

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

**SEZGİSEL ALGORİTMALARLA ÇOKLU DENETÇİ PARAMETRELERİNİN  
OPTİMİZASYONU**

**ABDULLAH ATEŞ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**  
**DOKTORA TEZİ**

**EKİM 2018**

**Tezin Başıđı: Sezgisel Algoritmalarla Çoklu Denetçi Parametrelerinin Optimizasyonu**

Tezi Hazırlayan: Abdullah ATEŞ

Sınav Tarihi: 19/10/2018

Yukarıda adı geçen tez jürimizce değerlendirilerek Bilgisayar Mühendisliđi Ana Bilim Dalında Doktora Tezi olarak kabul edilmiştir.

**Sınav Jürisi Üyeleri**

<b>Tez Danışmanı :</b>	<b>Doç. Dr. Celaleddin YEROĞLU</b>	.....
	İnönü Üniversitesi	
	<b>Prof. Dr. Ali KARCI</b>	.....
	İnönü Üniversitesi	
	<b>Prof. Dr. Hafız ALİSOY</b>	.....
	Tekirdađ Namık Kemal Üniversitesi	
	<b>Doç. Dr. Fevzi HANSU</b>	.....
	Siirt Üniversitesi	
	<b>Dr. Öğr. Üyesi Bilal ŞENOL</b>	.....
	İnönü Üniversitesi	

**Prof. Dr. Halil İbrahim ADIGÜZEL**

Enstitü Müdürü

## ONUR SÖZÜ

Doktora tezi olarak sunduđum “**Sezgisel Algoritmalarla Çoklu Denetçi Parametrelerinin Optimizasyonu**” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla doğrularım.

Abdullah ATEŞ

# ÖZET

Doktora Tezi

## SEZGİSEL ALGORİTMALARLA ÇOKLU DENETÇİ PARAMETRELERİNİN OPTİMİZASYONU

Abdullah ATEŞ

İnönü Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

172+XV sayfa

2018

Danışman: Doç. Dr. Celaleddin YEROĞLU

Mühendislik problemlerinin çözümünde özellikle denetçi katsayılarının belirlenmesinde nümerik optimizasyon yöntemleri tercih edilmektedir. Bu tez çalışmasında nümerik optimizasyon yöntemleriyle birçok sistem için denetçiler tasarlanmıştır. Öncelikle kesir dereceli sistemler Homotopy Pertürbasyon Yöntemi (HPY) ile çözümlenmiştir. Daha sonra birçok sistem için kesir dereceli ve tam sayı dereceli denetçi parametrelerini belirleyebilen bir yöntem olan Rassal Parametre Vektör Optimizasyon (RPVO) yöntemi önerilmiştir. Bu yöntem ile oluşturulan denetçiler deneysel sistemler üzerinde kullanılmış ve elde edilen cevaplar karşılaştırmalı olarak sunulmuştur. Tezin takip eden bölümünde, kesir dereceli denetçi parametrelerinin Tabu Arama Algoritması (TAA) ile ayarlanması gösterilmiştir. Elde edilen sonuçlar literatürdeki yöntemlerle karşılaştırılmıştır. Daha sonra literatürde olan fakat denetçi tasarımında kullanılmamış Yapay Fizik Optimizasyon Algoritması (YFOA) yöntemi, denetçi tasarımı için modifiye edilmiş ve bilgisayar destekli tasarım için kodlanmıştır. Böylece, literatürde olan matematiksel modeller için tam sayı dereceli ve kesir dereceli denetçiler tasarlanmıştır. Ayrıca bu algoritmanın çalışma mekanizmasına etki eden yapılar optimizasyon süreçlerinde ayrı ayrı incelenerek algoritmanın ilgili sistem için maksimum performans ile çalışabileceği konfigürasyon belirlenmiştir. Genellikle, optimizasyon süreçlerinde tekli amaç fonksiyonu kullanılır. Fakat bu tez çalışmasının son bölümünde çoklu amaç fonksiyonları kullanılarak optimizasyon sürecinin gerçekleştirilebileceği, Büyük Patlama Büyük Çöküş (BP-BÇ) Optimizasyon algoritması ve Tabu Arama Algoritması (TAA) kullanılarak gösterilmiştir.

**Anahtar Kelimeler:** Optimizasyon, kontrol, kesir dereceli denetçi, tam sayı dereceli denetçi sezgisel, stokastik, rassal, analitik, nümerik, çoklu amaç fonksiyonu, tekli amaç fonksiyonu.

## ABSTRACT

Phd. Thesis

### OPTIMIZATION OF MULTIPLE CONTROLLER PARAMETERS WITH HEURISTIC ALGORITHM

Abdullah ATEŞ

Inonu University  
Graduate School of Nature and Applied Sciences  
Department of Computer Engineering

172+XV sayfa

2018

Supervisor: Assoc. Prof. Dr. Celaleddin YEROĞLU

Numerical optimization methods are preferred for solving engineering problems particularly for determination of the controller parameters. In this thesis, many controllers were designed with numerical optimization method for many systems. Firstly, fractional order mathematical equations were solved with Homotopy Perturbation Method (HPM). Then, a stochastic multi parameter divergence optimization method (SMDO), which can determine fractional order and integer order controller parameters for the many systems, was proposed. Controllers, which were obtained by means of this method, was used for experimental systems and obtained responses were presented comparatively. In the following part of the thesis, tuning of fractional order controller parameters by Tabu Search Algorithm (TSA) was demonstrated. The results were compared with methods in literature. Then, Artificial Physics Optimization (APO) Algorithm method, which exists in literature but not used in controller design, was modified for controller design and coded for computer aided design. Thus, integer order and fractional order controller were designed for mathematical models that exist in the literature. Moreover, the structures, which affects the working mechanism of this algorithm, are separately examined in the optimization processes, and the configuration, where the algorithm can work with the maximum performance for the related system, was determined. In general, single objective function is used in optimization processes. However, in the last part of this thesis, Big Bang Big Crunch (BB-BC) Optimization Algorithm and Tabu Search Algorithm (TSA) were used to show that the optimization process can be performed by using multiple objective functions.

**Keywords:** Optimization, control, fractional order controller, integer order controller, heuristic, stochastic, random, analytical, numerical, multi objective function, single objective function.

## TEŐEKKÜR

Bu tez alıőmasının her aőamasında yardım, öneri, bilgi, tecrübe ve desteklerini esirgemedен beni her konuda yönlendiren danışman hocam Sayın Do. Dr. Celaleddin YEROĐLU'na; alıőmalarımда beni her konuda destekleyen ve beraber alıőmalarda bulunduĐum Sayın Hocam Dr. Öğretim Üyesi Barıő Baykant ALAGÖZ'e, alıőmalarım esnasında bilgi ve tecrübelerinden her fırsatta faydalandıĐım Sayın hocam Dr. Öğretim Üyesi Bilal ŐENOL'a, tez süresince beraber alıőmalarda bulunduĐum Sayın hocam Arő. Gör. Dr. Gürkan Kavuran'a, Sayın hocam Prof. Dr. Ali Karcı'ya, alıőmalarım esnasında laboratuvarında alıőmıő olduĐum University of California Merced'den Sayın Hocam Prof. Dr. YangQuan Chen'e; ayrıca tüm hayatım boyunca olduĐu gibi bu alıőmalarım süresince de benden her türlü desteklerini esirgemeyen eőim Rahmiye Nur Ateő'e, kızım Mihrinisa Ateő'e , annem Asiye Ateő'e, babam Ahmet Ateő'e, kardeőim Fatih Ateő'e, dayılarım Zeki Sara'a, Nevzat Sara'a, Nusret Sara'a ve Mustafa Sara'a ayrıca tezin uygulama aőamasında vermiő oldukları maddi ve manevi destekten dolayı, TÜBİTAK Bilim İnsanı Destekleme Birimi 2214/A Doktora sırası araştırma burs programına

teőekkür ederim.

## İÇİNDEKİLER

<b>ÖZET</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>TEŞEKKÜR</b> .....	<b>iii</b>
<b>İÇİNDEKİLER</b> .....	<b>iv</b>
<b>SİMGELER VE KISALTMA DİZİNİ</b> .....	<b>ix</b>
<b>ŞEKİLLER DİZİNİ</b> .....	<b>x</b>
<b>ÇİZELGELER DİZİNİ</b> .....	<b>xv</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. KURAMSALTEMELLER</b> .....	<b>5</b>
2.1. Sezgisel Optimizasyon Algoritmaları.....	5
2.2. Tam Sayı Dereceli ve Kesir Dereceli Denetçi Yapıları.....	6
2.2.1. Tam sayı Dereceli Denetçi.....	6
2.2.2. Kesir Dereceli Denetçi.....	7
2.2.2.1 Kesir Dereceli Denetçi Yapısının Pratik Olarak Gerçekleştirilmesi.....	9
2.3 Yarı Analitik Bir Optimizasyon Yöntemiyle Kesir Dereceli İntegro Diferansiyel Denklemlerin Çözümü.....	11
2.3.1 Homotopy Pertürbasyon Metodu.....	12
2.3.2 Homotopy Pertürbasyon Yöntemiyle Kesir Dereceli İntegro Diferansiyel Denetçi Denkleminin Çözülmesi.....	14
2.3.2.1 Orantılı Dereceli Doğrusal Olmayan Zamanla Değişmeyen Kesir Dereceli İntegro Diferansiyel Denklemin Homotopy Pertürbasyon Metodu ile Çözümü.....	17
2.3.2.2 Orantılı Dereceli Doğrusal Olmayan Zamanla Değişen Kesir Dereceli İntegro Diferansiyel Denklemin Homotopy Pertürbasyon Yöntemi ile Çözümü.....	27
<b>3. MATERİYAL VE YÖNTEM</b> .....	<b>35</b>

3.1.	Tezde Kullanılan Materyal .....	35
3.1.1.	Çok girişli Çok Çıkışlı Helikopter Modeli.....	35
3.1.2	MATLAB ve Simulink.....	35
3.2	Tezde Kullanılan Yöntem.....	36
<b>4.</b>	<b>RASSAL PARAMETRE VEKTÖR OPTİMİZAYON YÖNTEMİ.....</b>	<b>37</b>
4.1.	Rassal Parametre Vektör Optimizasyon Yöntemi Algoritmasıyla Denetçi Parametrelerinin Otomatik Olarak Ayarlanması.....	39
4.1.1	Çalışmanın Amacı.....	39
4.1.2	Problemin Tanımlanması.....	40
4.1.3	Rassal Parametre Vektör Optimizasyon Algoritması.....	41
4.1.4	Rassal Parametre Vektör Optimizasyon Algoritmasıyla Adaptif Kesir Dereceli Denetçi Tasarımı.....	45
4.1.5	Adaptif Kesir Dereceli Denetçi Parametrelerinin Matematiksel Bir Model İçin Optimizasyon ile Türetilmesi.....	50
4.1.6	RPVO Algoritması ile Deneysel Olarak Denetçi Tasarımı.....	57
4.1.6.1	Deneysel Sonuçlar.....	58
4.2.	RPVO Algoritması ile İki Serbestlik Dereceli Kontrol Yapıları İçin Kesir Dereceli Denetçi Tasarımı.....	65
4.2.1	İki Serbestlik Dereceli Kesir Dereceli Kontrol Döngüsünün Optimizasyonu.....	66
4.2.2.	İki Serbestlik Dereceli Kesir Dereceli PID Kontrol Döngüsünün Çift Motorlu Helikopter Prototipi Üzerinden Uygulanması.....	70
4.2.3	TRMS Sisteminin Matematiksel Modelinin Çıkarılması.....	70
4.2.4	Simülasyon Sonuçları.....	71
4.2.5	Deneysel Çalışma.....	75
<b>5.</b>	<b>TABU ARAMA ALGORİTMASI TABANLI BİR OPTİMİZASYON YÖNTEMİYLE KESİR DERECELİ DENETÇİ TASARIMI.....</b>	<b>78</b>
5.1	Tabu Arama Algoritması.....	79

5.1.1	Tabu Arama Algoritmasının Temel Bileşenleri.....	80
5.1.1.1	Komşuluk (Neighbourhood).....	80
5.1.1.2	Tabu Listesi (Tabu List).....	80
5.1.1.3	Tabu Koşulları (Tabu Conditions).....	81
5.1.1.4	Aspirasyon Kriteri (Aspiration Criterion).....	81
5.1.1.5	Durdurma Kriteri (Stopping Criterion).....	81
5.1.1.6	Tabu Görev Süresi (Tabu Tenure).....	81
5.2	Çok Parametrelili Fonksiyon İçin Tabu Arama Algoritmasının Düzenlenmesi.....	81
5.2.1	Önerilen Optimizasyon Yöntemi İçin Komşuluk Matrisi.....	81
5.2.2	Önerilen Optimizasyon Algoritması İçin Tabu Listesi.....	82
5.2.3	Önerilen Optimizasyon Algoritması İçin Tabu Koşulu.....	83
5.2.4	Önerilen Optimizasyon Algoritması İçin Tabu Yaşam Süresi.....	83
5.2.5	Önerilen Optimizasyon Algoritması İçin Aspirasyon Kriteri.....	83
5.2.6	Önerilen Optimizasyon Algoritması İçin Yeniden Başlatma Prosedürü.....	83
5.2.7	Önerilen Optimizasyon Algoritması için Durdurma Kriteri.....	84
5.3	Önerilen Optimizasyon Algoritması.....	84
5.4	Tabu Arama Algoritması Tabanlı Optimizasyon Yöntemiyle Kesir Dereceli Denetçi Tasarımı.....	88
5.5	Simülasyon Sonuçları.....	92
5.5.1	Simülasyon Örnek 1.....	92
5.5.2	Simülasyon Örnek 2.....	95
5.5.3	Simülasyon Örnek 3.....	97
5.5.4	Simülasyon Örnek 4.....	99
<b>6.</b>	<b>DENETÇİ PARAMETRELERİNİN YAPAY FİZİK OPTİMİZASYON ALGORİTMASI İLE BULUNMASI.....</b>	<b>101</b>
6.1	Yapay Fizik Optimizasyon Algoritması ve Temel Optimizasyon Algoritması.....	103

6.1.1	Yapay Fizik Optimizasyon Algoritması.....	103
6.1.2	Temel Optimizasyon Algoritması.....	105
6.2	Geliştirilmiş Yapay Fizik Optimizasyon Algoritmasıyla Çoklu Parametrelerin Optimizasyonu.....	105
6.3	Geliştirilmiş Yapay Fizik Optimizasyon Algoritması ile Denetçi Tasarımı.....	112
6.3.1	PID Denetçi Parametrelerinin Ayarlanması.....	112
6.3.2	Geliştirilmiş Yapay Fizik Optimizasyon Algoritmasıyla Kesir Dereceli Denetçi Parametrelerinin Optimizasyonu.....	118
6.4	Simülasyon Çalışması.....	120
6.4.1	Simülasyon Örnek 1.....	120
6.4.2	Simülasyon Örnek 2.....	125
6.4.3	Simülasyon Örnek 3.....	128
<b>7.</b>	<b>ÇOKLU AMAÇ FONKSİYONUNA GÖRE DENETÇİ OPTİMİZASYONU.....</b>	<b>130</b>
7.1.	Geliştirilmiş Tabu Arama Algoritması ile Çoklu Amaç Fonksiyonu Kullanılarak Kesir Dereceli PID Optimizasyonu.....	132
7.1.1	Monje-Vinagre'nin Kesir Dereceli PID Tasarımında Kullanılan Kriterleri.....	132
7.1.2	Geliştirilmiş Tabu Arama Algoritmasının Çoklu Amaç Fonksiyonuyla Çoklu Parametre Optimizasyonu için Kullanılması.....	133
7.1.3	Kesir Dereceli PID Denetçinin Çoklu Amaç Fonksiyonlu Geliştirilmiş Tabu Arama Algoritması ile Optimizasyonu.....	137
7.1.4	Simülasyon Sonuçları.....	139
7.1.4.1	Simülasyon Örnek 1.....	139
7.1.4.2	Simülasyon Örnek 2.....	140
7.2.	Çoklu Amaç Fonksiyonlu Büyük Patlama Büyük Çöküş (BP-BÇ) Optimizasyon Algoritmasıyla Kesir Dereceli PID Denetçi Tasarımı.....	141
7.2.1	Çoklu Amaç Fonksiyonu ile Denetçi Parametrelerinin Ayarlanması.....	143

7.2.2	Büyük Patlama Büyük Çöküş Optimizasyon Algoritması.....	144
7.2.3	Büyük Patlama Büyük Çöküş Optimizasyon Algoritması ile Çoklu Amaç Fonksiyonu Kullanılarak Kesir Dereceli PID Denetçinin Tasarımı.....	145
7.2.4	Simülasyon Sonuçları.....	148
<b>8.</b>	<b>SONUÇLAR ve ÖNERİLER.....</b>	<b>152</b>
<b>9.</b>	<b>KAYNAKLAR.....</b>	<b>154</b>
	<b>ÖZGEÇMİŞ.....</b>	<b>169</b>

## SİMGELER VE KISALTMA DİZİNİ

PID	Oransal-İntegral-Türev (Proportional-Integral-Derivative) Kontrol
$PI^\lambda D^\mu$	Kesir Dereceli PID Denetçi
TRMS	Twin Rotor MIMO System
$K_p$	Oransal Sabit
$K_i$	İntegral Sabit
$K_d$	Türev Sabit
$\lambda$	İntegral Sabitinin Derecesi
$\mu$	Türev Sabitinin Derecesi
CFE	Continued Fractional Expansions (Sürekli Kesirli Açılım)
CRONE	Comande Robuste D'ordre Entire (Tam Sayı Dereceli Dayanıklı Kontrol)
KD	Kesir Dereceli
RPVO	Rassal Parametre Vektör Optimizasyonu
TAA	Tabu Arama Algoritması
GYFOA	Geliştirilmiş Yapay Fizik Algoritması

## ŞEKİLLER DİZİNİ

Şekil 2.1.	Doğrusal olmayan zamanla değişmeyen sistemin $t \in [0,1]$ için kesin çözüm ile HPM çözümünün karşılaştırılması.....	23
Şekil 2.2.	Doğrusal olmayan zamanla değişmeyen sistemin HPM çözümünün göreceli hata değeri.....	23
Şekil 2.3.	Doğrusal olmayan zamanla değişmeyen sistemin farklı $\alpha$ değerleri için üretmiş olduğu cevapların karşılaştırılması ( $0 < \alpha < 1$ ).....	24
Şekil 2.4.	Doğrusal olmayan zamanla değişmeyen sistemin $t \in [0,1.5]$ için HPM ve kesin çözümün karşılaştırılması.....	25
Şekil 2.5.	Doğrusal olmayan zamanla değişmeyen sistemin bileşen sayısının artırılmasıyla elde edilen sonuçların karşılaştırılması.....	26
Şekil 2.6.	Doğrusal olmayan zamanla değişmeyen sistemin bileşen sayısının artırılmasıyla elde edilen hata fonksiyonunun değişimi.....	26
Şekil 2.7.	Doğrusal olmayan zamanla değişen sistemin $t \in [0,1]$ için kesin çözüm ile HPM çözümünün karşılaştırılması.....	31
Şekil 2.8.	Doğrusal olmayan zamanla değişen sistemin HPM çözümünün hata değeri.....	31
Şekil 2.9.	Doğrusal olmayan zamanla değişen sistemin farklı $\alpha$ değerleri için üretmiş olduğu cevapların karşılaştırılması ( $0 < \alpha < 1$ ).....	32
Şekil 2.10.	Doğrusal olmayan zamanla değişen sistemin bileşen sayısının artırılmasıyla elde edilen sonuçların karşılaştırılması.....	33
Şekil 2.11.	Doğrusal olmayan zamanla değişen sistemin bileşen sayısının artırılmasıyla elde edilen hata fonksiyonunun değişimi.....	33
Şekil 4.1.	Adaptif birim geri beslemeli kontrol sistemin temel blok diyagramı.....	40
Şekil 4.2.	$\tilde{X} = [x_1 \ x_2]$ iki bileşenli parametre vektöründen $\tilde{X}_{opt}$ optimum	

	sonucun elde edilmesi.....	42
Şekil 4.3.	$x_1$ ve $x_2$ değişkenlerinin ileri ve geri yönlü hareketleri. (Durum 1 ve 2, ileri yönlü hareketleri, durum 3 ve 4 ise geri yönlü hareketleri göstermektedir).....	43
Şekil 4.4.	$X_{opt}$ değerine sabit adım aralığı ile yakınsama.....	43
Şekil 4.5.	$X_{opt}$ değerine $\Delta X^{n+1,n}$ değişken adım aralığı ile yakınsama.....	44
Şekil 4.6.	$\tilde{X}^n = [x_1^n \ x_2^n]$ değişkenlerinin optimizasyon süresince konveks hata fonksiyonuna göre değişimi.....	45
Şekil 4.7.	Hata fonksiyonun iterasyonlar süresince global minimuma yakınsaması.....	45
Şekil 4.8.	KDPID denetçi parametrelerinin optimizasyonu için RVPO algoritmasının akış şeması.....	49
Şekil 4.9.	Denetçilerin 0.6. saniyedeki birim basamak cevapları.....	51
Şekil 4.10.	Denetçilerin 2.6. saniyedeki birim basamak cevapları.....	51
Şekil 4.11.	Denetçilerin 11. saniyedeki birim basamak cevapları.....	52
Şekil 4.12.	A-KDPID denetçi parametrelerinin 12 saniyelik simülasyon süresince değişimi.....	53
Şekil 4.13.	Optimizasyon süresince ilgili denetçilere göre hatanın değişimi.....	53
Şekil 4.14.	Denetçilerin 0.2. saniyedeki birim basamak cevapları.....	54
Şekil 4.15.	Denetçilerin 3.8. saniyedeki birim basamak cevapları.....	55
Şekil 4.16.	Denetçilerin 11.4. saniyedeki birim basamak cevapları.....	55
Şekil 4.17.	Klasik PID, Diferansiyel Gelişim Algoritmasıyla Elde edilen $PI^\lambda D^\mu$ ve RPVO algoritmasıyla elde edilen $PI^\lambda D^\mu$ denetçilerinin birim basamak cevaplarının karşılaştırılması.....	56
Şekil 4.18.	TRMS deney setinin laboratuvar ortamındaki kurulumu.....	57
Şekil 4.19.	RPVO algoritmasıyla TRMS için kesir dereceli denetçi tasarımını sağlayan blok şema.....	58
Şekil 4.20.	TRMS ana motoru için klasik PID, $PI^\lambda D^\mu$ ve A- $PI^\lambda D^\mu$ denetçilerinin birim basamak cevapları.....	61

