8:3 (1996) 340-345.
- [96] S. Omatu, M. Khalid, R. Yusof, *Neuro-Control and its Applications*, Springer, London, 1995, 85 -170.
- [97] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*, **John Wiley & Sons**, New York, NY, USA; 1993.
- [98] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, California, 1999.
- [99] G. Haung and S. Lee, *PC Based PID Speed Control of DC Motor*, In Proceeding of International Conference on Audio, Language and Image Processing (ICALIP), (2008), pp. 400-407.
- [100] A. L. Salih, M. Moghavvemi, H. A. F. Mohamed and K.S. Gaeid, *Modelling and PID Controller Design for a Quadrotor Unmanned Air Vehicle*, IEEE Int. Conf. on Automation Quality and Testing Robotics (AQTR), 1, (2010), pp. 1–5.
- [101] T. J. Ross, *Fuzzy Logic with Engineering Applications*, Third Edition, John Wiley & Sons, Ltd., 2010, 245 260, 437 - 484.
- [102] Z. Şen, *Bulanık (Fuzzy) Mantık ve Modelleme İlkeleri*, Bilge Sanat Yapım Yayınları, İstanbul, 2002.
- [103] G. J. Klir, B.Yuan, *Fuzzy Sets and Fuzzy Logic Theory and Applications*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1995, 418 -440, 443 -463.

- [104] E. H. Mamdani, *Application of fuzzy algorithm for control a simple dynamic plant*, Proc. IEEE 121, (1974), pp. 1585-1588.
- [105] T. Takagi and M. Sugeno, *Fuzzy Identification of Systems and its Applications to Modeling and Control*, **IEEE Trans. Syst., Man, Cybern.**, SMC-15 (1985) 116-132.
- [106] J. S. R. Jang, *ANFIS: Adaptive-Network-Based Fuzzy Inference Systems*, **IEEE Trans. Systems, Man & Cybernetics**, 23:03 (1993) 665-685.
- [107] L. V. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1994, 156 -218, 289 -334.
- [108] T. Yoshiyasu, *Neural Network Parallel Computing*, Springer Science & Business Media, 1992.
- [109] N. K. Base, P. Liang *Neural Network Fundamentals with Graphs, Algorithms and Applications*, McGraw-Hill Education, 1995.
- [110] G. Kevin, *An Introduction to Neural Networks*, CRC Press, 1997, 13 -16, 102 - 103.
- [111] H. H. Mohamad, *Fundamentals of Artificial Neural Networks*, The MIT Press, Cambridge, MA, USA, 1995.
- [112] J. G. Taylor, *Neural Network Applications*, Springer, 1992.
- [113] S. S. Ge. and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*, World Scientific Publishing Company, 1998.
- [114] S. Huang, *Neural Network Control: Theory and Application*, Research Studies Press Limited, 2004.
- [115] S. K. Halgamuge and M. Glesner, *Neural networks in designing fuzzy systems for real world applications*, **Fuzzy Sets and Systems**, 65 (1994) 1-12.
- [116] R. P. Lipmann, *An introduction to computing with neural networks*, **IEEE ASSP Mag.**, 4 (1987) 4-22.
- [117] M.Cinsdikici, A.C.Yüçetürk, Y.Öztürk, *ATM Networkleri için Yapay Sinir Ağı Çözümleri*, 14. Ulusal Bilişim Kurultayı (BİLİŞİM.97) Bildirileri, İstanbul, Eylül 3 -6, (1997), pp. 55 – 59.
- [118] S. Haykin, *Neural Networks and Learning Machines*, Pearson Education, Upper Saddle River, NJ, Third edition, 2009, 1 – 46.
- [119] E. Öztemel, *Yapay Sinir Ağları*, 2. Baskı, İstanbul, Papatya Yayıncılık, 2006.