Şekil 4.21.	TRMS kuyruk motoru için Klasik PID, $PI^\lambda D^\mu$ ve A- $PI^\lambda D^\mu$ denetçilerinin birim basamak cevapları.....	61
Şekil 4.22.	TRMS ana motoru için tasarlanan A- $PI^\lambda D^\mu$ denetçinin optimizasyon süresince parametre değişimi.....	62
Şekil 4.23.	Optimizasyon süresince elde edilen hata fonksiyonun değişimi.....	63
Şekil 4.24.	Ağırlık aparatının ortada olma durumunda elde edilen birim basamak cevabı.....	63
Şekil 4.25.	İleri Yönlü İki Serbestlik Dereceli KDPID yapısı.....	67
Şekil 4.26.	Geri Yönlü İki Serbestlik Dereceli KDPID yapısı.....	68
Şekil 4.27.	CS tipi 2SD KDPID yapısı.....	68
Şekil 4.28.	SPF 2SD KDPID yapısı.....	69
Şekil 4.29.	FPD tipi İki Serbestlik Dereceli KDPID yapısı.....	69
Şekil 4.30.	Optimizasyon süresince kullanılan sistemin blok şeması.....	71
Şekil 4.31.	TRMS ana motoru için FF, FB, CS, SPF, FPD tipi 2SD KDPID denetçilerin birim basamak cevapları.....	73
Şekil 4.32.	Optimizasyon süresince FB tipi kontrol yapısı için elde edilen hata fonksiyonunu değişimi.....	73
Şekil 4.33.	Optimizasyon süresince FB 2SD KDPID denetçinin parametre değişimi.....	74
Şekil 4.34.	Üretilen hata değerlerinin karşılaştırılması.....	74
Şekil 4.35.	5 tip iki serbestlik dereceli kesir dereceli denetçi yapısının birim basamak cevaplarının karşılaştırılması.....	75
Şekil 4.36.	Optimizasyon süresince FF tipi 2SD KDPID denetçi yapısının hata fonksiyonu değişimi.....	76
Şekil 5.1.	Önerilen TAA Akış Şeması.....	87
Şekil 5.2.	Önerilen TAA'nın fonksiyon akış şeması.....	88
Şekil 5.3.	Sistem optimizasyon modeli.....	89
Şekil 5.4.	İlk iki iterasyon için TAA'nın matris konfigürasyonu.....	91
Şekil 5.5.	Matlab Simulinkte Kullanılan Simülasyon Modeli.....	92

Şekil 5.6.	Denklem 5.7’de verilen sistem için PID ve KDPID’nin birim basamak cevapları.....	93
Şekil 5.7.	Optimizasyon süresince KDPID parametrelerinin değişimi.....	94
Şekil 5.8.	Optimizasyon süresince hata fonksiyonunun değişimi.....	94
Şekil 5.9.	Denklem 5.8’de verilen sistem için PID ve KDPID’nin birim basamak cevapları.....	95
Şekil 5.10.	Optimizasyon süresince KDPID parametrelerinin değişimi.....	96
Şekil 5.11.	Optimizasyon süresince hata fonksiyonunun değişimi.....	96
Şekil 5.12.	Denklem 5.9’da verilen sistem için PID ve KDPID’nin birim basamak cevapları.....	97
Şekil 5.13.	Optimizasyon süresince KDPID parametrelerinin değişimi.....	98
Şekil 5.14.	Optimizasyon süresince hata fonksiyonunun değişimi.....	98
Şekil 5.15.	Denklem 5.10’da verilen sistem için TAA ile üretilmiş KDPID ile Geliştirilmiş DE ile üretilmiş KDPID’nin birim basamak cevapları.....	100
Şekil 6.1.	Sürü matrisi konfigürasyonu.....	110
Şekil 6.2.	GYFA’nın akış şeması.....	111
Şekil 6.3.	PID denetçi için oluşturulan sürü matrisi.....	111
Şekil 6.4.	Optimizasyonda kullanılan sistem modeli.....	113
Şekil 6.5.	GYFO algoritmasının sözde kodu.....	117
Şekil 6.6.	$m_1$ kütle tanımı için farklı hata fonksiyon tanımlarının birim basamak cevapları.....	122
Şekil 6.7.	$m_2$ kütle tanımı için farklı hata fonksiyon tanımlarının birim basamak cevapları.....	122
Şekil 6.8.	$m_3$ kütle tanımı için farklı hata fonksiyon tanımlarının birim basamak cevapları.....	123
Şekil 6.9.	Çizelge 6.1’de verilmiş olan en iyi hata fonksiyon tanımlarının birim basamak cevapları.....	123
Şekil 6.10.	Çizelge 6.1’de verilen en iyi kütle-hata ikilisi olan ( $m_3$ -ITAE) yapısının optimizasyon süresince maliyet fonksiyonunun	

	değişimi.....	124
Şekil 6.11.	Çizelge 6.1’de verilen en iyi kütle-hata ikilisi olan ( $m_3$ -ITAE) yapısının optimizasyon süresince parametre değişimi.....	124
Şekil 6.12.	Denklem 6.29’deki verilmiş sistem için GYFO Algoritması ile diğer yöntemlerin birim basamak cevaplarının karşılaştırılması.....	125
Şekil 6.13.	Çizelge 6.2’de verilen en iyi kütle-hata ikilisi olan ( $m_1$ -IAE) yapısının optimizasyon süresince maliyet fonksiyonun değişimi.....	127
Şekil 6.14.	Çizelge 6.2’de verilen en iyi kütle-hata ikilisi olan ( $m_1$ -IAE) yapısının optimizasyon süresince parametre değişimi.....	127
Şekil 6.15.	Denklem 6.29’deki sistem için GYFO Algoritması ile diğer yöntemlerin birim basamak cevaplarının karşılaştırılması.....	128
Şekil 7.1.	Önerilen Çoklu Amaç Fonksiyonlu TAA’nın Akış Şeması.....	138
Şekil 7.2.	C1(s)G1(s) için birim basamak cevabı.....	140
Şekil 7.3.	C2(s)G2(s) için birim basamak cevabı.....	141
Şekil 7.4.	Optimizasyonla KDPID denetçi parametrelerinin ayarlandığı sistem modeli.....	147
Şekil 7.5.	TRMS sisteminin ana motorunun bozucu sinyalinin KDPID ile kontrol edildiği sistem şeması.....	148
Şekil 7.6.	KDPID denetçi optimizasyonu süresince maliyet fonksiyonun değişimi.....	149
Şekil 7.7.	KDPID ile Klasik PID denetçilerin birim basamak cevaplarının karşılaştırılması.....	149
Şekil 7.8.	BP-BÇ ve RPVO algoritmalarıyla türetilmiş olan KDPID denetçilerin birim basamak cevapları.....	150

## ÇİZELGELER DİZİNİ

Çizelge 1.1.	Bazı Stokastik Algoritmalar.....	3
Çizelge 2.1.	Sezgisel algoritmalarla denetçi parametrelerinin belirlenmesi....	6
Çizelge 2.2.	Doğrusal olmayan zamanla değişmeyen sistemin farklı $\alpha$ değerlerine göre HPM ile elde edilen nümerik sonuçları.....	24
Çizelge 2.3.	Doğrusal olmayan zamanla değişmeyen sistemin farklı $\alpha$ değerlerine göre HPM ile elde edilen nümerik sonuçları.....	32
Çizelge 4.1.	Ağırlık aparatının farklı konumlarına göre RPVO algoritmasıyla elde edilen A- $PI^\lambda D^\mu$ denetçi katsayıları.....	64
Çizelge 4.2.	Ağırlık aparatının farklı konumlarına ilişkin PID $PI^\lambda D^\mu$ ve A- $PI^\lambda D^\mu$ denetçilerin 100 saniye için üretmiş oldukları hata değerleri.....	64
Çizelge 4.3.	5 Farklı tip 2SD KDPID denetçilerin RPVO algoritmasıyla elde edilen değerleri.....	72
Çizelge 4.4.	Gerçek zamanlı TRMS modeli için RPVO algoritmasıyla üretilen 5 farklı 2SD PID denetçi parametreleri.....	76
Çizelge 5.1.	Önerilen algoritmaya göre elde edilen KDPID parametreleri.....	99
Çizelge 5.2.	Önerilen TAA ve geliştirilmiş diferansiyel gelişim algoritmasına (GDGA) göre elde edilen KDPID denetçi katsayıları.....	99
Çizelge 6.1.	$C_1(s)G_1(s)$ sistemi için kütle -hata ikililerinin optimizasyon sonuçları.....	121
Çizelge 6.2.	$C_2(s)G_2(s)$ sistemi için kütle -hata ikililerinin optimizasyon sonuçları.....	126
Çizelge 6.3.	$C_3(s)G_3(s)$ sistemi için kütle -hata ikililerinin optimizasyon sonuçları.....	129
Çizelge 7.1.	BP-BÇ ve RPVO algoritmalarıyla elde edilen denetçi katsayıları.....	151

## 1. GİRİŞ

Yüz yıllardan beri insanlar optimizasyon problemleriyle karşılaşmakta ve bu problemleri çözmek için büyük bir çaba sarf etmektedir. Problemlerin yapısına göre yöntemlerde farklılık gösterir. Uygun yöntemin seçilmesi optimizasyon sürecindeki en önemli adım olarak gösterebilir [1-4]. Optimizasyon yöntemleri i) analitik yöntemler ve ii) nümerik yöntemler olmak üzere 2 gruba ayrılabilir. Problemin matematiksel formu, amaç fonksiyonu ve kısıtları tam olarak biliniyorsa, doğrusal programlama, doğrusal olmayan programlama, dinamik programlama ve tam sayı programlama v.b. analitik yöntemler kullanılabilir. Nümerik yöntemlerde ise problemin matematiksel formunun tamamıyla bilinmesine ihtiyaç yoktur. Bu yöntemlerde amaç fonksiyonunun bilinmesi gerekir. Böylece ilgili amaca yakınsamak için optimizasyon süreci çeşitli yöntemler kullanılarak sürdürülür. Bu yöntemler deterministik ve stokastik olarak iki gruba ayrılabilir.

Analitik yöntemlerde, kendi aralarında çözülmesi gereken matematiksel denklemin yapısına göre doğrusal ve doğrusal olmayan programlama olmak üzere 2 grupta incelenir. Daha sonra ise denklemin konveks (dış bükey) veya konkav (iç bükey) olma durumları incelenerek bulunan optimal noktaların yerel veya global optimum nokta olma durumları tespit edilir [4-9]. Çünkü analitik optimizasyon problemlerinde, verilen denklem konveks bir formda ise bulunan lokal nokta aynı zamanda global optimum çözüm noktasını gösterir. Fakat denklem konkav bir yapıda ise bu durum hakkında kesinlik yoktur [4-9].

Bunun yanı sıra analitik yöntemlerde kullanılan kısıtların yapısına göre optimizasyon problemi, kısıtlı ve kısıtsız olmak üzere sınıflandırılabilir. Özellikle çözülen problemin kısıtının formu çözüm yöntemini tamamıyla değiştirir. Örneğin, problem doğrusal ve kısıtsız bir optimizasyon problemi ise bilinen denklem çözme yöntemleriyle çözülebilir. Doğrusal ve kısıtlı bir problem ise problemin çözümü için Lagrange denklemi kullanılarak doğrusal olmayan kısıtlı problem kısıtsız bir problem formuna getirilip çözülür. Diğer taraftan problemin doğrusal olmayan kısıtlı bir problem olması halinde ise Fritz John ve Kraush Khun Tucker yöntemleriyle bulunan noktanın optimalite testi yapılır [10].

Ancak analitik yöntemler sürekli değişkenlere, sürekli bir amaç fonksiyonuna ve iyi bir başlangıç noktasına ihtiyaç duyar. Bu şartlar sağlandığı

takdirde etkili bir optimizasyon süreci gerçekleştirilir [11, 12]. Analitik optimizasyon problemleri yasaklanmış bölgelere, dış limitlere, pürüzlü, pürüzsüz veya konveks amaç fonksiyonlarına ihtiyaç duyar. Konveks olmayan optimizasyon problemleri analitik yöntemlerle çözülememektedir. Dinamik programlama, karışık tam sayı programlama ve bunların türevleri konveks olmayan optimizasyon problemlerine çözümler getirebilmesine rağmen, genellikle bu yöntemlerle çözüm yapılırken çok fazla işlem adımı gerekir [11, 12].

Bu nedenle problem bileşenlerinin tamamıyla bilinmemesi durumunda çözümler üretebilen deterministik ve stokastik nümerik yöntemler global çözüme en yakın çözümü bulmak için yaygın olarak tercih edilir.

Genellikle nümerik yöntemler mühendislik problemlerinin çözümünde yaygın bir şekilde kullanılır. Özellikle stokastik yöntemler örneğin meta sezgisel yöntemler, zeki sürü yöntemleri (computational intelligent) v.b... uygun matematiksel yaklaşımlara alternatif olarak global çözüme en yakın çözümü bulmak için kullanılır. Stokastik yöntemler global çözüm aramaları için parametre uzayında daha rahat hareket imkânı sağladığından daha fazla tercih edilir. Çünkü bu yöntemlerde, analitik yöntemlerin ihtiyaç duyduğu gibi sürekli bir maliyet fonksiyonuna ve değişkenlere ihtiyaç duyulmaz.

Literatürde farklı problemlerin çözümü için birçok sezgisel algoritma bulunmaktadır. Çizelge 1.1'de bunlardan bazıları verilmektedir.

Özellikle kontrol mühendisliği problemleri için sezgisel algoritmalarla denetçi tasarlanması son derece önemlidir. Çünkü bu problemler, bilinen analitik yöntemlerle çözülemez.

Literatürde denetçi parametrelerini sezgisel algoritmalarla belirleyen farklı yöntemler yer almaktadır. Birçok problemde farklı bir optimizasyon yöntemi kullanılmakta ve sistemlere göre de değişiklik göstermektedir. Bundan dolayı bu alanda yeni algoritmaların önerilmesi ve denetçi tasarımında kullanılmamış algoritmaların denetçi tasarımında kullanılabilir hale getirilmesi bilim dünyasına önemli katkılar sağlayacaktır.

**Çizelge 1.1** Bazı Stokastik Algoritmalar

<b>Yöntem</b>	<b>Referans</b>
Yapay Fizik Algoritması	[13]
Temel Optimizasyon Algoritması	[2]
Tabu Arama Algoritması	[14]
Evrimsel Algoritma	[15]
Karıca Koloni Optimizasyonu Algoritması	[16]
Genetik Algoritma	[17]
Parçaçık Sürü Optimizasyonu Algoritması	[18]
Tavlama Simülasyon Metodu Algoritması	[19]
Büyük Patlama Büyük Çöküş Algoritması	[20]
Yerçekimi Arama Algoritması	[21]
Yapay Arı Kolonisi Algoritması	[22]
Yapay Bağışıklık Algoritması	[23]
Gazların Brownian Hareketi Algoritması	[24]

Karmaşık sistemler içerisinde uçuş kontrol sistemleri önemli yer tutar. Çünkü bu sistemler kompleks doğrusal olmayan bileşenlere sahiptir. Çevreyle doğrudan temasları olduğundan, sistemlerin matematiksel modelleri sürekli değişebilir. Bu da denetçi parametrelerinin belirlenmesini güçleştirir. Sistem havada olduğu için her türlü çevresel etkiye maruz kalacağından zor bir kontrol problemi olarak karşımıza çıkar. Bu sistemlere sezgisel algoritmalarla denetçi tasarımı fikri uçuş kontrol uygulamaları için önemlidir. Uçuş kontrol sistemlerine önerilen veya bu alana uygulanmamış algoritmalarla denetçi tasarlama fikri literatürde kabul görecektir yaklaşım olarak adlandırılabilir.

Uçuş kontrol ve diğer kontrol sistemleri için de sezgisel algoritmalarla denetçi parametrelerinin belirlendiği literatürde birçok çalışma bulunmaktadır [25-39]. Bu tarz sezgisel algoritmalarla sistemlere etkin denetçiler tasarlanır. Fakat sezgisel algoritmanın modifiye edilmesiyle veya mevcut probleme uygun algoritma oluşturulmasıyla çok daha etkin denetçiler elde edilebilir.

Bu tez çalışması şöyle düzenlenmiştir. Bölüm 2'de sezgisel algoritmalar ve kesir dereceli matematik ve denetçi yapılarından bahsedilmiştir. Bunun yanı sıra ilgili bölümün sonunda kesir dereceli integro diferansiyel bir ifadenin yarı analitik bir yöntem ile çözümü sunulmuştur. Bölüm 3'de tezde kullanılan materyal ve

yöntemden bahsedilmiştir. Bölüm 4’de uçuş kontrol sistemlerinde kullanılabilir bir algoritma olan Rassal Parametre Vektör optimizasyon yöntemi önerilmiş ve denetçi tasarımı süreci irdelenmiştir.

Bölüm 5’de literatürde olan fakat kesir dereceli denetçi tasarımı kullanılmamış Tabu Arama Algoritması özgün bir şekilde kodlanarak kesir dereceli denetçi tasarlayabilecek yeteneğe ulaştırılmıştır. Bu şekilde modifiye edilmiş bir algoritma elde edilmiştir. Bölüm 6’da literatürde olan fakat denetçi tasarımı kullanılmamış olan Yapay Fizik Optimizasyon Algoritması tam sayı dereceli ve kesir dereceli denetçi tasarlayabilecek şekilde kodlanıp sistemler için kesir dereceli ve tam sayı dereceli denetçiler tasarlanmıştır. İlgili algoritmanın performansına etki eden yapılar için farklı algoritmik süreçler uygulanarak ilgili problem için etkin optimizasyon konfigürasyonu elde edilmiştir. Tezin son bölümde çoklu amaç fonksiyonunun kullanılmasıyla optimizasyon süreçlerinin etkinliğinin analizi farklı problemler üzerinden Tabu Arama ve Büyük Patlama Büyük Çöküş optimizasyon algoritmaları kullanılarak irdelenmiştir.

## 2. KURAMSAL TEMELLER

### 2.1 Sezgisel Optimizasyon Algoritmaları

Sezgisel algoritmalar, sistemlerin matematiksel modelinin tam anlamıyla bilinmediği durumlarda sadece verilen bir amaç fonksiyonuna göre ilgili parametrelerin türetilmesini sağlamaktadır.

Sezgisel algoritmalar, parametre vektör uzayında ilgili algoritmaların temel felsefesine göre tarama yapar. Örneğin Genetik algorithmada genlerin değişim felsefesine göre, Tabu arama algoritmasında denenmiş olan parametre vektörlerini bir daha denemeyerek, yapay arı kolonisinde arıların hareket yapısına göre, karınca kolonisi algoritmasında karıncaların en kısa yolu bulma terminolojisine göre parametre vektör uzayını taramaktadır.

Genellikle bu tarz problemlerde amaç fonksiyonu olarak hata fonksiyonu kullanılır. İlgili hatayı minimize etmek için parametre vektör uzayı taranır. Fonksiyonun minimizasyonu  $\min f(x)$  şeklinde ifade edilir.

Sezgisel algoritmalarda, başlangıçtaki parametre vektör uzayı deterministik veya stokastik olarak oluşturulabilir. Birçok algorithmada başlangıç parametre vektör uzayı tamamıyla rastsal olarak oluşturulur. Bunun yanında bazı algoritmalarda ise bu yapı belli bir yönteme göre türetilir. Algoritma yeni değerleri türetirken, parametre vektör uzayındaki bir değerden hareketle bir sonraki iterasyondaki değerini üretir. Ama bazı durumlarda ilgi parametre vektör uzayından hareketle ve ondan farklı olarak bir sonraki iterasyondaki değeri türetebilmektedir. Bu yapı algoritmaların farklılaşmasındaki en önemli etkenler arasında gösterilir.

Dolayısıyla sezgisel optimizasyon problemleriyle çözüm yapılacağı zaman hangi durumun probleme uygun olduğuna bakılıp, yöntem ona göre seçilmelidir veya kullanılacak yöntem, problemin durumuna göre yöntemin felsefesini bozmayacak şekilde düzenlenmelidir.

Bu nedenle kontrol mühendisliği çalışmalarında denetçi parametrelerin belirlenmesi süresince sezgisel algoritmalar sıklıkla tercih edilir. Çizelge 2.1'de denetçi parametrelerinin belirlenmesi için literatürde en çok kullanılan sezgisel algoritmalar verilmektedir.

**Çizelge 2.1** Sezgisel algoritmalarla denetçi parametrelerinin belirlenmesi

<b>Yöntem</b>	<b>Kullanıldığı Alan</b>
Karınca Koloni Algoritması	Bulanık denetçi tasarlanması [25]
Rekabetçi Sömürge Algoritması	PID denetçi parametrelerinin belirlenmesi [26]
Parçacık Sürü Optimizasyonu	Kesir dereceli PID denetçi tasarımı [27]
Karışık Kurbağa Sıçraması Algoritması	Çok değişkenli PID denetçi tasarımı [28]
Evrensel Gelişim Algoritması	Çok Değişkenli PID denetçi tasarımı [29]
Diferansiyel Gelişim Algoritması	Kesir dereceli PID denetçi tasarımı [30]
Genetik Algoritma	PID denetçi tasarımı [31]
Elektromayetizma Algoritması	Kesir dereceli denetçi optimizasyonu [32]
Sirkesineği Optimization Algoritması	Bulanık KDPID denetçi optimizasyonu [33]
Bakterilerin Beslenmesi Algoritması	Yüksek frekansı denetçi tasarımı [34]
Ateş böceği Optimizasyon Algoritması	Çok değişkenli PID denetçi optimizasyonu [35]
Yapay Arı kolonisi Algoritması	PID denetçi optimizasyonu [36]
Yarasa Arama Algoritması	Güç sistemi denetçi optimizasyonu [37]
Emparyalist Rekabetçi Algoritması	KDPID Optimizasyonu [38]
Guguk Kuşu optimization Algoritması	Bulanık denetçi optimizasyonu [39]

Optimizasyon yöntemleri ve kullanılması planlanan denetçi yapıları hakkında kısa bilgi sonraki bölümde sunulacaktır.

## **2.2 Tam Sayı Dereceli ve Kesir Dereceli Denetçi Yapıları**

### **2.2.1 Tam Sayı Dereceli Denetçi**

Oransal-integral-türev (PID) denetçiler basit yapısı ve dayanıklı performansından dolayı sistemin dinamik cevabını geliştirmek ve kalıcı hal hatasını azaltmak için kullanılır. PID denetçiler özellikle endüstriyel uygulamalarda geniş bir uygulama alanı bulmaktadır.

Denetçi içerisindeki türev operatörü açık çevrim geçiş fonksiyonuna sınırlı bir sıfır ekleyerek geçiş fonksiyon cevabını geliştirir. İntegral operatörü ise başlangıç noktasına bir kutup ekleyerek sistemin birim basamak cevabındaki kalıcı hal hatasını azaltır. PID denetçinin geçiş fonksiyonu şöyledir:

$$C(s) = k_p + \frac{k_i}{s} + k_d s \quad (2.1)$$

$k_p$  oransal katsayısı,  $k_i$  integral katsayısını ve  $k_d$  türev katsayısını gösterir. Görüldüğü gibi PID denetçi yapısında optimize edilmesi gereken 3 parametre bulunur. Sezgisel algoritmalarla çeşitli sistemler için parametreler optimize edilebilmektedir.  $k_i$  ve  $k_d$  parametrelerin 0 olması durumlarına göre de farklı denetçi konfigürasyonları oluşturulabilir. Örneğin  $k_i = 0$  olma durumunda PD denetçi,  $k_d = 0$  durumunda ise PI denetçi yapıları oluşur.

### 2.2.2 Kesir Dereceli Denetçi

Kesir dereceli denetçi yapısını açıklamadan önce kesir dereceli matematiğin açıklanması gerekir. Kesir dereceli türev ve integraller yaklaşık 300 yıldır tartışılmaktadır [40]. Fakat son yıllarda, sistem modellenmesinde ve mühendislik uygulamalarında tam değer integro-diferansiyel ifadelerin yerine kesir dereceli integro-diferansiyel ifadeler kullanılmaya başlanmıştır [40]. Kesir dereceli sistemler, tamsayı dereceli sistemlerin genelleştirilmiş hali olarak ele alınabilir [41-46]. Denklem 2.2’de kesir dereceli PID denetçinin matematiksel formu gösterilmektedir.

$$C(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu \quad (2.2)$$

Burada  $\lambda$  kesir dereceli integralin derecesi,  $\mu$  ise kesir dereceli türevin derecesidir. Kesir dereceli bir fonksiyonda integral veya türev hesabını doğrudan yapmak zor olduğu için bu konuda bazı yaklaşık tanımlamalardan yararlanır. En çok kullanılan tanımlamalar Grünwald-Letnikov, Riemann-Liouville ve Caputo tanımlarıdır [41, 42].

Grünwald-Letnikov tanımı aşağıdaki şekilde verilir:

$${}_a D^r_t f(t) = \lim_{h \rightarrow 0} h^{-r} \sum_{j=0}^{\left[ \frac{t-a}{h} \right]} (-1)^j \binom{r}{j} f(t-jh) \quad (2.3)$$

Riemann-Liouville tanımı aşağıdaki şekilde gösterilir:

$${}_a D^r_t f(t) = \frac{1}{\Gamma(n-r)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{r-n+1}} d\tau, \quad (n-1 < r < n) \quad (2.4)$$

Burada  $h$  zaman artış miktarıdır.  $\Gamma(\cdot)$  Euler gamma fonksiyonudur ve şöyle ifade edilir,

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \quad (2.5)$$

Burada  $x$  tamsayı olduğunda gamma fonksiyonu faktöriyel ifadesi cinsinden yazılabilir,

$$\Gamma(x+1) = x! \quad (2.6)$$

Caputo tanımı ise aşağıdaki gibi verilir:

$${}_a D^r_t f(t) = \frac{1}{\Gamma(n-r)} \int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{r-n+1}} d\tau, \quad (n-1 < r < n) \quad (2.7)$$

Denklemlerdeki  $a$  ve  $t$ , operasyon limitlerini ve  $r \in R$  türev veya integral derecesini göstermektedir.

Yapılan çalışmalar sonucunda kesir dereceli sistemlerin tam sayı dereceli sistemlere göre daha iyi sonuçlar üretebildiği gözlemlenmiştir [47].

Kesir dereceli aritmetikte sağlanan gelişimle birlikte, kontrol uygulamalarında kesir dereceli sistemler kullanılmaya başlanmıştır. İlk olarak PID denetçinin kesir

dereceli versiyonu Podlubny tarafından Deklem 2.2 gösterildiği gibi önerilmiştir [47]. Ayrıca literatürde kesir dereceli kontrol sistemlerine ilişkin birçok çalışma bulunur [48-50].

Geleneksel PID denetçiler, kolay gerçekleştirilebilir olması ve iyi çalışılmış bir denetçi yapısı olmaları gibi avantajları nedeni ile endüstriyel işlemlerde, otomasyon sistemlerinde, güç elektroniğinde yaygın olarak kullanılır [51-56]. Günümüzde, gelişen teknoloji daha karmaşık ve ileri kontrol tekniklerinin uygulanmasına imkân sağlar duruma gelmiştir. Bu nedenle geleneksel PID denetçi yapısını geliştirmeye dönük çalışmalar son dönemde artmıştır: Yapay sinir ağları PID ile kullanılması [57]'de , değişken PID (Variable PID) [58]'de ve adaptif PID yapıları [59]'da sunulmaktadır.

Geleneksel PID denetçi performansını artırmaya dönük bir diğer çalışma alanı ise kesir dereceli PID yapıları için açılmıştır [47-50]. Kesir dereceli PID yapılarında, geleneksel PID katsayıları  $k_p$ ,  $k_i$  ve  $k_d$ 'ye ek olarak, kesirli integral derecesi ( $\lambda$ ) ve kesirli türev derecesi ( $\mu$ ) parametreleri denetçi yapısına kazandırılmıştır. Ayarlanabilir beş parametreye sahip olan kesir dereceli PID denetçilerin ( $PI^\lambda D^\mu$ ), üç parametreye sahip geleneksel PID denetçilerine göre daha iyi bir kontrol performansı sergileyebildiği görülmüştür [60-62]. Bunun başlıca nedeni,  $PI^\lambda D^\mu$  denetçilerin, PID'lere göre daha geniş bir yelpazede tepki cevabı sunabilmesidir.  $PI^\lambda D^\mu$  denetçi yapılarının pratikte uygulama sahası bulabilmesi için, beş denetçi parametresinin ( $k_p, k_i, k_d, \lambda, \mu$ ) optimize edilmesi gerekir.

### **2.2.2.1 Kesir Dereceli Denetçi Yapısının Pratik Olarak Gerçekleştirilmesi**

Kesir dereceli denetçi yapısının pratikte gerçekleştirilmesi birçok zorluğu da beraberinde getirir. Literatürde kesir dereceli denetçi yapısının pratik olarak gerçekleştirilmesini sağlayan birçok çalışma bulunmaktadır. [63-64]. Valerio, Crone, Carlson ve CFE yöntemlerini birleştirerek kesir dereceli denetçi yapısının pratikte gerçekleştirilmesini sağlayan bir toolbox önermiştir [64]. Bunlardan Crone yöntemi Denklem 2.8'e göre kesir dereceli denetçi yapısı gerçekleştirir.

$$C(s) = \tilde{k} \prod_{n=1}^N \frac{1 + s/w_{zn}}{1 + s/w_{pn}} \quad (2.8)$$

$\tilde{k}$  ayarlanabilir kazançtır.  $k=1$  olması durumunda 1 rad/s’de  $|C(s)|=0$ ’dır. Sıfırlar ve kutuplar verilen frekans  $[w_l; w_h]$  aralığında bulunur.

$$\alpha = (w_h / w_l)^{\frac{v}{N}}, \eta = (w_h / w_l)^{\frac{1-v}{N}}$$

$$w_{zl} = w_l \sqrt{\eta}, w_{zn} = w_{p,n-1} \eta, n = 2 \dots N \quad (2.9)$$

$$w_{pn} = w_{z,n-1} \alpha, n = 1 \dots N$$

İkinci yöntem ise Carlson’s metodudur.  $C^\alpha(s) = g(s)$  yapısı “Newton iterative” metodu kullanılarak çözümlenir:

$$C_n(s) = C_{n-1}(s) \frac{\alpha - (1)C_{n-1}^\alpha(s) + (\alpha + 1)g(s)}{(\alpha + 1)C_{n-1}^\alpha + (\alpha - 1)g(s)} \quad (2.10)$$

Diğer bir yöntem ise Matsuda yöntemidir. CFE metodu kullanılarak irrasyonel fonksiyonlar şeklinde elde edilir [65].

$$C(s) = a_0 + \frac{s - s_0}{a_1 +} \frac{s - s_1}{a_2 +} \frac{s - s_2}{a_3 +} \quad (2.11)$$

Denklemden,  $a_i = v_i(s), v_0 = H(s), v_{i+1} = \frac{s - s_i}{v_i(s) - a_i}$ . Valerionun önermiş olduğu

toolbox’ta bu üç yöntemde ayrı ayrı kullanılır. Bu tez çalışmasında kesir dereceli

denetçi yapısının pratik olarak gerçekleştirilmesi için Valerio'nun araç kutusu kullanılmıştır.

### **2.3 Yarı Analitik Bir Optimizasyon Yöntemiyle Kesir Dereceli İntegro Diferansiyel Denklemlerin Çözümü**

Literatürde optimizasyon problemleri genellikle analitik ve nümerik yöntemlerle çözümlenir. Kullanılanacak yöntemi belirleyen, problemin kendi dinamikleridir. Tezin önceki bölümlerinde de bahsedildiği gibi problemin matematiksel modeli ve kısıtları tamamıyla biliniyor ise analitik yöntemler kullanılır. Sisteme ilişkin bilinmeyenlerin sayısının artmasıyla kullanılan yöntemler analitik çözümlerden nümerik çözümlere doğru evrilmektedir [67].

Aslında analitik yöntemlerdeki işlem maliyeti, araştırmacıları bu konulardan biraz uzaklaştırmıştır. Nümerik yöntemlerdeki hızlı sonuç elde etme durumu analitik yöntemlerde başlangıçta pek mümkün değildir. Ancak çözümler manuel olarak yapıldıktan sonra üretilen denklemler parametrik yapılarak bilgisayarda kodlanmasıyla hızlı çözümler elde edilebilir.

Analitik çözümden nümerik çözüme geçişin keskin bir şekilde olmaması durumunda yarı analitik veya yarı nümerik yöntemlerden söz edilebilir. Özellikle sistemin matematiksel modelinin tam anlamıyla çözümlenememesinden dolayı bu tarz yöntemler tercih edilebilir [66, 67]. Kesir dereceli integro diferansiyel denklemlerde bu denklem sınıfına girmektedir. Tam sayı dereceli integro diferansiyel yöntemlerle kesir dereceli (KD) integro diferansiyel denklemler çözümlenememektedir. Sistemin matematiksel modeli mevcuttur hatta çoğu zaman modele ilişkin kısıtlarda bilinmektedir. Fakat KD integro diferansiyel ifadeleri tam anlamıyla çözebilecek bir yöntem mevcut değildir. Bundan dolayı bu tarz bir problemin çözümünde yarı analitik veya yarı nümerik bir yöntemin tercih edilmesi son derece uygun düşmektedir [67].

Tezin bu bölümünde KD doğrusal zamanla değişmeyen integro diferansiyel ve KD doğrusal zamanla değişen integro diferansiyel denklemler için yarı analitik veya yarı nümerik bir yöntem olan Homotopy Pertürbasyon Metodu (HPM) kullanılarak bir çözüm önerilmektedir.

Çözümlerin doğruluğu matematiksel denklemlerdeki kesirli derecenin 1'e eşit olma durumu yani tam sayı dereceli çözümü ile karşılaştırılarak sunulmaktadır.

Doğrusal zamanla değişmeyen ve doğrusal zamanla değişen integro diferansiyel denklemler HPM'nin prosedürüne göre matematiksel olarak çözümlenmiş ve daha sonra elde edilen çözümler bilgisayarda MATLAB ortamına aktararak parametrik hale getirilmiş ve böylelikle analizler kolaylıkla yapılmıştır.

Takip eden bölümlerde HPM ve KD integro diferansiyel ifadelerle geliştirilen çözümler ayrıntılarıyla ve açıklayıcı grafiklerle sunulmuştur [67].

### 2.3.1 Homotopy Pertürbasyon Metodu

Durum uzay modeli, gerçek zamanlı sistemleri zaman tabanında modellemek için yaygın olarak kullanılır. Bu modeller durum geçiş denklemleriyle gerçek zamanlı sistemin davranışlarını daha iyi ifade eder. Hatta durum geçiş denklemleriyle diferansiyel denklem sistemleri de tamamıyla ifade edilebilir. Denklemler giriş çıkış değişkenlerine göre analitik ve nümerik olarak çözülebilir. Bundan dolayı durum uzay modelleri mühendislikte ve diğer bilim dallarında yaygın olarak kullanılır [67, 68].

Gerçek sistemin doğrusal olmayan davranışını göz önünde bulundurmak için doğrusal olmayan durum uzay denklemlerinin çözümü bilim dünyası açısından büyük önem taşır. Araştırmacıların sunmuş oldukları raporlara göre gerçek sistemlerin düşük veya yüksek oranda KD bileşenlere sahip olduğu belirtilmektedir [67, 69,70].

Günümüzde ise KD sistemler araştırmacıların son derece ilgisini çekmektedir [70-72]. KD matematik, gerçek sistemleri daha iyi karakterize etmesinden ötürü sistem modellemesinde ve analizinde kullanılır [71]. Chen ve arkadaşları kesir dereceli türev ve integralin viskoelastik sistem, dielektrik polarizasyon, elektrot-elektrolit polarizasyonu ve elektromagnetik dalgalar gibi birçok alanda kullanılabileceği göstermiştir [72].

Bilgi işleme sistemlerindeki gelişmelerden dolayı, durum uzay modellerinin çözümlerinde de genellikle nümerik yöntemler tercih edilir. Çünkü nümerik yöntemler analitik yöntemlere göre daha az karmaşıklığa sahiptir. Bu sebepten analitik çözüm yöntemleri nadiren kullanılır. Ancak elde edilen analitik çözümler sistem optimizasyonunda, karalılıkta ve tasarım problemlerinde önemli ölçüde

fayda sağlar. Bunun yanı sıra birçok mühendislik alanında doğrusal ve doğrusal olmayan problemlerin çözümünde pertürbasyon yöntemleri kullanılır. Fakat bilindiği gibi pertürbasyon yöntemlerinin diğer birçok analitik yöntemlerde olduğu gibi bazı kısıtları bulunur. İlk kısıt olarak birçok pertürbasyon yönteminde bulunan pertürbasyon serisi açılırken küçük bir kat sayının kullanılması gösterilebilir. Bundan dolayı yaklaşık çözüm bu pertürbasyon serisine göre elde edilir. Bu parametrenin doğru seçilmesiyle tatmin edici çözümler üretilebilirken yanlış seçilmesi halinde gerçek çözümden ıraksanabilmektedir. HPM ile küçük parametrenin seçimi problemi elimine edilebilir.

Liao 1995 yılında yeni bir pertürbasyon tekniği olan Homotopy tekniğini önermiştir. Bu teknik pertürbasyon yöntemlerindeki bir kısıt olan seriye açılırken kullanılan küçük kat sayının belirlenmesi problemini çözmektedir [73]. Daha sonra 1999 yılında Ji Huan He tarafında homotopy tekniği ile pertürbasyon tekniği birleştirilerek HPM oluşturulmuştur [74].

Bu yöntem doğrusal olmayan denklem çözümlerinde kullanılır. Önerilen bu yöntem diğer pertürbasyon yöntemlerinde olduğu gibi seri açılımında küçük kat sayılara ihtiyaç duymamaktadır. Bu avantaj HPM'nin diğer pertürbasyon yöntemlerine karşın en belirgin üstünlüğüdür. Bu gelişmeden sonra bu yöntem He'nin pertürbasyon metodu olarak adlandırılmıştır. Bu yöntem ile doğrusal ve doğrusal olmayan problemlere asimptotik çözümler üretilebilir [75]. Yöntemde tanımlanmış bir katıştırma parametresi kullanılmıştır [76]. Literatürde birçok diferansiyel denklem He'nin HPM metodu kullanılarak çözümlenmiştir [75-77]. Örneğin HPM kullanılarak doğrusal olmayan kesir dereceli parçacık diferansiyel denklemlerin çözümü [78]'de sunulmaktadır. HPM metodunun gelişmiş versiyonunun doğrusal olmayan problemlerin çözümündeki kullanımı da [79]'da gösterilir.

Tezin takip eden bölümlerinde bu yöntem kullanılarak kesir dereceli doğrusal ve doğrusal olmayan denetçi denklemlerine ilişkin çözümler sunulmuştur.

### 2.3.2 Homotopy Pertürbasyon Yöntemiyle Kesir Dereceli İntegro Diferansiyel Denetçi Denkleminin Çözülmesi

Aslında KD denklemlerle ilgili geniş bir literatür çalışması tezin önceki bölümlerinde verilmiştir. Fakat tezin her bölümünde KD integro diferansiyel ifadeler kullanıldığı için ilgili konuya başlamadan KD integro diferansiyel ifadeleri tanımlamaya yönelik girişler yapılmıştır [67].

Önceki bölümlerden de belirtildiği gibi KD türev ve integral, tamsayı dereceli türev ve integralin bir uzantısı olarak kullanılır. Yani tam sayı dereceli türev ve integralin derecesinin kesir derece ile değiştirilmesiyle elde edilir. Bu değişiklik bu kadar basit olmasına karşın yapılan çözümler tamsayı dereceli integro diferansiyel ifadelerin çözümünden çok farklılaşmaktadır.  ${}_a D_t^\alpha$ , operatörün de  $\alpha > 0$ , durumu KD türevi ( $d^\alpha/dt^\alpha$ ) gösterir.  $\alpha < 0$ , durumu ise

kesir dereceli integrali ( $\int_a^t (d\tau)^{(-\alpha)}$ ) gösterir. KD ifadenin genel formu Denklem 2.7'de verilmektedir [67, 80, 81].

Kesir dereceli integro diferansiyel denklemlerin yapısal esnekliği birçok güncel çalışmada tercih edilmesini sağlamıştır [82, 83]. Orantılı KD sistemler, kesir dereceli sistemlerin bir sınıfıdır. Bu sistemlerde, sistemin türev derecesi  $n\alpha$  şeklinde belli bir oranda artmaktadır. Bu ifadede  $\alpha = 0,1,2,3\dots$  ve  $n \in Z$  şeklindedir.

Orantılı dereceli doğrusal olmayan zamanla değişen sistemin matematiksel formu aşağıdaki gibidir [41,84].

$$D^{n\alpha}y(t) + a_{n-1}(t)D^{(n-1)\alpha}y(t) + \dots + a_1(t)D^\alpha y(t) + a_0(t)y(t) + f(y(t)) = b_0(t)u(t) \quad (2.12)$$

Denklem 2.12 durum uzay modeli formatında  $D^{i\alpha}x_i = y^{(i\alpha)}$  eşitliği kullanılarak aşağıdaki gibi oluşturulur.

$$D^\alpha x(t) = A(t)x(t) + f(x(t), t) - B(t)u(t) \quad (2.13)$$

$$y(t) = C(t)x(t) \quad (2.14)$$

Durum vektörü  $x(t) = [x_0 \ x_1 \ x_2 \ \dots x_n]$  şeklinde ifade edilir. Denklemdeki  $A(t)$ , zamanla değişen durum matrisini gösterir. Tezin bu bölümünde orantılı dereceli doğrusal olmayan zamanla değişen ve orantılı dereceli doğrusal olmayan zamanla değişmeyen KD integro diferansiyel ifadelerin HPM metoduyla çözümü sunulmaktadır.

Homotopy kavramı doğrusal olmayan topolojide kullanılan önemli bir yapıdır. Genellikle doğrusal olmayan matematiksel denklemlerin köklerini bulmak için kullanılır [73, 85, 86]. Özellikle HPM, KD integro diferansiyel ifadelerin çözümünde kullanılan, He tarafında önerilmiş bir yaklaşık çözüm metodudur [87]. HPM tekniğinin KD küresel Riccati diferansiyel ifadelerin çözümündeki kullanımını Odibat ve arkadaşları tarafından [88]'de gösterilmiştir.

Genellikle KD diferansiyel denklemlerin HPM ile çözümünü literatürde iki farklı formda bulunmaktadır [88]. Bunlar;

$$\frac{du^m}{dt^m} + L(u) - f(t) = p \left[ \frac{du^m}{dt^m} - N(u) - D^\alpha u \right] \quad p \in [0,1] \quad (2.15)$$

$$\frac{du^m}{dt^m} - f(t) = p \left[ \frac{du^m}{dt^m} - L(u) - N(u) - D^\alpha u \right] \quad p \in [0,1] \quad (2.16)$$

şeklindedir. Denklemlerdeki  $u$  bir analitik fonksiyondur.  $D^\alpha$  ise Caputo ifadesinin KD türevini gösterir.  $\alpha$  ise kesirli derecenin miktarını tanımlar.  $L(u)$  ve  $N(u)$ , fonksiyoları ise sırasıyla denklemin doğrusal ve doğrusal olmayan

bölmelerini gösterir.  $p$ , homotopy parametresidir.  $p = 0$  olması durumunda denklem (2.15) doğrusal bir denklem olmaktadır.

$$\frac{d^m u}{dt^m} + L(u) = f(t) \quad (2.17)$$

ve buna bağlı olarak denklem (2.16)'da doğrusal bir denkleme dönüşür.

$$\frac{d^m u}{dt^m} = f(t) \quad (2.18)$$

Denklem (2.15) veya denklem (2.16) kullanılan HPM'nin temel yaklaşımı  $p$ 'nin güç serisine açılmış şekli aşağıda gibi gösterilir.

$$u = u_0 + pu_1 + p^2u_2 + p^3u_3 + \dots \quad (2.19)$$

Bu güç serisi denklem (2.15) veya denklem (2.16)'da yerine konulduğunda  $p$ 'nin güç tanımı serisi denklem (2.20)'deki gibi elde edilir.  $p$ 'nin ilgili güç serisine göre açılımı yardımıyla istenilen doğrusal olmayan kesir dereceli denklem çözülebilir [85-88]. Denklem (2.20) güç serisinin Denklem (2.15)'de yerine konulmuş halidir. Denklem 2.21'de ise güç serisinin Denklem (2.16)'da yerine konulmuş versiyonu gösterilmektedir.

$$\begin{aligned} p^0; \frac{d^m u_0}{dt^m} + L(u_0) &= f(t) \\ p^1; \frac{d^m u_1}{dt^m} + L_1(u_0, u_1) &= \frac{d^m u_0}{dt^m} + N(u_0) - D^\alpha u_0 \\ p^2; \frac{d^m u_2}{dt^m} + L_1(u_0, u_1, u_2) &= \frac{d^m u_1}{dt^m} + N(u_0, u_1) - D^\alpha u_1 \\ p^3; \frac{d^m u_3}{dt^m} + L_1(u_0, u_1, u_2, u_3) &= \frac{d^m u_2}{dt^m} + N(u_0, u_1, u_2) - D^\alpha u_2 \\ &\vdots \end{aligned} \quad (2.20)$$

$$\begin{aligned}
p^0; \frac{d^m u_0}{dt^m} &= f(t) \\
p^1; \frac{d^m u_1}{dt^m} &= \frac{d^m u_0}{dt^m} - L_0(u_0) - N_0(u_0) - D^\alpha u_0 \\
p^2; \frac{d^m u_2}{dt^m} &= \frac{d^m u_1}{dt^m} - L_0(u_0, u_1) - N_0(u_0, u_1) - D^\alpha u_1 \\
p^3; \frac{d^m u_3}{dt^m} &= \frac{d^m u_2}{dt^m} - L_0(u_0, u_1, u_2) - N_0(u_0, u_1, u_2) - D^\alpha u_2 \\
&\vdots
\end{aligned} \tag{2.21}$$

Çözümler yapılırken denklem (2.21)'deki konfigürasyondan faydalanılmıştır.