- [120] K.Turkoğlu, *Bir Uçağın Kararlılık Analizi Ve Otomatik Kontrol Sistemi Tasarımları: Uyarlamalı (adaptif) Kontrol Yaklaşımı*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Türkiye, 2007.
- [121] E. Demirtaş, *F-16 Uçağının Model Referans Adaptif Ve Bulanık Kontrolü*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Türkiye, 2001.
- [122] B. M. Vinagre, I. Petras, I. Podlubny and Y. Q. Chen, *Using Fractional Order Adjustment Rules and Fractional Order Reference Models in Model Reference Adaptive Control*, **Nonlinear Dynamics** 29 (2002) 269–279.
- [123] H. Arpacı, Ö. F. Özgüven, M. Serhat, *Fractional Model Reference Adaptive  $PI^{\lambda}D^{\mu}$  Control*, **Journal of New Results in Science**, 11 (2016) 102-117.
- [124] J. J. Buckley and Y. Hayashi, *Fuzzy neural networks: A survey*, **Fuzzy Sets and Systems**, 66 (1994)1-13.
- [125] J. J. Buckley and Y. Hayashi, *Neural networks for fuzzy systems*, **Fuzzy Sets and Systems**, 71 (1995) 265-276.
- [126] D. Nauck, F. Klawonn and R. Kruse, *Foundations of Neuro-Fuzzy Systems*, Wiley, New York, 1997.
- [127] D. Nauck and R. Kruse, *Designing neuro-fuzzy systems through Backpropagation*, In: W. Pedrycz, Ed., *Fuzzy Modelling: Paradigms and Practice*, Kluwer, Boston, (1996), pp. 203-228.
- [128] C. C. Lee, *Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part 1-2*, **IEEE Transactions on Systems, Man and Cybernetics**, 20:2 (1990) 404-435.
- [129] T. Takagi, M. Sugeno, *Derivation of fuzzy control rules from human operator's control actions*, In Proc. IFAC Symp. Fuzzy Inform., Knowledge Representation and Decision Analysis, (1983), pp. 55-60.
- [130] H. Arpacı, Ö. F. Özgüven, *Design of Adaptive Fractional Order PID Controller to Enhanced Robustness by Means of Adaptive Network Fuzzy Inference System*, **Int. J. Fuzzy Syst.**, 19:4 (2017) 1118–1131.
- [131] B. M. Vinagre, Y. Q. Chen, I. Petras, *Two Direct Tustin Discretization Methods for Fractional-Order Differentiator/Integrator*, **Journal of The Franklin Institute**, 340 (2003) 349-362.
- [132] Y.Q. Chen, K.L. Moore, *Discretization schemes for fractional order differentiators and integrators*, **IEEE Trans. Circuits Systems-I Fund. Theory Appl.**, 49:3 (2002) 363-367.

- [133] W. D. Chang, R. C. Hwang, J. G. Hsieh, *Adaptive control of multivariable dynamic systems using independent self-tuning neurons*, In Proceeding of the Tenth International Conference on Tools with Artificial Intelligence, Taipei, Taiwan, (1998), pp. 68-73.
- [134] H. Arpacı, Ö. F. Özgüven, M. Serhat, *Otomatik Ayarlamalı Sinir Hücresi ile Adaptif Kesir Dereceli PID Kontrolör Tasarımı*, **Journal of New Results in Engineering and Natural Science**, 8 (2018) 100-118.
- [135] A. Oustaloup, F. Levron, B. Mathieu and F. M. Nanot, *Frequency-Band Complex Noninteger Differentiator: Characterization and Synthesis*, **IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications**, 47:1 (2000) 25 - 39.
- [136] Y.Q. Chen, *Oustaloup Recursive Approximation for Fractional Order Differentiators*, MathWorks, Inc. Matlab Central File Exchange, 2003.
- [137] Anonymous. (2016). <http://ctms.engin.umich.edu/CTMS/index.php?example = Motor Speed&Section = SystemModeling> (on - line accessed in 2016).
- [138] Anonymous. (2016). [http://www.mathworks.com/matlabcentral/fileexchange/36508-fuzzy-controller-based-speed\\_control-of-dc-motor/all\\_files](http://www.mathworks.com/matlabcentral/fileexchange/36508-fuzzy-controller-based-speed_control-of-dc-motor/all_files) (on - line accessed in 2016).
- [139] J. J. Huang and D.B.DeBra, *Liquid Temperature Control for a Hydraulic Turning Machine*, **IEEE Control Syst.** 17:4 (1997) 55–63.
- [140] A. V. Duka, S. E Oltean, *Fuzzy control of a heat exchanger*, In: 2012 IEEE International Conference on Automation Quality and Testing Robotics (AQTR), May 24–27 , (2012), pp. 135–139.
- [141] V. Feliu, L. Gorostiaga, B. M. Vinagra and C. A. Monje, *Robust Smith Predictor for First Order Proceses with Dead Time Based on a Fractional Controller*, Internal report, 2002.
- [142] K. Ogata, *Modern Control Engineering*, Prentice Hall, New Jersey, 2002.