### 2.3.2.1 Orantılı Dereceli Doğrusal Olmayan Zamanla Değişmeyen Kesir Dereceli İntegro Diferansiyel Denklemin Homotopy Pertürbasyon Metodu ile Çözümü

Literatürde HPM ile birçok probleme yaklaşık çözümler üretilmektedir [85-88]. Bu bölümde ise Denklem 2.13 ve Denklem 2.14'de tanımlanmış olan orantılı dereceli doğrusal olmayan zamanla değişmeyen sistemlerinin durum uzay modeli için HPM ile bir analitik çözüm sunulmaktadır. Odibat ve arkadaşlarının [88]'de verdiği gibi KD doğrusal olmayan denklemlerin HPM ile çözümünde kullanılan denklemler aşağıdaki gibidir.

$$\frac{d^n x(t)}{dt^n} - A(t)x(t) - B(t)u(t) = p \left[ \frac{d^n x(t)}{dt^n} + f(t, x(t)) - D_t^\alpha x(t) \right] \tag{2.22}$$

veya

$$\frac{d^n x(t)}{dt^n} - B(t)u(t) = p \left[ \frac{d^n x(t)}{dt^n} + A(t)x(t) + f(t, x(t)) - D_t^\alpha x(t) \right] \tag{2.23}$$

Denklemden  $B(t)$  durum uzay modelinin doğrusal bölümü gösterirken  $A(t)$ , doğrusal olmayan bölümünü gösterir.  $u(t)$ , sistemin girişini göz önünde bulunduran tanımlı bir analitik fonksiyondur. Homotopy parametresi  $p$  ise

sürekli 0 ile 1 arasında değişmektedir.  $p = 0$ , olması durumunda, önceki bölümlerde de bahsedildiği gibi Denklem (2.22) doğrusal bir denklem olmaktadır ve aşağıdaki gibi hesaplanır.

$$\frac{d^n x(t)}{dt^n} + A(x(t)) = Bu(t) \quad (2.24)$$

aynı şekilde Denklem (2.23)'de doğrusal bir denklem olmaktadır ve aşağıdaki gibi gösterilir.

$$\frac{d^n x(t)}{dt^n} = Bu(t) \quad (2.25)$$

Denklem (2.22) ve (2.23)'deki temel yaklaşımlar göz önünde bulundurularak  $p$ 'nin güç serisi açılımı Denklem (2.26)'daki gibi olur.

$$x = x_0 + px_1 + p^2x_2 + p^3x_3 + \dots \quad (2.26)$$

Denklem (2.26)'nın Denklem (2.22) veya (2.23)'de yerine konulmasıyla  $p$ 'nin özdeş güç serisiyle ilgili denklemleri sırasıyla aşağıdaki gibi hesaplanır.

$$\begin{aligned} p^0; \frac{d^m x_0}{dt^m} + Ax(t) &= Bu(t) \\ p^1; \frac{d^m x_1}{dt^m} + Ax_0(t) &= \frac{d^m x_0}{dt^m} + f(t, x_0(t)) - D^\alpha x_0 \\ p^2; \frac{d^m x_2}{dt^m} + Ax_1(t) &= \frac{d^m x_1}{dt^m} + f(t, x_1(t)) - D^\alpha x_1 \\ p^3; \frac{d^m x_3}{dt^m} + Ax_2(t) &= \frac{d^m x_2}{dt^m} + f(t, x_2(t)) - D^\alpha x_2 \\ &\vdots \end{aligned} \quad (2.27)$$

$$\begin{aligned}
p^0; \frac{d^m x_0}{dt^m} &= Bu(t) \\
p^1; \frac{d^m x_1}{dt^m} &= \frac{d^m x_0}{dt^m} - Ax_0(t) - f(t, x_0(t)) - D^\alpha x_0 \\
p^2; \frac{d^m x_2}{dt^m} &= \frac{d^m x_1}{dt^m} - Ax_1(t) - f(t, x_1(t)) - D^\alpha x_1 \\
p^3; \frac{d^m x_3}{dt^m} &= \frac{d^m x_2}{dt^m} - Ax_2(t) - f(t, x_2(t)) - D^\alpha x_2 \\
&\vdots
\end{aligned} \tag{2.28}$$

Bu bölümde, Denklem (2.28) kullanılarak doğrusal zamanla değişmeyen KD integro diferansiyel ifadelerin çözümü gerçekleştirilir. Denklem (2.29)'da verildiği gibi orantılı dereceli doğrusal olmayan zamanla değişmeyen sistemi ele alalım.

$$\frac{d^\alpha x(t)}{dt^\alpha} = x(t) - x^2(t) - u(t) \tag{2.29}$$

Denklemden  $x(0) = 0$  olması şartıyla  $u(t) = 1$ , birim basamak girişidir [89-90]. Denklem (2.23) kullanılarak Denklem (2.29)'daki sistem için homotopy denklemi Denklem (2.30)'daki gibi elde edilir.

$$\frac{d^n x(t)}{dt^n} + 1 = p \left[ \frac{d^n x(t)}{dt^n} + x^2(t) - x(t) + D^\alpha x(t) \right] \tag{2.30}$$

$p$ 'nin güç serisi açılımı  $x = x_0 + px_1 + p^2x_2 + p^3x_3 + \dots$  şeklinde olduğu diğer bölümlerde belirtilmişti. İlgili bu açılım Denklem (2.30)'da yerine konulmasıyla Denklem (2.31)'de verilen güç serisi elde edilir.

$$\begin{aligned}
x_0; \frac{d^m x_0}{dt^m} &= f(t) \\
x_1; \frac{d^m x_1}{dt^m} &= \frac{d^m x_0}{dt^m} + x_0^2 - x_0 - D^\alpha x_0 \\
x_2; \frac{d^m x_2}{dt^m} &= \frac{d^m x_1}{dt^m} + 2x_0 x_1 - x_1 - D^\alpha x_1 \\
x_3; \frac{d^m x_3}{dt^m} &= \frac{d^m x_2}{dt^m} + x_1^2 - x_2 + 2x_0 x_1 - D^\alpha x_2 \\
&\vdots
\end{aligned} \tag{2.31}$$

Denklem (2.31)'in çözülmesiyle aşağıda sunulmuş olan güç serisi denklemleri hesaplanmıştır. Güç serisi 4. dereceye kadar açılmıştır. Derecenin miktarı artırılarak daha iyi sonuçlar elde edilebilir. Fakat yapılan çözümler manuel yapıldığından çok fazla işlem maliyeti gerektirir.

$$\begin{aligned}
x_0 &= -2t \\
x_1 &= -2t + \frac{4t^3}{3} + t^2 + \frac{2}{\Gamma(2+\alpha)} t^{(1+a)} \\
x_2 &= -2t + \frac{4t^3}{3} + t^2 + \frac{4}{\Gamma(2+\alpha)} t^{(1+a)} + \frac{7t^3}{18} - \frac{16t^5}{15} - \frac{4t^4}{3} - \frac{8}{\Gamma(2+\alpha)(3+\alpha)} \\
&\quad t^{(3+\alpha)} - \frac{2}{\Gamma(3+\alpha)} t^{(2+\alpha)} - \frac{8}{\Gamma(4+\alpha)} t^{(3+\alpha)} - \frac{2}{\Gamma(3+\alpha)} t^{(3+\alpha)} - \frac{2}{\Gamma(2+2\alpha)} t^{(1+2\alpha)} \\
x_3 &= -2t + \frac{8t^3}{3} + t^2 + \frac{2}{\Gamma(2+\alpha)} t^{(1+a)} + \frac{7t^3}{18} - \frac{16t^5}{15} - \frac{4t^4}{3} - \frac{8}{\Gamma(2+\alpha)(3+\alpha)} t^{(3+\alpha)} \\
&\quad - \frac{2}{\Gamma(3+\alpha)} t^{(2+\alpha)} - \frac{8}{\Gamma(4+\alpha)} t^{(3+\alpha)} - \frac{2}{\Gamma(3+\alpha)} t^{(3+\alpha)} - \frac{2}{\Gamma(2+2\alpha)} t^{(1+2\alpha)} - t^2 \\
&\quad + \frac{9t^4}{4} + \frac{2t^3}{3} + \frac{2}{\Gamma(3+a)} t^{(1+a)} + 2t^4 - \frac{8t^6}{45} - \frac{2t^5}{5} \\
&\quad + \left( -\frac{8}{\Gamma(2+\alpha)(3+a)(4+a)} - \frac{2}{\Gamma(4+\alpha)} - \frac{8}{\Gamma(5+\alpha)} - \frac{2}{\Gamma(3+\alpha)(4+a)} \right) t^{(4+a)} \\
&\quad - \frac{2}{\Gamma(3+2\alpha)} t^{(2+2\alpha)} + \frac{16t^7}{63} + \frac{t^5}{5} + \frac{4}{(\Gamma(2+2\alpha))^2(3+2\alpha)} t^{(3+2\alpha)} + \frac{8t^3}{3} \\
&\quad - \frac{16t^5}{15} - 2t^4 - \frac{16}{\Gamma(2+\alpha)(3+a)} t^{(3+a)} - \frac{32t^5}{15} - \frac{128t^7}{105} - \frac{2t^6}{3} \\
&\quad + \left( \frac{32}{\Gamma(2+\alpha)(3+a)(5+a)} + \frac{32}{\Gamma(4+\alpha)(5+a)} + \frac{8}{\Gamma(3+\alpha)(5+a)} \right) t^{(5+a)} \\
&\quad + \frac{2t^6}{9} + \frac{4t^5}{15} + \frac{16}{\Gamma(3+\alpha)(4+a)} t^{(4+a)} + \frac{8}{\Gamma(2+2\alpha)(3+2\alpha)} t^{(3+2\alpha)} + \frac{1}{\Gamma(2+\alpha)} t^{(1+a)} \\
&\quad - \frac{8}{\Gamma(4+\alpha)} t^{(3+a)} - \frac{2}{\Gamma(3+\alpha)} t^{(2+a)} - \frac{2}{\Gamma(2+2\alpha)} t^{(1+2\alpha)} - \frac{16}{\Gamma(4+\alpha)} t^{(3+a)} + \\
&\quad \frac{108}{\Gamma(6+\alpha)} t^{(5+a)} + \frac{24}{\Gamma(5+\alpha)} t^{(4+a)} + \frac{8}{\Gamma(2+\alpha)(3+\alpha)(4+\alpha)} t^{(3+2\alpha)} - \frac{2}{\Gamma(3+\alpha)} t^{(2+\alpha)} \\
&\quad + \frac{24}{3\Gamma(5+\alpha)} t^{(4+a)} + \frac{2}{\Gamma(4+\alpha)} t^{(3+a)} + \frac{2}{\Gamma(3-2\alpha)} t^{(2+2\alpha)} - \frac{2}{\Gamma(2+2\alpha)} t^{(1+a)} \\
&\quad - \frac{8}{\Gamma(4+2\alpha)} t^{(3+2\alpha)} + \frac{2\Gamma(4+\alpha)}{\Gamma(3+\alpha)\Gamma(4+2\alpha)} t^{(3+2\alpha)} + \frac{2}{\Gamma(2+3\alpha)} t^{(1+3\alpha)} \\
&\quad \vdots
\end{aligned}$$

İlk 4 derece için elde edilmiş olan çözüm  $p = 1$  olması koşuluyla denklem (2.32) kullanılarak ifadenin yaklaşık çözümü hesaplanır.

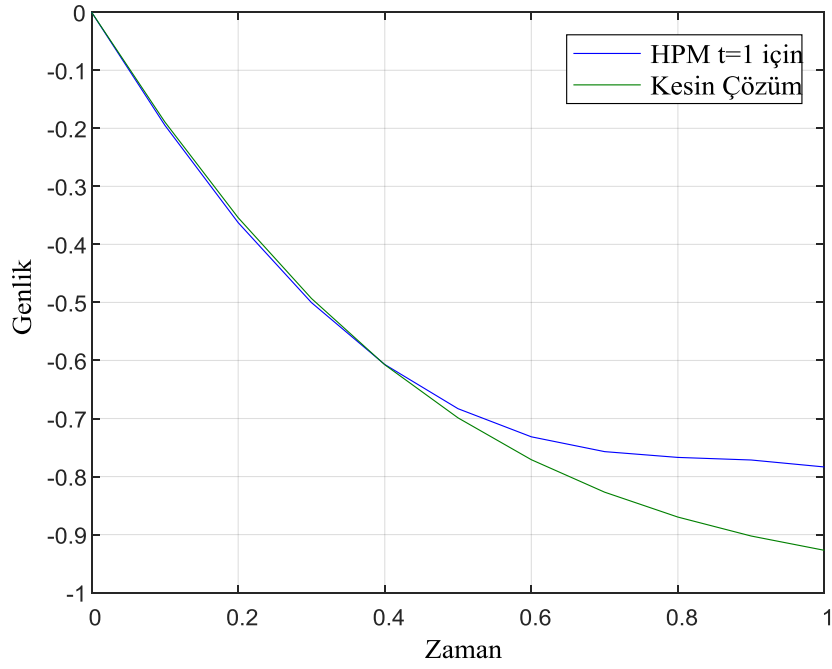
$$x = x_0 + x_1 + x_2 + x_3 \quad (2.32)$$

Yapılan çözümün doğruluğunu test etmek için şöyle bir yöntem izlenmiştir. Öncelikle elde edilen tüm denklemler MATLAB ortamına aktarılıp zaman ve diğer denklem değişkenleri için parametrik hale getirilmiştir. Böylece istenilen zaman diliminin istenilen kat sayı konfigürasyonuna göre sistem cevapları hesaplanır. Bundan dolayı tam sayı dereceli integro diferansiyel ifadeler analitik olarak çözülebilir.

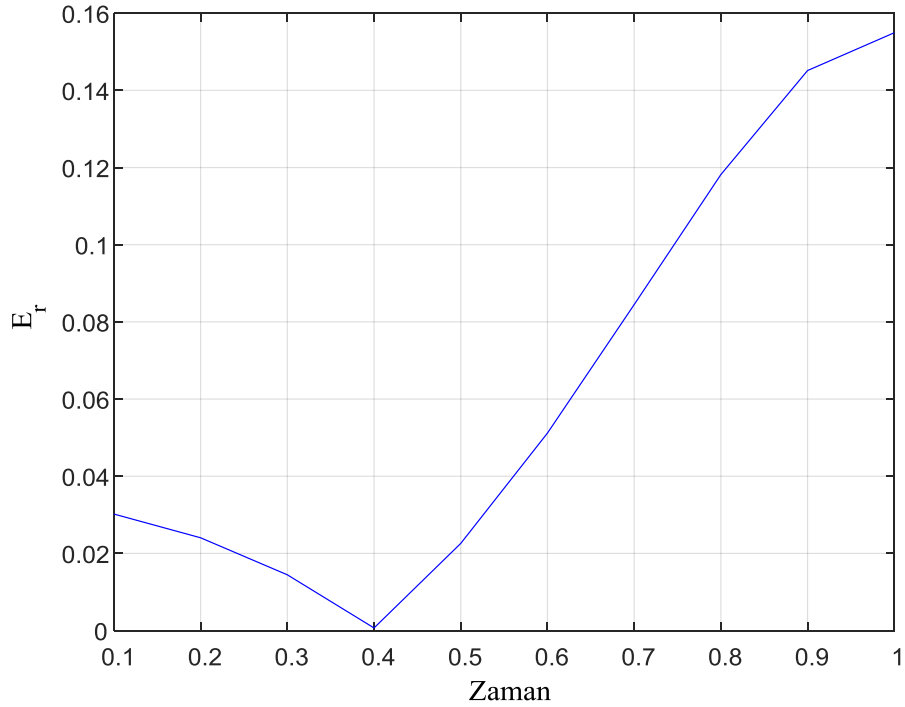
Kesir dereceli ifadenin derecesinin 1 olması durumu yani  $\alpha=1$  olması ifadeyi tam sayı dereceli yapar. Elde edilen sonuçlarda  $\alpha=1$  durumunda üretilen denklemler ile tam sayı dereceli sonuçlar karşılaştırılmıştır. Böylece üretilen KD denklemlerin doğruluğu test edilir.

Şekil 2.1’de aynı durum söz konusudur. HPM sonucunun  $\alpha=1$  olma durumuyla ilgili denklemin tam sayı dereceli sonuçları grafiksel olarak  $t \in [0,1]$  için karşılaştırılmıştır. Görüldüğü üzere HPM çözümü tam sayı dereceli çözüme büyük ölçüde yakınsamaktadır. Şekil 2.2’de ise  $E_r = |x_{HPM} - x_{kesin}| / x_{kesin}$  şeklinde tanımlanan hata fonksiyonuna göre hesaplanan hata değerinin zamana göre değişimi gösterilmiştir. Şekilden de anlaşılacağı üzere güç serisinin derecesi ne kadar artılırsa hata o kadar düşmektedir. Zaten bu kural güç serileri için genel geçer bir yaklaşımdır. İlgili ifade ne kadar fazla güç serisine açılırsa gerçek sonuca o kadar yakınsayabilir.

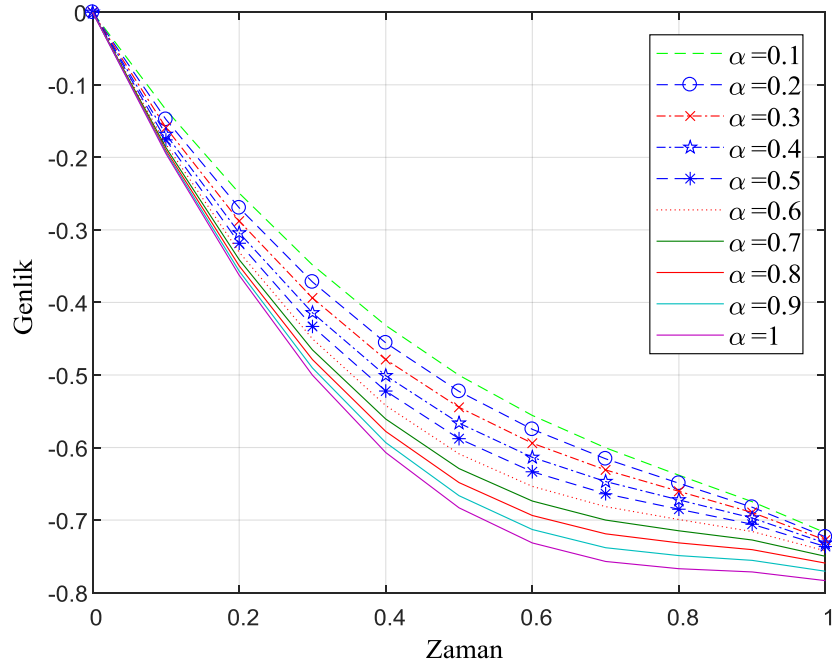
Şekil 2.3’de ise farklı  $\alpha$  değerleri için Denklem 2.30’un ürettiği sonuçlar gösterilmektedir. Şekilden de görüldüğü üzere  $\alpha$  değerinin artırılmasıyla gerçek çözüme yakınsandığı görülür. Bu da yapılmış olan çözümün doğruluğunu bir kez daha teyit etmektedir. Farklı  $\alpha$  değerlerine ilişkin ilgili denklemin nümerik sonuçları Çizelge 2.2’de sunulmaktadır.



**Şekil 2.1** Doğrusal olmayan zamanla değişmeyen sistemin  $t \in [0,1]$  için kesin çözüm ile HPM çözümünün karşılaştırılması



**Şekil 2.2** Doğrusal olmayan zamanla değişmeyen sistemin HPM çözümünün göreceli hata değeri



**Şekil 2.3** Doğrusal olmayan zamanla değişmeyen sistemin farklı  $\alpha$  değerleri için üretmiş olduğu cevapların karşılaştırılması ( $0 < \alpha < 1$ )

**Çizelge 2.2** Doğrusal olmayan zamanla değişmeyen sistemin farklı  $\alpha$  değerlerine göre HPM ile elde edilen nümerik sonuçları

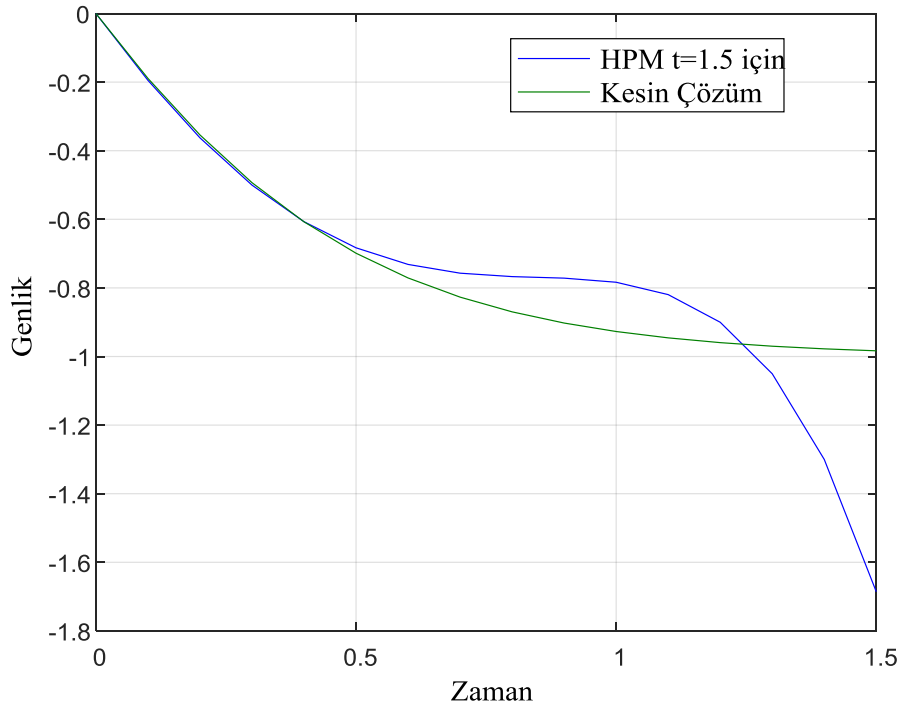
	$t=0$	$t=0.1$	$t=0.2$	$t=0.3$	$t=0.4$	$t=0.5$	$t=0.6$	$t=0.7$	$t=0.8$	$t=0.9$	$t=1$
$\alpha=0.1$	0	-0.1343	-0.2493	-0.3483	-0.4317	-0.5004	-0.5558	-0.6004	-0.6382	-0.6746	-0.7175
$\alpha=0.2$	0	-0.1477	-0.2696	-0.3718	-0.4557	-0.5227	-0.5749	-0.6155	-0.6492	-0.6823	-0.7231
$\alpha=0.3$	0	-0.1590	-0.2879	-0.3938	-0.4787	-0.5446	-0.5940	-0.6309	-0.6604	-0.6895	-0.7276
$\alpha=0.4$	0	-0.1684	-0.3042	-0.4143	-0.5009	-0.5663	-0.6135	-0.6469	-0.6722	-0.6971	-0.7318
$\alpha=0.5$	0	-0.1759	-0.3184	-0.4331	-0.5221	-0.5878	-0.6334	-0.6637	-0.6851	-0.7058	-0.7365
$\alpha=0.6$	0	-0.1819	-0.3307	-0.4502	-0.5421	-0.6088	-0.6536	-0.6815	-0.6993	-0.7158	-0.7424
$\alpha=0.7$	0	-0.1866	-0.3410	-0.4654	-0.5607	-0.6290	-0.6738	-0.7000	-0.7148	-0.7275	-0.7500
$\alpha=0.8$	0	-0.1901	-0.3497	-0.4788	-0.5778	-0.6483	-0.6937	-0.7190	-0.7314	-0.7408	-0.7593
$\alpha=0.9$	0	-0.1928	-0.3568	-0.4904	-0.5932	-0.6664	-0.7130	-0.7381	-0.7490	-0.7555	-0.7706
$\alpha=1$	0	-0.1948	-0.3626	-0.5003	-0.6069	-0.6831	-0.7315	-0.7571	-0.7670	-0.7715	-0.7835

Şekil 2.4’de ise  $t \in [0,1.5]$  için HPM ile kesin çözümü karşılaştırmaktadır. Şekilden de anlaşıldığı üzere  $t > 1$  olması durumunda 4. dereceden güç serisi açılımı için göreceli hata önemli bir ölçüde artış gösterir. Güç serisinin derece miktarının artırmak bu hatayı düşürecektir. Çünkü diğer güç serilerinde de olduğu gibi bu örnekte de seriyi daha fazla açmak sonucu gerçek sonuca daha fazla yakınsamasını sağlar.

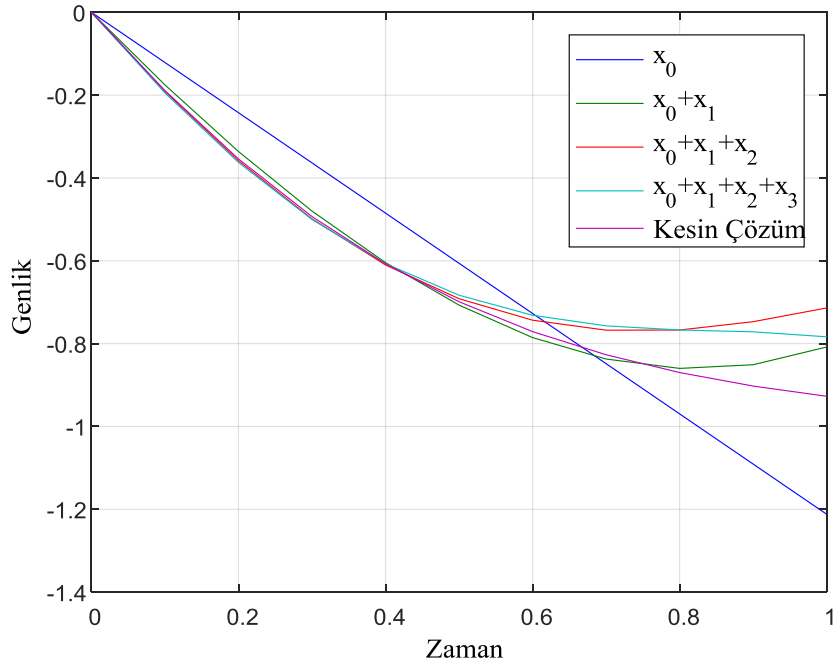
Bundan dolayı Şekil 2.5’de güç serisinin her teriminin gerçek sonuca yakınsama trendi gösterilmiştir. Şekilden de görüldüğü gibi seriye eklenen terim artıkça gerçek çözüme daha fazla yakınsanılır. Bu yakınsamının diğer bir ispatı ise her bir terim eklendikçe karesel hatada ki düşüştür (Şekil 2.6). Karesel hata

$$MSE = \frac{1}{T} \sum_0^T (x_{HPM} - x_{exact})^2$$

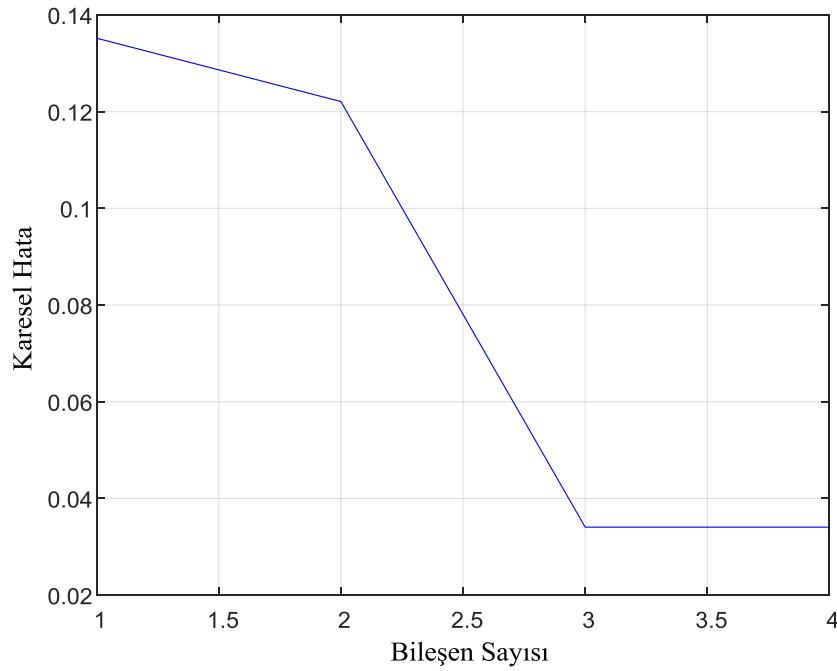
formülüyle hesaplanmış ve formülde  $T = 1$ , alınmıştır.



**Şekil 2.4** Doğrusal olmayan zamanla değişmeyen sistemin  $t \in [0,1.5]$  için HPM ve kesin çözümün karşılaştırılması



**Şekil 2.5** Doğrusal olmayan zamanla değişmeyen sistemin bileşen sayısının artırılmasıyla elde edilen sonuçların karşılaştırılması



**Şekil 2.6** Doğrusal olmayan zamanla değişmeyen sistemin bileşen sayısının artırılmasıyla elde edilen hata fonksiyonunun değişimi

### 2.3.2.2 Orantılı Dereceli Doğrusal Olmayan Zamanla Değişen Kesir Dereceli İntegro Diferansiyel Denklemin Homotopy Pertürbasyon Yöntemi ile Çözümü

Bu bölümde Denklem 2.33'de verilen doğrusal olmayan zamanla değişen bir sistem kullanılmaktadır. Denklemden de görüldüğü gibi değişkenlerimizin katsayıları da zamanla değişmektedir. Bu şekilde bir önceki bölümde belirtmiş olduğumuz denklemleri kullanmakla beraber türetmiş olduğumuz matematiksel denklem sayısı exponansiyel bir şekilde artış göstermektedir.

$$\frac{d^\alpha x(t)}{dt^\alpha} = t^5 - 2t^4 x(t) + t^3 x^2(t) + u(t) \quad (2.33)$$

Bir önceki örnekte olduğu gibi  $x(0) = 0$  için birim basamak girişimiz  $u(t) = 1$ 'dir [90]. İlgili denklemin HPM çözümü Denklem (2.34)'deki gibi türetilir.

$$\frac{d^n x(t)}{dt} + 1 + t^5 = p \left[ \frac{d^n x(t)}{dt} - 2t^4 x(t) + t^3 x^2(t) - D^\alpha x(t) \right] \quad (2.34)$$

$x = x_0 + px_1 + p^2 x_2 + p^3 x_3 + \dots$  güç serisinin Denklem (2.34)'de yerine konulmasıyla Denklem (2.35)'deki güç serisi denklemleri elde edilir.

$$\begin{aligned} x_0; \frac{d^m x_0}{dt^m} &= 1 + t^5 \\ x_1; \frac{d^m x_1}{dt^m} &= \frac{d^m x_0}{dt^m} + t^3 x_0^2 - 2t^4 x_0 - D^\alpha x_0 \\ x_2; \frac{d^m x_2}{dt^m} &= \frac{d^m x_1}{dt^m} + 2t^3 x_0 x_1 - 2t^4 x_1 - D^\alpha x_1 \\ x_3; \frac{d^m x_3}{dt^m} &= \frac{d^m x_2}{dt^m} + t^3 x_1^2 - 2t^4 x_2 + 2t^3 x_1 - D^\alpha x_2 \\ &\vdots \end{aligned} \quad (2.35)$$

Denklem 2.35'in Denklem (2.33)'de kullanılmasıyla ilgili denklemin güç serisi çözümünü aşağıdaki gibi yapılır.

$$\begin{aligned}
x_0 &= t + \frac{t^6}{6} \\
x_1 &= t + \frac{t^{16}}{576} - \frac{1}{\Gamma(2+\alpha)} t^{(1+\alpha)} - \frac{720}{\Gamma(7+\alpha)} t^{(6+\alpha)} \\
x_2 &= t + \frac{t^{16}}{576} - \frac{1}{\Gamma(2+\alpha)} t^{(1+\alpha)} - \frac{720}{\Gamma(7+\alpha)} t^{(6+\alpha)} - \frac{t^{21}}{6048} + \frac{720}{\Gamma(7+\alpha)(11+\alpha)} t^{(11+\alpha)} \\
&\quad + \frac{t^{11}}{3168} - \frac{2}{\Gamma(2+\alpha)(6+a)} t^{(6+a)} - \frac{1440}{\Gamma(7+\alpha)(11+a)} t^{(11+a)} + \frac{t^{11}}{33} + \frac{t^{16}}{27648} \\
&\quad - \frac{1}{3\Gamma(2+\alpha)(11+a)} t^{(11+a)} - \frac{240}{\Gamma(7+\alpha)(16+a)} t^{(16+a)} - \frac{1}{\Gamma(2+\alpha)} t^{(1+\alpha)} \\
&\quad - \frac{\Gamma(17)}{\Gamma(17+\alpha)576} t^{(16+a)} + \frac{1}{\Gamma(3+\alpha)} t^{(1+2a)} + \frac{720}{\Gamma(8+\alpha)} t^{(7+a)} \\
x_3 &= t + \frac{t^{16}}{576} - \frac{1}{\Gamma(2+\alpha)} t^{(1+\alpha)} - \frac{720}{\Gamma(7+\alpha)} t^{(6+\alpha)} - \frac{t^{21}}{6048} + \frac{720}{\Gamma(7+\alpha)(11+\alpha)} t^{(11+\alpha)} \\
&\quad + \frac{t^{11}}{3168} - \frac{2}{\Gamma(2+\alpha)(6+a)} t^{(6+a)} - \frac{1440}{\Gamma(7+\alpha)(11+a)} t^{(11+a)} + \frac{t^{11}}{33} + \frac{t^{16}}{27648} \\
&\quad - \frac{1}{3\Gamma(2+\alpha)(11+a)} t^{(11+a)} - \frac{240}{\Gamma(7+\alpha)(16+a)} t^{(16+a)} - \frac{1}{\Gamma(2+\alpha)} t^{(1+\alpha)} \\
&\quad - \frac{\Gamma(17)}{\Gamma(17+\alpha)576} t^{(16+a)} + \frac{1}{\Gamma(3+\alpha)} t^{(1+2a)} + \frac{720}{\Gamma(8+\alpha)} t^{(7+a)} - \frac{t^6}{3} - \frac{t^{21}}{6048} \\
&\quad + \frac{2}{\Gamma(2+\alpha)(6+a)} t^{(6+a)} + \frac{1440}{\Gamma(7+\alpha)(11+a)} t^{(11+a)} \\
&\quad + \frac{t^{26}}{78624} - \frac{1440}{\Gamma(7+\alpha)(11+a)(16+a)} t^{(16+a)} - \frac{t^{16}}{25344} + \frac{2880}{\Gamma(7+\alpha)(11+a)(16+a)} t^{(16+a)} \\
&\quad - \frac{t^{16}}{264} - \frac{t^{21}}{290304} + \frac{2}{3\Gamma(2+\alpha)(11+a)(16+a)} t^{(16+a)} + \frac{480}{\Gamma(7+\alpha)(16+a)(21+\alpha)} t^{(21+a)} \\
&\quad + \frac{2}{\Gamma(2+\alpha)(16+a)} t^{(6+a)} + \frac{\Gamma(17)}{\Gamma(17+\alpha)288(21+\alpha)} t^{(16+a)} - \frac{2}{\Gamma(3+\alpha)(6+2\alpha)} t^{(6+2a)} \\
&\quad - \frac{1440}{\Gamma(8+\alpha)(12+\alpha)} t^{(12+a)} + \frac{t^5}{5} + \frac{t^{20}}{576} - \frac{1}{\Gamma(2+\alpha)(5+\alpha)} t^{(5+a)} - \frac{720}{\Gamma(7+\alpha)(10+\alpha)} t^{(10+a)} \\
&\quad + \frac{t^6}{6} + \frac{t^{36}}{11943936} - \frac{1}{[\Gamma(2+\alpha)]^2(6+2\alpha)} t^{(6+2\alpha)} - \frac{720^2}{[\Gamma(2+\alpha)]^2(16+2\alpha)} t^{(16+2\alpha)} + \\
&\quad \frac{t^{21}}{6048} - \frac{t^{(21+\alpha)}}{288(2+\alpha)(21+\alpha)} - \frac{1440t^{(11+\alpha)}}{\Gamma(7+\alpha)(11+\alpha)} - \frac{2t^{(5+\alpha)}}{\Gamma(2+\alpha)(5+\alpha)}
\end{aligned}$$

$$\begin{aligned}
& + \frac{1440t^{(11+2\alpha)}}{\Gamma(7+\alpha)\Gamma(2+\alpha)(11+2\alpha)} - \frac{1440t^{(26+\alpha)}}{576\Gamma(7+\alpha)(26+\alpha)} + \frac{t^6}{3} + \frac{t^{11}}{33} + \frac{t^{21}}{6048} + \frac{t^{26}}{44928} \\
& - \frac{2t^{(6+\alpha)}}{\Gamma(2+\alpha)(6+\alpha)} - \frac{t^{(10+\alpha)}}{3\Gamma(2+\alpha)(11+\alpha)} - \frac{1440t^{(11+\alpha)}}{\Gamma(7+\alpha)(11+\alpha)} - \frac{720t^{(16+\alpha)}}{3\Gamma(7+\alpha)(16+\alpha)} \\
& - \frac{t^{26}}{139104} - \frac{t^{31}}{562464} + \frac{4t^{(11+\alpha)}}{\Gamma(2+\alpha)(11+\alpha)(6+\alpha)} - \frac{2t^{(16+\alpha)}}{3\Gamma(2+\alpha)(16+\alpha)^2} \\
& + \frac{1440t^{(16+\alpha)}}{\Gamma(7+\alpha)(11+\alpha)(16+\alpha)} - \frac{720t^{(21+\alpha)}}{3\Gamma(7+\alpha)(11+\alpha)(21+\alpha)} + \frac{t^{16}}{25344} + \frac{t^{21}}{199584} \\
& + \frac{1440t^{(11+\alpha)}}{\Gamma(2+\alpha)(11+\alpha)(6+\alpha)} - \frac{2t^{(13+\alpha)}}{3\Gamma(6+\alpha)(13+\alpha)} - \frac{2880t^{(16+\alpha)}}{\Gamma(7+\alpha)(11+\alpha)(16+\alpha)} \\
& - \frac{1440t^{(21+\alpha)}}{3\Gamma(7+\alpha)(11+\alpha)(21+\alpha)} + \frac{t^{16}}{264} + \frac{t^{21}}{2079} + \frac{t^{21}}{290304} + \frac{t^{26}}{2156544} - \frac{2t^{(16+\alpha)}}{3\Gamma(2+\alpha)(11+\alpha)(16+\alpha)} \\
& - \frac{t^{(21+\alpha)}}{480t^{(21+\alpha)}} - \frac{80t^{(21+\alpha)}}{80t^{(21+\alpha)}} \\
& - \frac{9\Gamma(7+\alpha)(11+\alpha)(16+\alpha)}{2t^{(21+\alpha)}} - \frac{\Gamma(3+\alpha)(16+\alpha)(21+\alpha)}{t^{(11+\alpha)}} - \frac{\Gamma(3+\alpha)(16+\alpha)(26+\alpha)}{\Gamma(17)t^{(21+\alpha)}} - \frac{\Gamma(17)t^{(26+\alpha)}}{\Gamma(17)t^{(26+\alpha)}} + \\
& \frac{\Gamma(2+\alpha)(6+\alpha)}{2t^{(6+2\alpha)}} - \frac{3\Gamma(2+\alpha)(11+\alpha)}{t^{(11+2\alpha)}} - \frac{\Gamma(17+\alpha)288(21+\alpha)}{1440t^{(17+\alpha)}} - \frac{1728\Gamma(17+\alpha)(26+\alpha)}{240t^{(17+\alpha)}} - \frac{t^{(1+\alpha)}}{t^{(1+\alpha)}} \\
& \frac{\Gamma(3+\alpha)(6+2\alpha)}{\Gamma(3+\alpha)(6+2\alpha)} + \frac{3\Gamma(3+\alpha)(11+2\alpha)}{3\Gamma(3+\alpha)(11+2\alpha)} + \frac{\Gamma(8+\alpha)(12+\alpha)}{\Gamma(8+\alpha)(12+\alpha)} - \frac{\Gamma(8+\alpha)(17+\alpha)}{\Gamma(8+\alpha)(17+\alpha)} - \frac{\Gamma(2+\alpha)}{\Gamma(2+\alpha)} \\
& - \frac{\Gamma(17)t^{(16+\alpha)}}{576\Gamma(17+\alpha)} + \frac{\Gamma(2+\alpha)t^{(26+\alpha)}}{\Gamma(2+2\alpha)} - \frac{720t^{(6+\alpha)}}{\Gamma(7+2\alpha)} + \frac{\Gamma(22)t^{(21+\alpha)}}{\Gamma(22+2\alpha)6048} \\
& - \frac{\Gamma(7+\alpha)t^{(6+2\alpha)}}{\Gamma(2+\alpha)\Gamma(7+2\alpha)(6+\alpha)} - \frac{720\Gamma(12+\alpha)t^{(11+2\alpha)}}{\Gamma(7+\alpha)\Gamma(12+2\alpha)(11+\alpha)} - \frac{\Gamma(12)t^{(11+\alpha)}}{\Gamma(12+\alpha)3168} - \frac{2\Gamma(7+\alpha)t^{(6+2\alpha)}}{\Gamma(2+\alpha)\Gamma(7+2\alpha)(6+\alpha)} \\
& + \frac{1440\Gamma(12+\alpha)t^{(11+2\alpha)}}{\Gamma(7+\alpha)\Gamma(12+2\alpha)(11+\alpha)} - \frac{\Gamma(12)t^{(11+\alpha)}}{\Gamma(12+\alpha)33} - \frac{\Gamma(12+\alpha)t^{(11+\alpha)}}{3\Gamma(12+\alpha)\Gamma(12+\alpha)(11+\alpha)} + \\
& \frac{240\Gamma(17+\alpha)t^{(16+2\alpha)}}{\Gamma(3+\alpha)\Gamma(17+2\alpha)(16+\alpha)} + \frac{\Gamma(2+\alpha)t^{(1+2\alpha)}}{\Gamma(2+\alpha)\Gamma(2+2\alpha)} + \frac{2t^{(21+\alpha)}}{\Gamma(2+\alpha)(6+\alpha)} \\
& - \frac{t^{(11+\alpha)}}{3\Gamma(2+\alpha)(11+\alpha)} - \frac{\Gamma(17)t^{(21+\alpha)}}{\Gamma(17+\alpha)288(21+\alpha)} - \frac{\Gamma(17)t^{(26+\alpha)}}{1728\Gamma(17+\alpha)(26+\alpha)} + \frac{2t^{(6+2\alpha)}}{\Gamma(3+\alpha)(6+2\alpha)} \\
& + \frac{t^{(11+2\alpha)}}{t^{(11+2\alpha)}} + \frac{1440t^{(17+\alpha)}}{\Gamma(3+\alpha)(11+2\alpha)} - \frac{240t^{(17+\alpha)}}{\Gamma(8+\alpha)(12+\alpha)} - \frac{t^{(1+\alpha)}}{\Gamma(8+\alpha)(17+\alpha)} - \frac{\Gamma(2+\alpha)}{\Gamma(2+\alpha)} \\
& - \frac{\Gamma(17)t^{(16+\alpha)}}{576\Gamma(17+\alpha)} + \frac{\Gamma(2+\alpha)t^{(26+\alpha)}}{\Gamma(2+2\alpha)} - \frac{720t^{(6+\alpha)}}{\Gamma(7+2\alpha)} + \frac{\Gamma(22)t^{(21+\alpha)}}{\Gamma(22+2\alpha)6048} - \\
& \frac{\Gamma(7+\alpha)t^{(6+2\alpha)}}{\Gamma(2+\alpha)\Gamma(7+2\alpha)(6+\alpha)} - \frac{720\Gamma(12+\alpha)t^{(11+2\alpha)}}{\Gamma(7+\alpha)\Gamma(12+2\alpha)(11+\alpha)} - \frac{\Gamma(12)t^{(11+\alpha)}}{\Gamma(12+\alpha)3168} \\
& - \frac{2\Gamma(7+\alpha)t^{(6+2\alpha)}}{\Gamma(2+\alpha)\Gamma(7+2\alpha)(6+\alpha)} + \frac{1440\Gamma(12+\alpha)t^{(11+2\alpha)}}{\Gamma(7+\alpha)\Gamma(12+2\alpha)(11+\alpha)} - \frac{\Gamma(12)t^{(11+\alpha)}}{\Gamma(12+\alpha)33} - \frac{\Gamma(17)t^{(16+\alpha)}}{\Gamma(17+\alpha)27648} + \\
& \frac{\Gamma(12+\alpha)t^{(11+\alpha)}}{3\Gamma(12+\alpha)\Gamma(12+\alpha)(11+\alpha)} + \frac{240\Gamma(17+\alpha)t^{(16+2\alpha)}}{\Gamma(3+\alpha)\Gamma(17+2\alpha)(16+\alpha)} + \frac{t^{(1+2\alpha)}}{\Gamma(2+2\alpha)} - \frac{\Gamma(17)t^{(16+\alpha)}}{\Gamma(17+\alpha)27648}
\end{aligned}$$

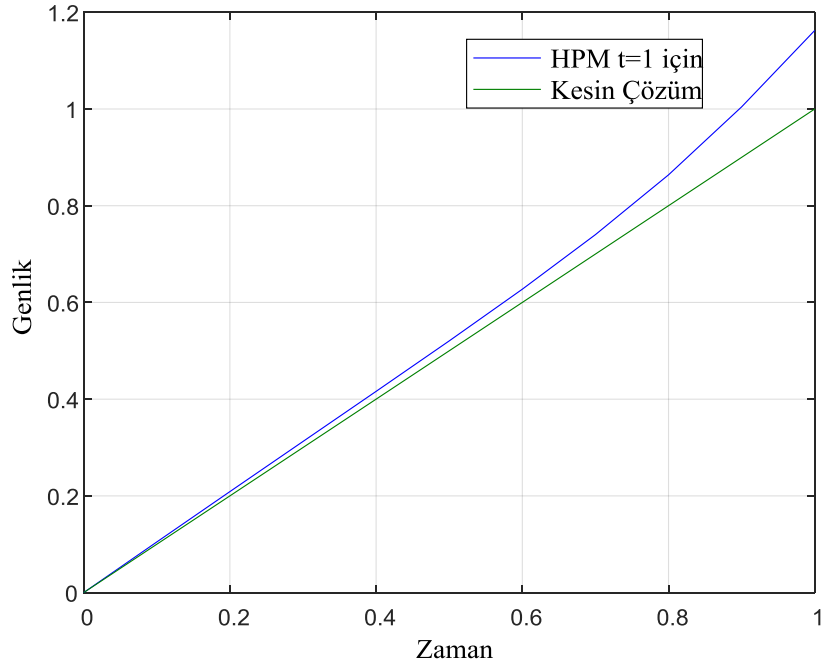
$$\begin{aligned}
& + \frac{\Gamma(17)\Gamma(12+\alpha)t^{(16+2\alpha)}}{576\Gamma(17+\alpha)\Gamma(12+2\alpha)} - \frac{\Gamma(2+2\alpha)t^{(1+3\alpha)}}{\Gamma(3+\alpha)\Gamma(2+2\alpha)} - \frac{720t^{(7+2\alpha)}}{576\Gamma(8+2\alpha)} \\
& - \frac{720t^{(6+\alpha)}}{\Gamma(7+2\alpha)} + \frac{\Gamma(22)t^{(21+\alpha)}}{\Gamma(22+2\alpha)6048} - \frac{\Gamma(7+\alpha)t^{(6+2\alpha)}}{\Gamma(2+\alpha)\Gamma(7+2\alpha)(6+\alpha)} - \frac{720\Gamma(12+\alpha)t^{(11+2\alpha)}}{\Gamma(7+\alpha)\Gamma(12+2\alpha)(11+\alpha)} \\
& - \frac{\Gamma(12)t^{(11+\alpha)}}{\Gamma(12+\alpha)3168} - \frac{2\Gamma(7+\alpha)t^{(6+2\alpha)}}{\Gamma(2+\alpha)\Gamma(7+2\alpha)(6+\alpha)} + \frac{1440\Gamma(12+\alpha)t^{(11+2\alpha)}}{\Gamma(7+\alpha)\Gamma(12+2\alpha)(11+\alpha)} \\
& - \frac{\Gamma(12)t^{(11+\alpha)}}{\Gamma(12+\alpha)33} - \frac{\Gamma(17)t^{(16+\alpha)}}{\Gamma(17+\alpha)27648} + \frac{\Gamma(12+\alpha)t^{(11+\alpha)}}{3\Gamma(12+\alpha)\Gamma(12+\alpha)(11+\alpha)} + \\
& \frac{240\Gamma(17+\alpha)t^{(16+2\alpha)}}{\Gamma(3+\alpha)\Gamma(17+2\alpha)(16+\alpha)} + \frac{t^{(1+2\alpha)}}{\Gamma(2+2\alpha)} + \frac{\Gamma(17)\Gamma(12+\alpha)t^{(16+2\alpha)}}{576\Gamma(17+\alpha)\Gamma(12+2\alpha)} \\
& - \frac{\Gamma(2+2\alpha)t^{(1+3\alpha)}}{\Gamma(3+\alpha)\Gamma(2+2\alpha)} - \frac{720t^{(7+2\alpha)}}{576\Gamma(8+2\alpha)} \\
& \vdots
\end{aligned}$$

$p = 1$  olması şartıyla  $x(t)$  güç serisi açılımı Denklem (2.36) ile hesaplanır.