## ÖZGEÇMİŞ

**Ad Soyad:** Hüseyin ARPACI

**Doğum Yeri ve Tarihi:** Malatya 19.08.1974

**Adres:** Şeyh Bayram Mah. Lavinya Sitesi Kat 5 No:17

MALATYA

**E-Posta:** huseyin.arpaci@inonu.edu.tr

**Lisans:** Orta Doğu Teknik Üniversitesi – Elektrik Elektronik Mühendisliği

**Yüksek Lisans:** İnönü Üniversitesi – Elektrik Elektronik Mühendisliği Anabilim Dalı

**Mesleki Deneyim ve Ödüller:** 1998 – 1999 yıllarında bilgisayar ve ingilizce öğretmeni olarak çalıştı. 2000 – 2001 yılları arasında askerlik görevini tamamladı. 2001 - 2010 yılları arasında İnönü Üniversitesi Enformatik Bölüm Başkanlığında bilgisayar ve bilişim teknolojileri Öğretim Görevlisi olarak görev yaptı. 2010 yılında İnönü Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümüne geçiş yaptı ve halen buradaki görevine devam etmektedir.

### Tezden Türetilen Yayınlar

1. **Arpacı, H.**, Özgüven, F.Ö., “Design of Adaptive Fractional Order PID Controller to Enhanced Robustness by Means of Adaptive Network Fuzzy Inference System”, Int. J. Fuzzy Syst., 19(4):1118-1131, 2017.
2. **Arpacı, H.**, Özgüven, F.Ö., “ANFIS & PI<sup>λ</sup>D<sup>μ</sup> controller design and comparison for overhead cranes”, Indian Journal of Engineering & Materials Sciences Vol. 18, June 2011, pp. 191-203
3. **Arpacı H.**, Özgüven, Ö. F., Serhat, M., “Fractional Model Reference Adaptive PI<sup>λ</sup>D<sup>μ</sup> Control”, Journal of New Results in Science, Number 11, 102-117, 2016.



4. **Arpacı, H., Özgüven, Ö. F., Serhat, M.,** “Otomatik Ayarlamalı Sinir Hücresi ile Adaptif Kesir Dereceli PID Kontrolör Tasarımı”, Journal of New Results in Engineering and Natural Science, No:8, 100-118, 2018.
5. **Arpacı, H., Özgüven, F.Ö.,** “MRAC ile Kesir Dereceli PID Kontrolör Tasarımı”, TOK2011 Ulusal Otomatik Kontrol Kongresi, 14-16 Eylül 2011, pp. 2-7, İzmir, Türkiye.
6. **Arpacı, H., Özgüven, F.Ö.,** “ANFIS ile Kesir Dereceli PID Kontrolör Tasarımı”, TOK2011 Ulusal Otomatik Kontrol Kongresi, 14-16 Eylül 2011, pp. 8-13, İzmir, Türkiye.
7. **Arpacı, H., Özgüven, Ö. F., Serhat, M.,** “Otomatik Ayarlamalı Sinir Hücresi ile Adaptif Kesir Dereceli PID Kontrolör Tasarımı”, ISMSIT 20171st International Symposium on Multidisciplinary Studies and Innovative Technologies November 2-4, 2017 Gaziosmanpaşa University, Tokat, Turkey.