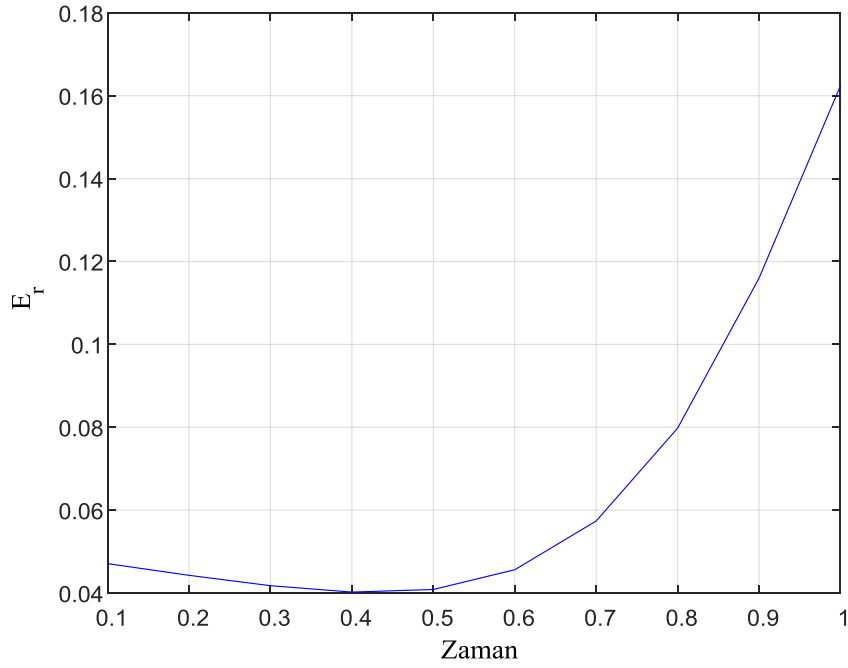
$$x = x_0 + x_1 + x_2 + x_3 \quad (2.36)$$

Şekil 2.7’de Denklem (2.33) için HPM kesin çözümü  $\alpha = 1$  ve  $t \in [0,1]$  şartları altında karşılaştırmalı olarak sunulmaktadır. Şekilden de anlaşılacağı üzere HPM  $E_r < 0.2$  şartı altında tatmin edici sonuçlar üretebilir.

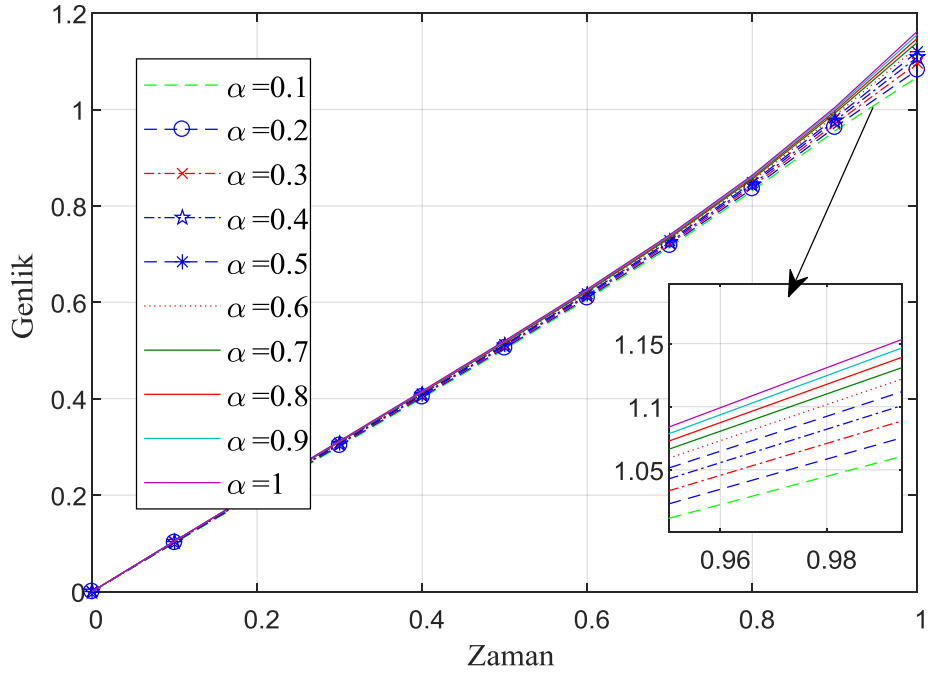
Şekil 2.8’de ise HPM çözümü için zamanla hatanın değişim grafiği gösterilmektedir. Zamanın artmasıyla hatanın da exponansiyel olarak arttığı gözlemlenmiştir. Şekil 2.9’da ise farklı  $\alpha$  değerlerine ilişkin HPM nin üretmiş olduğu cevaplar karşılaştırılmaktadır. Çizelge 2.3’de ise her  $\alpha$  değeri için HPM çözümünün üretmiş olduğu nümerik değerler gösterilmiştir. Şekil 2.10’da güç serisinin eleman sayısının artırılmasıyla gerçek çözüme yakınsama durumu sunulmuştur. Şekil 2.11’de ise güç serisine eleman eklendiği esnada ortaya çıkan hata değişim grafiği verilmiştir.



**Şekil 2.7** Doğrusal olmayan zamanla değişen sistemin  $t \in [0,1]$  için kesin çözüm ile HPM çözümünün karşılaştırılması



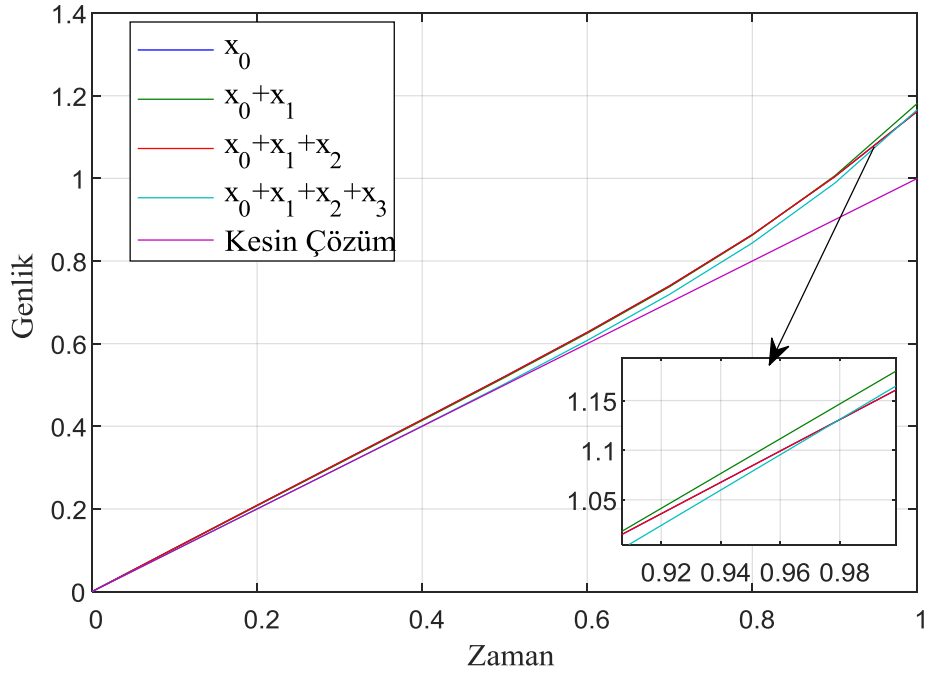
**Şekil 2.8** Doğrusal olmayan zamanla değişen sistemin HPM çözümünün hata değeri



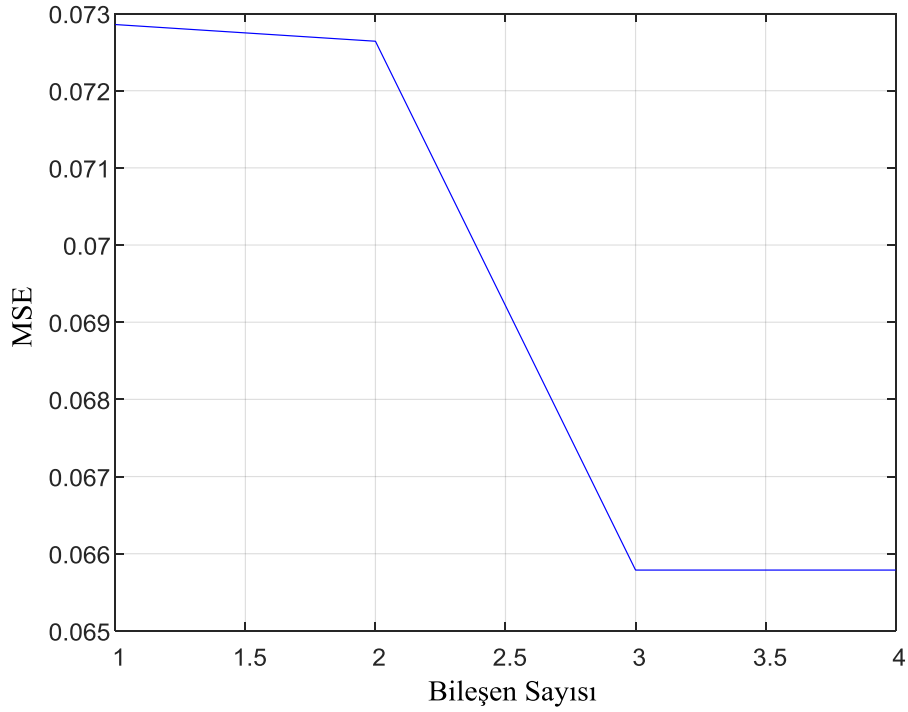
**Şekil 2.9** Doğrusal olmayan zamanla değişen sistemin farklı  $\alpha$  değerleri için üretmiş olduğu cevapların karşılaştırılması ( $0 < \alpha < 1$ )

**Çizelge 2.3** Doğrusal olmayan zamanla değişen sistemin farklı  $\alpha$  değerlerine göre HPM ile elde edilen nümerik sonuçları

	$t=0$	$t=0.1$	$t=0.2$	$t=0.3$	$t=0.4$	$t=0.5$	$t=0.6$	$t=0.7$	$t=0.8$	$t=0.9$	$t=1$
$\alpha=0.1$	0	0.1008	0.2011	0.3012	0.4014	0.5027	0.6062	0.7146	0.8310	0.9554	1.0672
$\alpha=0.2$	0	0.1018	0.2027	0.3033	0.4039	0.5054	0.6093	0.7183	0.8359	0.9630	1.0823
$\alpha=0.3$	0	0.1026	0.2041	0.3051	0.4061	0.5080	0.6123	0.7218	0.8404	0.9700	1.0961
$\alpha=0.4$	0	0.1032	0.2052	0.3067	0.4081	0.5103	0.6150	0.7251	0.8446	0.9764	1.1086
$\alpha=0.5$	0	0.1036	0.2062	0.3081	0.4099	0.5125	0.6176	0.7282	0.8485	0.9822	1.1199
$\alpha=0.6$	0	0.1040	0.2070	0.3093	0.4115	0.5145	0.6199	0.7310	0.8521	0.9875	1.1302
$\alpha=0.7$	0	0.1043	0.2076	0.3104	0.4129	0.5162	0.6221	0.7336	0.8555	0.9923	1.1395
$\alpha=0.8$	0	0.1045	0.2081	0.3112	0.4141	0.5178	0.6240	0.7360	0.8585	0.9967	1.1479
$\alpha=0.9$	0	0.1046	0.2085	0.3119	0.4152	0.5192	0.6258	0.7382	0.8613	1.0007	1.1555
$\alpha=1$	0	0.1047	0.2089	0.3125	0.4161	0.5204	0.6274	0.7402	0.8638	1.0044	1.1624



**Şekil 2.10** Doğrusal olmayan zamanla değişen sistemin bileşen sayısının artırılmasıyla elde edilen sonuçların karşılaştırılması



**Şekil 2.11** Doğrusal olmayan zamanla değişen sistemin bileşen sayısının artırılmasıyla elde edilen hata fonksiyonunun değişimi

Kesir dereceli durum uzay modellerinin analitik olarak çözümü, pratik kontrol problemlerinin çözümü için son derece önemlidir. Bu tarz yöntemlerle çözüm üretilmesi, sistem cevabının optimize edilmesiyle daha iyi kontrol cevapları üretilebilir. Önceki bölümde HPM metodu kullanılarak doğrusal olmayan zamanla değişmeyen ve değişen KD sistemler için çözümler önerilmiş ve sonuçlarda adil bir şekilde karşılaştırmalar yapılarak sunulmuştur.

Örneklerden de anlaşıldığı üzere analitik çözüm yapılırken çok basit bir denklemin çözümünde bile çok fazla matematiksel işlem yapmak gerekir. Bunun yanı sıra güç serisindeki terim sayısının artırılmasıyla sonuç istenen cevaba daha fazla yakınsayabilmektedir. Fakat terim sayısını artırmak matematiksel işlemleri exponansiyel olarak artırmaktadır. Bunun için çözümler yapılırken kullanılan denklemler arasında bir trendin olduğu gözlemlenmiştir. Belki bu trend MATLAB ortamında kodlanarak parametrik olarak çok daha yüksek derecedeki güç serilerine açılım sağlanabilir.

Bu açılımla birlikte istenilen denklem çok yüksek bir doğrulukla sonuçlar üretebilir. Bu bölümde analitik yöntemler ile optimal bir problemin çözümü anlatılmış ve HPM metodu üzerinden durumlar irdelenmiştir.

### 3. MATERYAL VE YÖNTEM

#### 3.1. Tezde Kullanılan Materyal

Bu bölümde tez çalışmasında kullanılacak materyaller verilmektedir.

##### 3.1.1 Çok Girişli Çok Çıkışlı Helikopter Modeli

Çok girişli çok çıkışlı helikopter prototipi (Twin Rotor MIMO System-TRMS) Feedback firması tarafında üretilmiş prototip helikopter deney setidir. Çift motorlu çok girişli çok çıkışlı her iki ucunda birer doğru akım motoru ile çalışan pervaneleri bulunan ve helikoptere benzeyen aerodinamik bir sistemdir. Bağlantı noktaları denge kolunun uç noktalarında küresel bir hareket yapacak şekilde tasarlanmıştır. TRMS helikopter modelinde 2 adet motor bulunmaktadır. Bunlardan biri dikey seviye hareketi sağlayan asıl motordur. Diğeri ise yatay seviye hareketini sağlayan kuyruk motorudur [66].

##### 3.1.2 MATLAB ve Simulink

Bu tez çalışmasında sezgisel algoritmaların geliştirme ortamı olarak MATLAB 2014b sürümü kullanılmaktadır. MATLAB, yüksek seviye bir teknik bilgisayar programlama dilidir ve algoritma geliştirme, bilgi görselleştirme, bilgi analizi ve nümerik hesaplamalar için interaktif bir ortam sunmaktadır. Programın genel özellikleri aşağıda verilmiştir.

- Teknik hesap problemlerini, C, C++ ve Fortran gibi geleneksel programlama dillerinde olduğundan daha hızlı çözebilme,
- Sinyal ve görüntü işleme, haberleşme, kontrol tasarımı, deneme ve ölçme, finansal modelleme ve analizi de içeren geniş bir uygulama alanı sağlama,
- Ek araçlar (programdan bağımsız olarak bulunan özel amaçlı MATLAB fonksiyonları) ile MATLAB programının kullanım alanını, özel problemleri çözebilecek şekilde genişletme,
- Teknik programlama için yüksek seviye dil ortamı sağlama,
- Kod, dosya ve bilgi düzenleme için geliştirme ortamı içermeye,

- Tekrarlayan arařtırmalar, tasarım ve problem çözüme için interaktif bir araç sunma,
- Lineer cebir, istatistik, Fourier analizi, filtreleme, optimizasyon ve nümerik integrasyon için matematiksel fonksiyonlar içermeye,
- Bilgi görselleřtirme için 2-D ve 3-D grafik fonksiyonları içermeye,
- Özel görsel ara yüzler geliřtirmek için araçlar sunma,
- MATLAB tabanlı algoritmaların C, C++, Fortran, Java, COM ve Microsoft Excel gibi dış uygulamalar ve diller ile entegrasyonu için fonksiyonlar içermeye,

Bu tez çalışmasında sezgisel algoritmalarla uygulama yapmak ve geliřtirmek için MATLAB editöründen faydalanılmaktadır.

Bilindiđi üzere “editör” de tüm programlama dillerinde olduđu gibi istenilen işlemleri yapabilen kodlar yazılabilmektedir.

MATLAB Simulink yapısı ile de sistemlerin sürekli zamandaki modelleri oluşturulabilir. Bu kodlar ile istenilen sistemler yazılan kodda bulunan komutlar arayıcılıđıyla yönlendirilebilir. Editörde oluşturulan kod ve sistem için oluşturulan simulink dosyası birleřtirilerek eş zamanlı çalışabilecek hale getirilmektedir.

Bu yapılar yardımıyla sezgisel algoritmalar uygulanarak sistemler için denetçi parametreleri belirlenebilmektedir.

### **3.2 Tezde Kullanılan Yöntem**

Bu tez çalışmasının tamamlanmasında izlenecek yöntem aşağıda verilmektedir.

- Literatürdeki sezgisel algoritmalar incelenip denetçi parametrelerinin belirlenmesinde etkin olan algoritmalar belirlenmiştir.
- Seçilecek algoritmaların gerçek zamanlı sistemlerde çalışabilirliđi test edilmiştir.
- Uçuş kontrol sistemlerinde kullanılabilir sezgisel algoritmalar tespit edilerek veya bu yapılarda çalışabileceđi test edilmiştir.
- Literatürde sezgisel algoritmalarla tam sayı dereceli denetçi parametrelerinin belirleyebilecek yapıya dönüřtürülebilirliđi arařtırılmıştır.
- Denetçi parametrelerinin belirlenmesinin yanı sıra seçilecek ve oluşturulacak algoritmaların performansları da objektif olarak denetçi tasarımı üzerinden irdelenmektedir. Böylece sezgisel algoritmaların performans analizleri yapılabilmektedir.

#### 4. RASSAL PARAMETRE VEKTÖR OPTİMİZASYON YÖNTEMİ

Kesir dereceli matematik ilk defa 1690'da L'Hospital ile Leibniz arasında geçen tartışmada ortaya çıkmıştır. Bu tartışma türev derecesinin tam sayı olmama durumunda türevin nasıl davranacağı sorusuna cevap aranmasıyla başlanmıştır. Tabii o dönemlerde bunun bir paradoksa sebebiyet vereceği düşünülürken sonraki yıllarda bunun bir paradoks olmayacağı muhtemel çözümlerinin olduğu bilim adamları tarafından gösterilmiştir [82, 91, 92]. 19. yüzyılda bu konuyla ilgili sistematik çalışmalar Liouville (1832), Holmagren (1864), Rieman (1953) tarafından başlatılmıştır [41-43].

Aslında KD matematik teorik olarak çok faydalı durmasına rağmen pratik gerçekleştirme noktasında birçok zorluğu bulunmaktadır.

Özellikle son yıllarda KD sistemlerin çözümü ve pratik gerçekleştirilmesine dönük çalışmaların artmasıyla birçok bilim dalında özellikle mühendislik alanında geniş bir uygulama alanı bulunmaktadır [93, 94].

Yapılan uygulamalarda tatmin edici sonuçların alınması araştırmacıların bu yöndeki çalışmalarına devam etmesini teşvik etmektedir. Son dönemdeki çalışmalar genellikle tam sayı dereceli sistemler yerine KD sistemlerin kullanılmasının nasıl bir etki doğuracağı yönündedir [94]. Birçok çalışmada KD sistemler için basit, pratik, dayanıklı ve gerçek zamanlı olarak tatmin edici sonuçlar üretebilen sistem tasarımına ve optimizasyonuna yönelik çalışmalar yapılmaktadır [95,96]. Bu çalışmaların genel amacı optimum sistem performansının elde etmek olarak gösterilir.

Bilindiği gibi optimizasyon yöntemleri ile gerçek zamanlı sistemler için etkili denetçiler tasarlanarak etkin ve dayanıklı kontrol performansları elde edilebilir. Optimum kontrol yapısının öneminden dolayı birçok çalışmada farklı optimizasyon yöntemleri kullanılarak birçok farklı kontrol problemine çözüm üretilmektedir [4]. Örneğin, su altı araç manipulator sistemlerinin kontrolü için sinir-bulanık (neuro-fuzzy) yapısının kullanılması çalışması [97]'de önerilmiştir. PI denetçi kat sayılarının Dipohantine denklemlerinin kullanılarak elde edilmesi [98]'de sunulmuştur. Optimal bulanık denetçi stratejisi temel alınarak genetik algoritmanın gerçek zamanlı olarak kodlandığı ve bu kod kullanılarak DSP üzerinden fırçasız motorun kontrol edildiği çalışma [99]'da gösterilmiştir.

Literatürde son dönemdeki çalışmalara baktığımızda optimizasyon algoritmaları kullanılarak tam sayı dereceli denetçi tasarımı giderek yaygınlaşmaktadır [100].

Birçok optimizasyon yönteminde kullanılan algoritma optimize edilen sistemin matematiksel modeline ihtiyaç duyar [47, 101]. Fakat gerçek zamanlı sistemlerin matematiksel modellerinin tamamıyla bilinmesi neredeyse imkansızdır. Çünkü matematiksel modeli bilinse dahi sistem çevresel faktörlerden dolayı birçok bozucu tarafından etkilenir. Bundan dolayı kullanılan optimizasyon algoritması sistem modelinden bağımsız bir şekilde çalışabilecek yetenekte olması gerekir. Ancak bu şekilde nümerik optimizasyon yöntemlerinin kullanım amacı ortaya çıkar.

Bu tez çalışmasında bu tarz bir optimizasyon algoritması oluşturmaya dönük çalışmalar gerçekleştirilmiştir.

Tezin bu bölümünde gerçek zamanlı sistemler için modelden bağımlı ve bağımsız olarak denetçi tasarımı yapabilen Rassal Parametre Vektör Optimizasyon (RPVO) yöntemi önerilmiştir [82]. Bu yöntem ile sistemlere KD denetçi ( $PI^{\lambda}D^{\mu}$ ), ve adaptif KD denetçi ( $A-PI^{\lambda}D^{\mu}$ ) tasarımı yapılabilir. Bu yöntem temelde parametreleri sisteme yükleme ve daha sonra deneme felsefesine göre çalışır. Yapılan hareketler parametre vektör uzayında rassal olarak ilerler. İlgili koşulların sağlanmasıyla parametreler güncellenir. Önerilen algoritmanın performansı simülasyon ve gerçek zamanlı sistemler üzerinde test edilmiştir. Uygulama platformu olarak çift motorlu çok girişli çok çıkışlı helikopter prototipi TRMS kullanılmıştır [66]. TRMS uçuş kontrol uygulamalarının yapılabileceği MATLAB ile uyumlu çalışabilen Feedback firması tarafından üretilmiş olan bir deney setidir. Bu deney setiyle, uçuş kontrol uygulamalarında karşılaşılan birçok problemi deney seti üzerinde gerçekleştirme imkanı vardır [82].

Aslında uçuş kontrol problemleri denetçiler açısından son derece güç kontrol problemleri olarak adlandırılır. Çünkü sistemler birçok doğrusal olmayan bileşeni bünyesinde barındırır. Bunun yanı sıra sistemde modellenemeyen birçok bileşen de bulunur [96, 102]. Sistemin tam modeli çıkarılsa bile sistem görev esnasında yani havadayken birçok bozucuya maruz kalmaktadır. Bundan dolayı tasarlanacak denetçinin bahsedilen bu parametre değişimlerinin üstesinden gelmesi veya denetçi tasarımı yapılacak yöntem bu problemleri çözebilecek yetenekte olması beklenir. Yani kullanılan optimizasyon algoritması denetçi parametrelerinin sistem çalışırken bile ayarlayabilecek hızda olması gerekir.

Aslında bu tez çalışmasında RPVO yöntemi önerilerek etkin kontrol performansı sağlayabilmek amaçlanmaktadır.

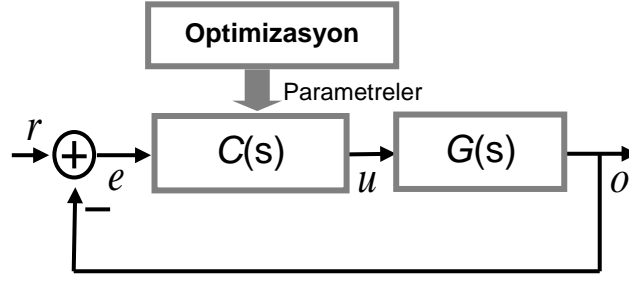
Takip eden bölümlerde önerilen algoritmanın matematiksel alt yapısı TRMS sistemi kullanılarak gerçekleştirilen simülasyon ve gerçek zamanlı uygulamalar detaylarıyla anlatılmaktadır.

#### **4.1. Rassal Parametre Vektör Optimizasyon Algoritmasıyla Denetçi Parametrelerinin Otomatik Olarak Ayarlanması**

##### **4.1.1 Çalışmanın Amacı**

Kesir dereceli PID denetçi  $PI^\lambda D^\mu$  formunda gösterilir [47]. Birçok çalışmada  $PI^\lambda D^\mu$  denetçinin tam sayı dereceli PID denetçilerine karşın daha iyi sonuçlar üretebildiği gösterilmektedir. Denetçi yapısına eklenen  $(\lambda, \mu)$  parametreleriyle parametre vektör uzayı genişletilerek daha iyi kontrol performansını verebilen parametre konfigürasyonlar optimizasyon ile tespit edilir [47, 94]. PID denetçiler denetçinin olduğu her yerde basit ve kolay olmasından dolayı tercih edilir. Kesir dereceli denetçilerin pratikte uygulanabilirliğinin gösterilmesiyle bu yapıların uygulama alanlarında yaygınlaşması sağlanır. Bundan dolayı  $PI^\lambda D^\mu$  denetçinin parametrelerinin optimize edilmesi bilim dünyası açısından son derece güncel bir problem olarak gösterilir.

Şekil 4.1’de temel adaptif kontrol sisteminin şekli verilmiştir. Şekilde  $G(s)$  sistemin matematiksel modelini gösterirken  $C(s)$  ise sistemin denetçisini karakterize eder. Şekilde gösterilen optimizasyon bloğu sistemden aldığı geri beslemeye göre kontrol performansını iyileştirmeye dönük denetçi parametrelerini değiştirir.



**Şekil 4.1** Adaptif birim geri beslemeli kontrol sistemin temel blok diyagramı

Aslında tezin bu bölümünde amacımız gerçek zamanlı bir sistem için Şekil 4.1'deki yapıyı kullanarak sistem üzerinde çalışabilen adapif  $PI^\lambda D^\mu$  denetçi tasarımı yapabilen bir yöntem önermektir. Denetçi parametrelerinin otomatik olarak ayarlanması parametre optimizasyon problemi olarak adlandırılır. İlgili parametreler mevcut maliyet fonksiyonunu minimize veya maximize ederek optimizasyonla ayarlanır. Literatüre bakıldığında bu konuda bazı çalışmalar bulunmaktadır. Örneğin [96]'da çoklu amaç fonksiyonu kullanılarak evrimsel algoritma ile denetçi tasarımı gerçekleştirilmiştir. Fakat bu çalışmada önerilen RPVO yöntemi çoklu denetçi parametrelerini çok fazla matematiksel yöntemle gerek kalmaksızın sistemin kendi modelini kullanarak hızlı ve doğru bir şekilde ayarlayabilir. Elde edilen denetçilerin adaptif kontrol performansı sergiledikleri deneysel testlerle gösterilmiştir.

#### 4.1.2 Problemin Tanımlanması

$PI^\lambda D^\mu$  denetçi parametreleri  $t = nh$  zaman aralığında  $X$  parametre uzayı olmak üzere  $X^n = [k_p^n k_i^n k_d^n \lambda^n \mu^n]$  şeklinde ifade edilir.  $h$  ayrık durumlu sistemin örnekleme periyodudur.  $n = \{1,2,3,\dots\}$  ise ayrık zamanlı sistemin zaman artışıdır. Denklem 4.1 verilen şartın sağlanmasıyla  $X^n$  denetçi parametre vektörü zaman içerisinde  $X^m$  vektörüne dönüşmektedir.

$$E(X^m) - E(X^n) < 0 \quad (4.1)$$

Denklemdaki  $E(\cdot)$  pozitif gerçel tanımlı hata fonksiyonudur. Bu denklem tüm optimizasyon süresince parametrelerin güncellenme kriteri olarak kullanılır. Bu kriterin sağlanmasıyla ilgili parametre güncellenir.  $p = m - n > 0$   $PI^\lambda D^\mu$  denetçinin adaptasyon periyodunu gösterir. İfadedeki  $n$  ve  $m$  ise sırasıyla şu anki ve sonraki örneklemleri gösterir. Adaptasyon süresi boyunca  $PI^\lambda D^\mu$  denetçi parametreleri hatayı minimize etmek için sürekli güncellenir. Adaptasyon oranı  $\Delta E^{m,n} / p$  ifadesiyle hesaplanır.  $\Delta E^{m,n}$  ifadesi parametre vektör hareketinin uygunluğunun bir ölçüsüdür. Aynı zamanda  $n$  örnekten  $m$  örneğe geçerken  $X$  parametre vektör uzayının değişim derecesini ifade eder. Bu ifadede daha önce bahsedildiği üzere  $\Delta E^{m,n} = E(X^m) - E(X^n)$  eşitliği ile hesaplanır.

Eğer  $\Delta X^{m,n}$  ifadesi Denklem (4.1) deki şartı sağlıyorsa  $nh$  örnekleme zamanında bu hareket uygun bir hareket olarak adlandırılır. Eğer bu şartı sağlamıyorsa, bu hareket  $PI^\lambda D^\mu$  denetçi parametrelerinin optimize edildiği süreç içerisinde uygun bir hareket değildir diye adlandırılır.

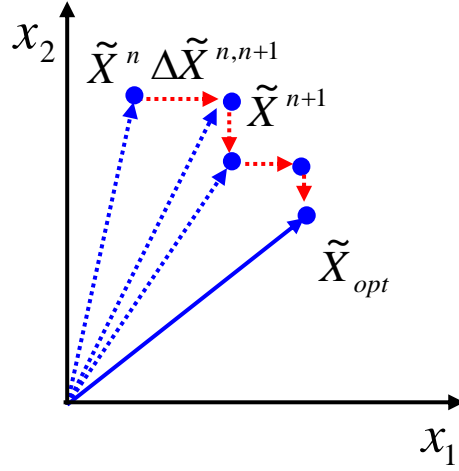
#### 4.1.3 Rassal Parametre Vektör Optimizasyon Algoritması

RPVO yöntemi  $X$  parametre vektör uzayında rassal hareketler yaparak hatayı minimize etmeye çalışır. Algoritma amaç fonksiyonu başlangıç değerlerini içerisinde bulunduran  $X^0$  başlangıç parametre vektörünü optimizasyon süresince optimal değerleri içerisinde bulunduran  $X_{opt}$  vektörüne dönüştürür. Optimal değerlerin aranması, önceden tanımlanan parametre değerleri arasında yapılır. Yapılan işlemin matematiksel formu Denklem 4.2’de verilmektedir [82].

$$E(X_{opt}) = \min_{X^n \in X_s} \{E(X^n)\} \quad (4.2)$$

Yapılan işlemi daha anlaşılır hale getirmek amacıyla iki değer için optimizasyon süreci aşağıda verilmiştir.  $\tilde{X} = [x_1, x_2]$  bir parametre vektörüdür ve optimize edilmesi gereken iki değere sahiptir. Şekil 4.2’de iki yönlü olarak optimizasyon sürecinin nasıl gerçekleştiğini gösterir. İki elemanlı başlatılan parametre vektörü

$\tilde{X}^n$  hatayı minimize ederek  $\tilde{X}_{opt}$  parametre vektörüne dönüşür. Bu süreçte gerekli rassal hareketlerde  $\Delta\tilde{X}^{n+1,n}$  değişim ifadesine göre gerçekleştirilir.

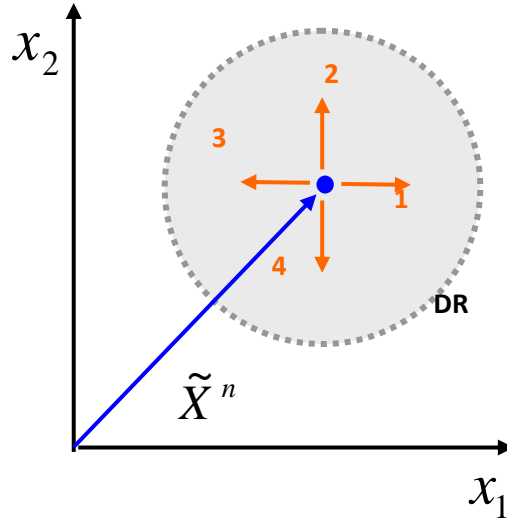


**Şekil 4.2**  $\tilde{X} = [x_1 \ x_2]$  iki bileşenli parametre vektöründen  $\tilde{X}_{opt}$  optimum sonucun elde edilmesi [82]

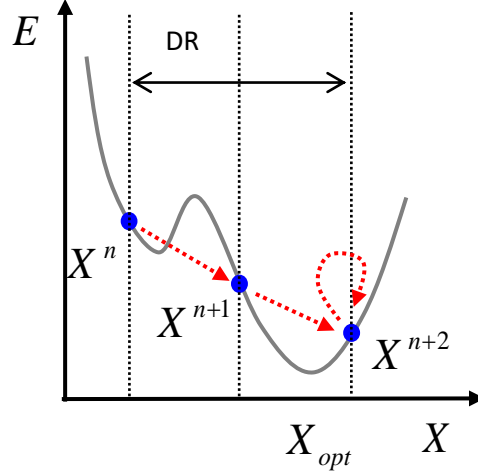
Şekil 4.3’de ise  $\tilde{X}^n$  vektörünün değişim bölgesi (divergence region (DR)) çizgili daire ile gösterilmektedir.  $\tilde{X}^n$  vektörünün değişim bölgesi parametre vektör uzayında  $\Delta\tilde{X}^{n+1,n}$  değişim vektörüyle sınırlandırılmıştır. Değişim bölgesinin sınırlandırılması  $\tilde{X}_{opt}$  vektörüne daha yavaş yakınsamayı sağlar. Değişim bölgesi büyütülerek belki daha hızlı adaptasyon sağlanabilir ama bu durum optimizasyon sürecinin doğruluk oranı düşürebilir. Değişim adımı  $\Delta\tilde{X}^{n+1,n}$  in değişimi  $\tilde{X}_{opt}$  sonucunun doğruluğunu geliştirmeye dönük olmalıdır.

Şekil 4.4 ve Şekil 4.5’de parametrenin sabit ve değişken adımlarla parametre vektör uzayındaki değişimleri gösterilir. Sabit adım miktarı olan  $\Delta\tilde{X}^{n+1,n}$  ile  $\tilde{X}_{opt}$  değeri ilgili iterasyonlar arasında sıkışmış durumdadır. Adım aralığının sabit olmasından dolayı optimum noktayı bulma ihtimali bulunmamaktadır. Ancak adım aralığının düşürülmesiyle mümkün olmaktadır.

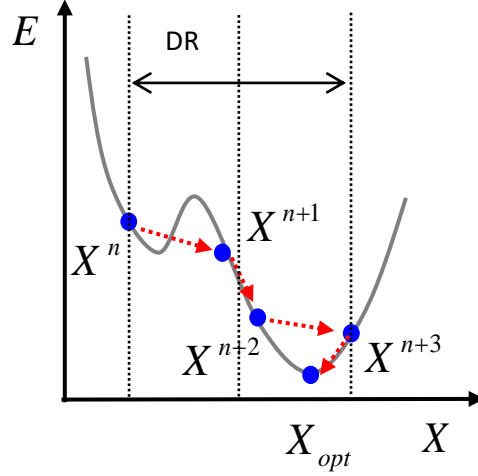
Fakat adım aralığının değişken olması durumunda ise Şekil 4.5’de gösterildiği gibi optimizasyon süresince optimal değeri bulabilmektedir. RPVO yönteminde değişken ve rassal adım aralığı kullanmasından dolayı algoritma her zaman ilgili optimal noktayı bulabilir.



**Şekil 4.3**  $x_1$  ve  $x_2$  değişkenlerinin ileri ve geri yönlü hareketleri. (Durum 1 ve 2, ileri yönlü hareketleri, durum 3 ve 4 ise geri yönlü hareketleri göstermektedir.)



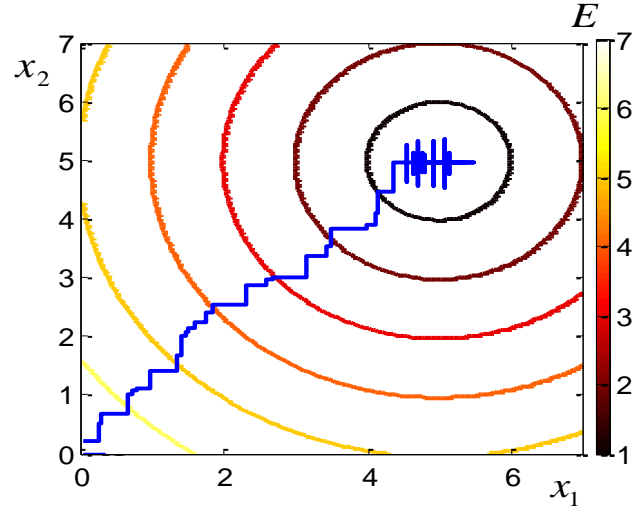
**Şekil 4.4**  $X_{opt}$  değerine sabit adım aralığı ile yakınsama



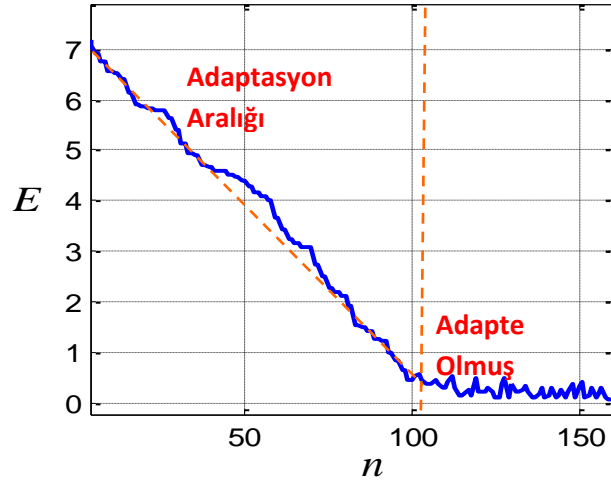
Şekil 4.5  $X_{opt}$  değerine  $\Delta X^{n+1,n}$  değişken adım aralığı ile yakınsama

RPVO yöntemi parametrelerin sisteme uygulanması ve denenmesi (set and trial (ST)) olarak adlandırılan iki yönlü yapıya göre çalışır. Algoritma Denklem 4.1'deki şartı sağlama prensibine göre uygun parametre değişimini bulmayı amaçlar.  $X$  parametre vektörü algoritmanın başlangıcında rassal olarak ileri yönlü hareket eder. Eğer ileri yönlü hareket  $\Delta E^{n+1,n} \geq 0$  şartı sağlamıyorsa algoritma geri yönlü rassal bir hareket yapar. Eğer geri yönlü hareket de ilgili şart sağlanmıyorsa bu hareket algoritma için yanlış bir yön anlamına gelir ve algoritma başlangıç değerlerini kullanarak aramaya devam eder. Şekil 4.3'de iki değişkenli parametre vektör uzayı için olası ileri yönlü ve geri yönlü hareketler gösterilmektedir.

Şekil 4.6'da iki bileşenli parametre vektörü için RPVO yönteminin global optimum değere ulaşırken hataya göre parametre değişimleri gösterilmiştir. Şekil 4.7'de ise hata fonksiyonun global minimuma ulaşırken değişimi sunulmuştur.



Şekil 4.6  $\tilde{X}^n = [x_1^n \ x_2^n]$  değişkelerinin optimizasyon süresince konveks hata fonksiyonuna göre değişimi



Şekil 4.7 Hata fonksiyonun iterasyonlar süresince global minimuma yakınsaması

#### 4.1.4 Rassal Parametre Vektör Optimizasyonu ile Adaptif Kesir Dereceli Denetçi Tasarımı

RPVO algoritması optimize edilecek parametre sayısının iki katı kadar ileri ve geri yönlü hareket yapar. Örneğin optimize edilecek parametre sayısı  $k$  ise ilgili iterasyonda  $2k$  kadar test yapılır.  $PI^\lambda D^\mu$  KD denetçi parametrelerinin

optimizasyon süresince parametre vektörümüz 5 elemanlıdır ve testler bu yapıya göre yapılır [82].

Adaptif  $PI^\lambda D^\mu$  denetçinin değişim vektörü aşağıda gibidir.

$$\Delta X^n = \phi_v^n [\Delta k_p^n \Delta k_i^n \Delta k_d^n \Delta \lambda^n \Delta \mu^n]. \quad (4.3)$$

İleri yönlü hareket aşağıdaki gibi hesaplanır.

$$X^{n+1} = X^n + \phi_v^n [\Delta k_p^n \Delta k_i^n \Delta k_d^n \Delta \lambda^n \Delta \mu^n] \quad (4.4)$$

İlgili parametreler uygun bir değişim oranıyla ileri yönlü hareket ettirilir ve ilgili harekete ilişkin hata değeri Denklem 4.5’de verildiği gibi elde edilir.

$$E(X^n + \Delta X^n) - E(X^n) < 0 \quad (4.5)$$

Denklemdaki  $\phi_v$  ise değişim oran vektörünü gösterir. Benzer şekilde geri yönlü hareket aşağıdaki gibidir.

$$X^{n+1} = X^n - \phi_v^n [\Delta k_p^n \Delta k_i^n \Delta k_d^n \Delta \lambda^n \Delta \mu^n] \quad (4.6)$$

İlgili parametreler uygun bir değişim oranıyla geri yönlü hareket ettirilir ve ilgili harekete ilişkin hata değeri Denklem 4.7’de verildiği gibi elde edilir.

$$E(X^n - \Delta X^n) - E(X^n) < 0 \quad (4.7)$$

$PI^\lambda D^\mu$  kesir dereceli denetçinin RPVO yöntemiyle tasarımında  $\phi_v$  vektörü  $v = \{1,2,3,4,5\}$  için  $\{[\eta 0000]^T, [0\eta 000]^T, [00\eta 00]^T, [000\eta 0]^T, [0000\eta]^T\}$  değerlerini alır. Matrisde bulunan  $\eta$  değeri her adımda  $[0,1]$  aralığında elde edilen rassal bir değerdir. Alt indeks  $v$  ise parametre vektörü olan  $X$ ’in hangi elemanının ileri veya geri yönlü hareket yaptığını gösterir. Örneğin  $\phi_1$  ifadesi

$[\eta 0000]^T$  vektörünü ifade eder. Bu durum  $PI^\lambda D^\mu$  denetçinin ilk elemanı olan  $k_p^n$ 'nin ileri ve geri yönlü hareketini temsil eder.

Şekil 4.8'de adaptif  $PI^\lambda D^\mu$  denetçi parametrelerinin RPVO algoritması ile yapılan optimizasyon sürecinin akış grafiği gösterilmektedir.  $PI^\lambda D^\mu$  denetçiyi pratik olarak gerçekleştirmek için sürekli kesin genişleme metodu (Continuous Fraction Expansion-CFE) kullanılmıştır. Bu metoda göre kesir dereceli türev ve integralin 4. dereceden tam sayı dereceli eşdeğer modelleri alınarak sistemler üzerinde kullanılır. Bu yöntem kesir dereceli integro diferansiyel ifadelerin pratik olarak gerçekleşmesinde sıklıkla tercih edilir. Denklem 4.8'de bu yaklaşımın matematiksel formudur [103-105].

$$(1+x)^\alpha = \frac{1}{1-} \frac{\alpha x}{1+} \frac{(1+\alpha)x}{2+} \frac{(1-\alpha)x}{3+} \frac{(2+\alpha)x}{2+} \frac{(2-\alpha)x}{5+\dots} \quad (4.8)$$

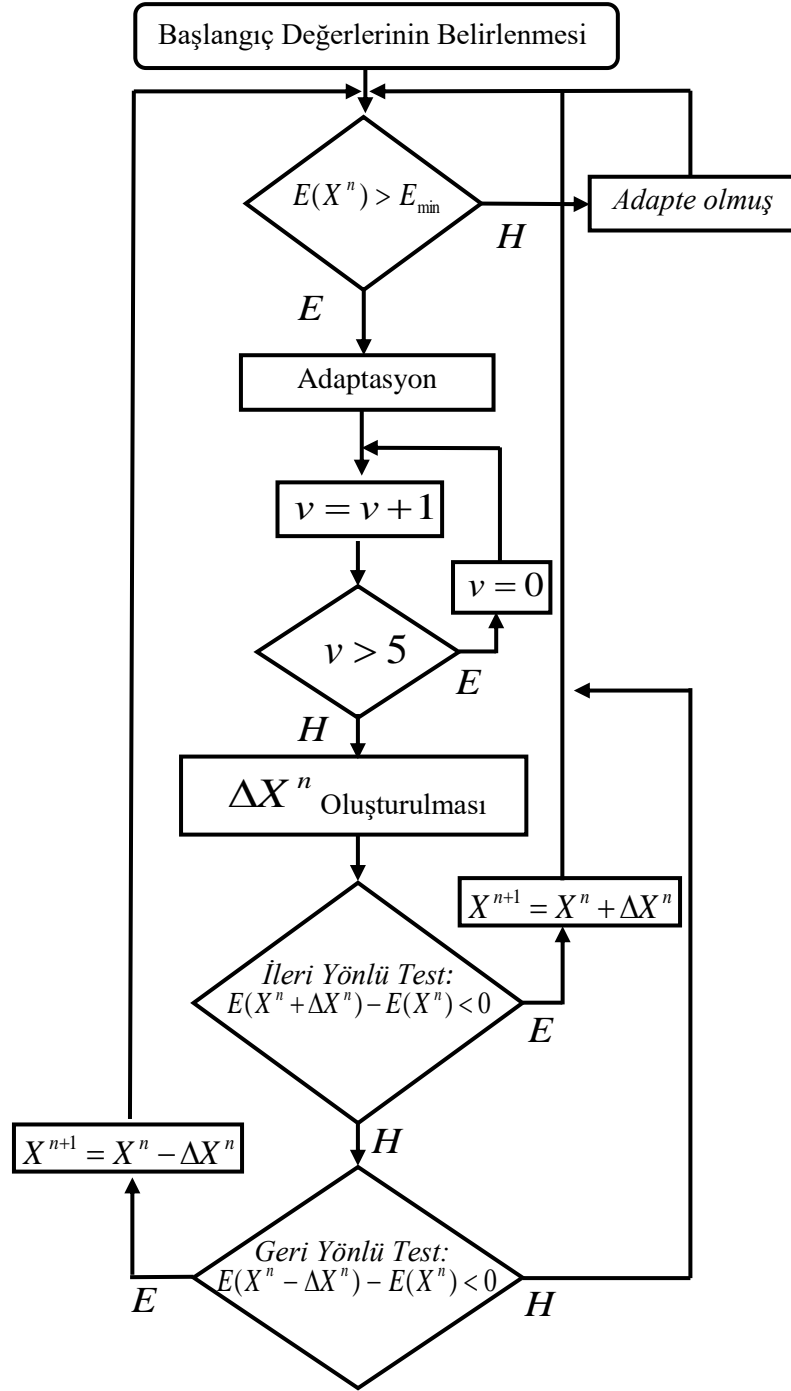
Şekil 4.8'de verilmiş olan RPVO algoritması gerçek zamanlı olarak KD denetçi parametrelerini optimize eder. Algoritma iki aşamadan oluşur. Bunlar adaptasyon modu ve adapte olmuş mod olarak adlandırılır. Sistem adaptasyon süresince parametre vektör uzayını tarar. Önceden belirlenen hata değerinin altına düştüğü andan itibaren parametreleri daha küçük adımlarla daha hassas ayarlamaya çalışır. Böylelikle çok daha iyi sonuçlar elde edilebilir.

RPVO algoritmasıyla A- $PI^\lambda D^\mu$  denetçinin parametrelerinin optimize edilme süresince Denklem 4.9'da verilen eşitlik kullanılır. Denklemde bulunan  $L$  (Batch error) hata yığını olarak adlandırılır.

$$E = \frac{1}{L} \int_{t-L}^t e^2(t) dt \quad (4.9)$$

Denklem 4.9 verilen hata değeri RPVO algoritmasıyla Şekil 4.1'de gösterildiği gibi minimize etme doğrultusunda çalışır. RPVO algoritması tüm bileşenleri ayrı ayrı ileri veya geri yönlü hareket ettirir ve her hareketin sonucunda Denklem 4.9'da verilen hata değerini hesaplar.

Hesaplanan hata deęeri bir önceki harekette hesaplanan hata deęerinden küçükse hareketin doğru yönde olduğunu gösterir. Bu hareket ileri ve geri yönlü olabilir. Bu deęeri belleğinde tutarak bir sonraki bileşenin ileri ve geri yönlü hareketini gerçekleştirir. Her iki hareketinde hata deęerini düşüren bir etkisi söz konusu değilse hareket orjinal halinde bırakılır ve diğer bileşenlerin testine devam edilir. RPVO yönteminin en önemli avantajı sistemin modeline ihtiyaç duymaksızın optimizasyon sürecini devam ettirebilmesidir. Fakat bunun için yeterli miktarda optimizasyon döngüsüne yani iterasyon sayısına ihtiyaç duyar. Bunun yanı sıra uygun başlangıç koşullarının verilmesi algoritmanın optimum sonuca hızlı bir şekilde ulaşması anlamına gelir. Bu sayede tüm iterasyonları ilgili deęeri iyileştirmeye dönük kullanılabilir. Aksi durumda birçok iterasyonu uygun parametre vektör uzayını tespit etmek için tüketir. Aslında başlangıç deęer problemi birçok algoritmada olan bir problemdir. Tezin diğer bölümlerinde bu problemin çözümüne ilişkin öneriler bulunmaktadır.



Şekil 4.8 KDPID denetçi parametrelerinin optimizasyonu için RVPO algoritmasının akış şeması

#### 4.1.5 Adaptif Kesir Dereceli Denetçi Parametrelerinin Matematiksel Bir Model İçin Optimizasyon ile Türetilmesi

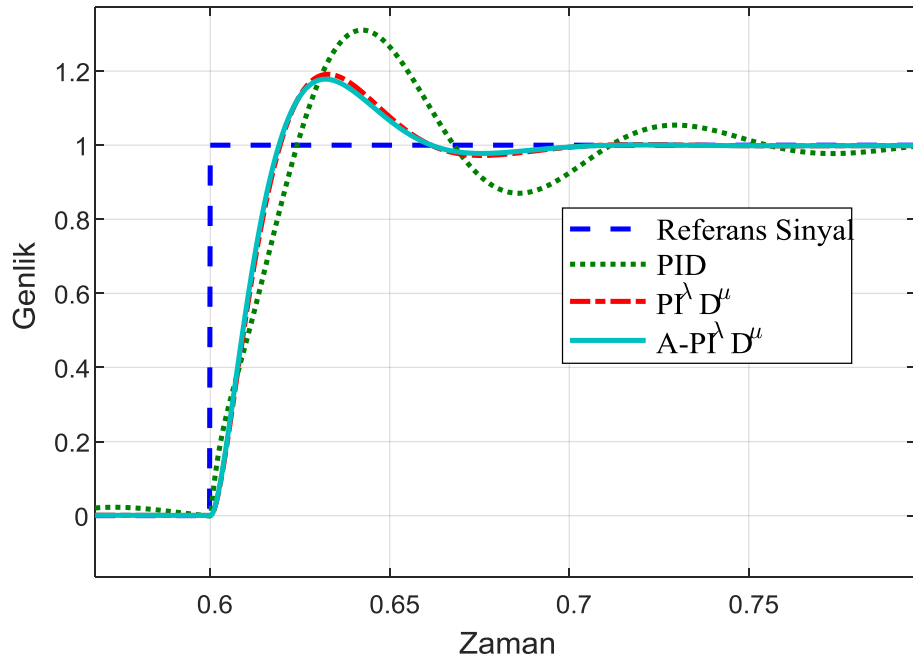
Bu bölümde A- $PI^\lambda D^\mu$  denetçi cevabı klasik tam sayı dereceli denetçi cevabı ve adaptif olmayan  $PI^\lambda D^\mu$  denetçi cevapları ile karşılaştırılmaktadır. Denklem 4.10'da verilen 2 kutuplu transfer fonksiyonu ve denetçi konfigürasyonları MATLAB Simulink ortamında gerçekleştirilerek optimizasyon algoritmasında kullanılmıştır.

$$G(s) = \frac{K}{a_2 s^2 + a_1 s + a_0} \quad (4.10)$$

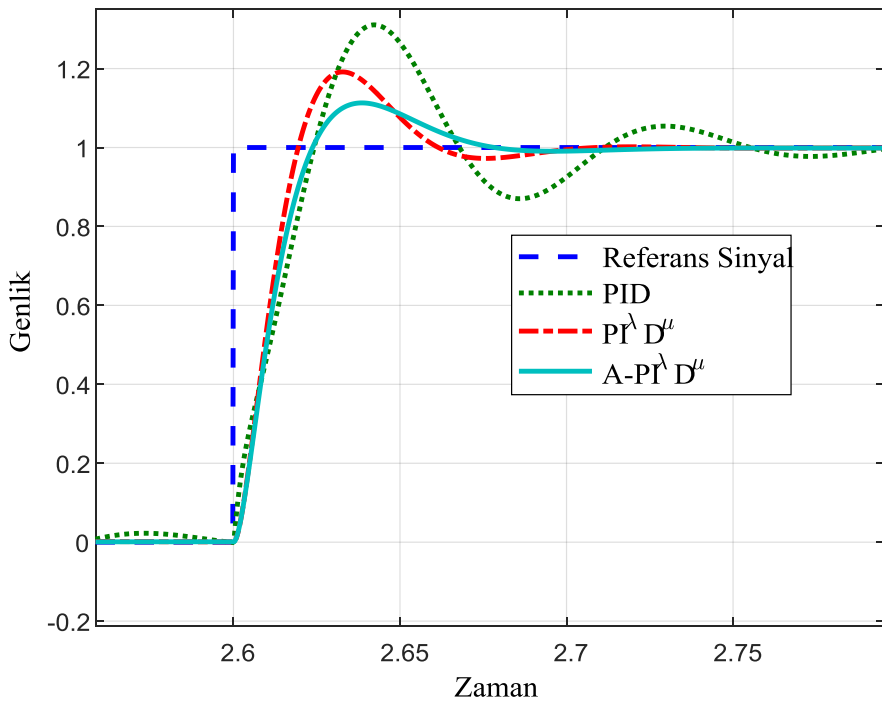
Referans sinyal olarak periyodik kare dalga kullanılmıştır. G(s)'in sabit katsayılı durumu ve katsayıların dinamik olarak değiştirildiği durum için sonuçlar irdelenmiştir.

Kesir dereceli denetçileri pratik olarak gerçekleştirmek için CFE yöntemi kullanılarak 4. dereceden tam sayı dereceli yaklaşık modeller elde edilmiştir [104-105]. Klasik PID parametreleri ( $k_p = 1$ ,  $k_i = 1$ ,  $k_d = 1$ ) literatürde bilinen bir yöntem ile türetilmiştir [48, 106]. Önerilen RPVO algoritmasıyla adaptif olmayan  $PI^\lambda D^\mu$  denetçi katsayıları ( $k_p = 1$ ,  $k_i = 2$ ,  $k_d = 1$ ,  $\lambda = 0.95$ ,  $\mu = 0.3$ ) bulunmuştur. Daha sonra aynı model için RPVO algoritmasıyla adaptif  $PI^\lambda D^\mu$  denetçi katsayıları ( $k_p = 1$ ,  $k_i = 1.5$ ,  $k_d = 7.85$ ,  $\lambda = 0.95$ ,  $\mu = 0.3$ ) elde edilmiştir. Bu algoritma ayrıca tam sayı dereceli denetçi optimizasyonunda da kullanılmıştır [106, 107].

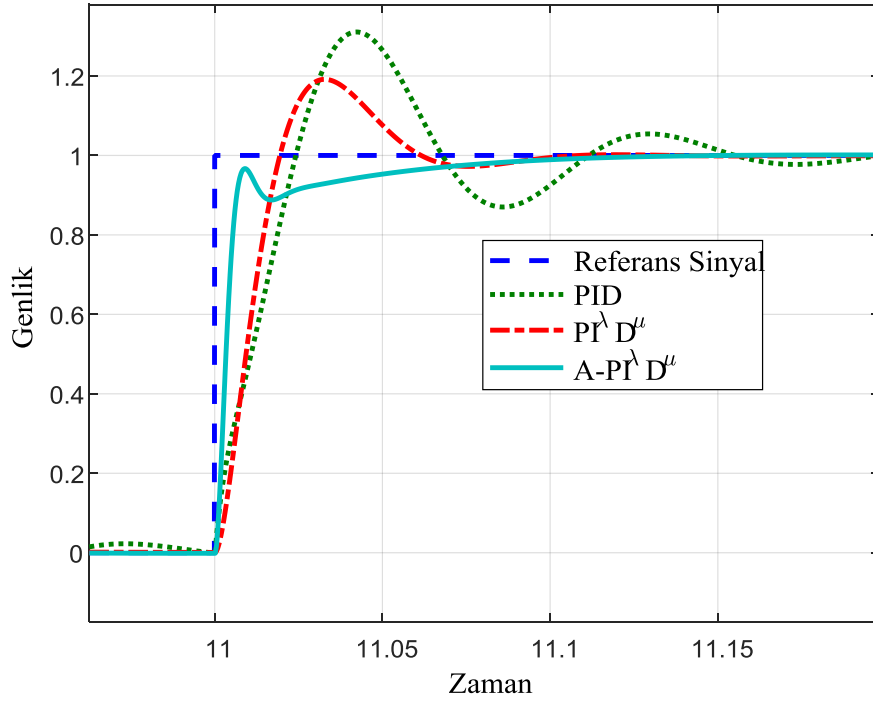
Şekil 4.9'da önerilen A- $PI^\lambda D^\mu$  denetçi cevabı adaptif olmayan  $PI^\lambda D^\mu$  denetçi cevapları ve klasik tam sayı dereceli PID cevapları karşılaştırılmıştır. Şekilden de görüldüğü gibi A- $PI^\lambda D^\mu$  denetçi yapısıyla adaptif olmayan  $PI^\lambda D^\mu$  denetçi yapısı benzer sonuçlar üretebilirken bu cevapların klasik PID denetçiye karşı üstünlüğü vardır. 2,6. saniye sonundaki birim basamak cevapları Şekil 4.10'da karşılaştırılmıştır. Şekilden de anlaşılacağı üzere A- $PI^\lambda D^\mu$  denetçi cevabının çok daha iyi bir oturma zamanına sahip olduğu görülmektedir. Şekil 4.11'de ise 11. saniyeden sonra önerilen denetçinin diğer denetçilere nazaran çok daha iyi oturma ve yükselme zamanlarına sahip olduğu görülmektedir.



Şekil 4.9 Denetçilerin 0.6. saniyedeki birim basamak cevapları



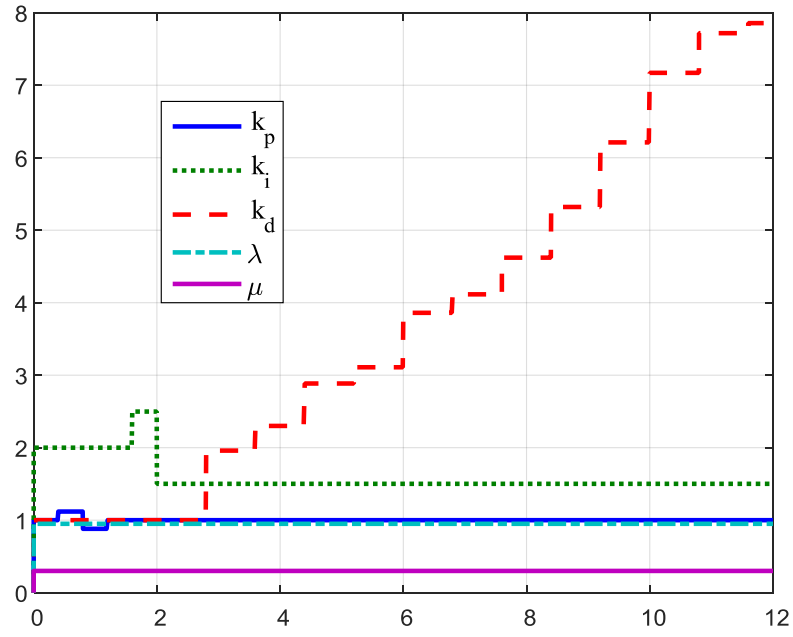
Şekil 4.10 Denetçilerin 2.6. saniyedeki birim basamak cevapları



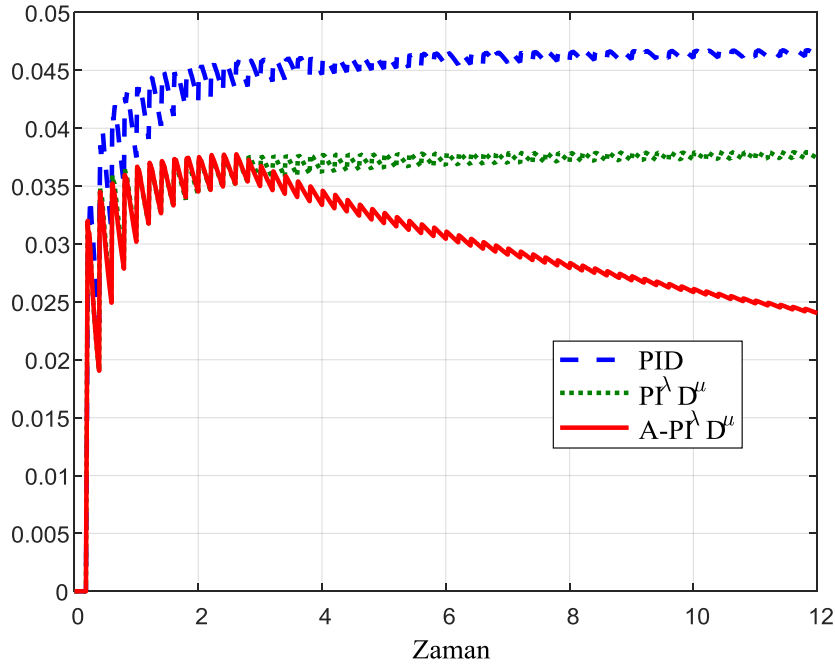
**Şekil 4.11** Denetçilerin 11. saniyedeki birim basamak cevapları

Şekil 4.12’de optimizasyon süresince denetçi parametrelerinin değişimi gösterilmiştir. Şekil 4.13’de simülasyon zamanındaki ortalama hata değerinin değişimi, kullanılan 3 farklı denetçi yapıları için verilmiştir.  $G(s)$  matematiksel modelinin kat sayıları optimizasyon süresince değiştirilerek RPVO algoritmasının adaptif olarak denetçi tasarlayabileceği Şekil 4.14’de verilmiştir.

$G(s)$ , matematiksel modeli  $a_0 = [-1,1]$ ,  $a_1 = [1,3]$  ve  $K = [3,5]$  değerleri arasında değiştirilerek farklı modeller oluşturulmuştur. Daha sonra her konfigürasyon için RPVO algoritmasının adaptif olarak denetçi tasarlayabildiği gösterilmiştir.



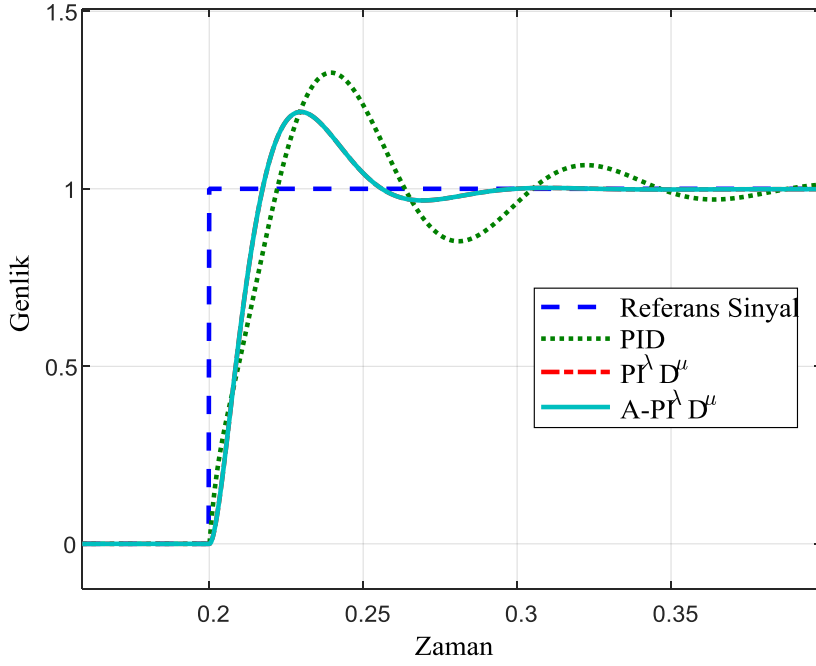
Şekil 4.12 A-KDPID denetçi parametrelerinin 12 saniyelik simülasyon süresince değişimi



Şekil 4.13 Optimizasyon süresince ilgili denetçilere göre hatanın değişimi

Şekil 4.14, Şekil 4.15 ve Şekil 4.16'da iki kutuplu modelin tüm konfigürasyonları için RPVO ile üretilen  $A-PI^\lambda D^\mu$  denetçilerin adaptasyon yetenekleri

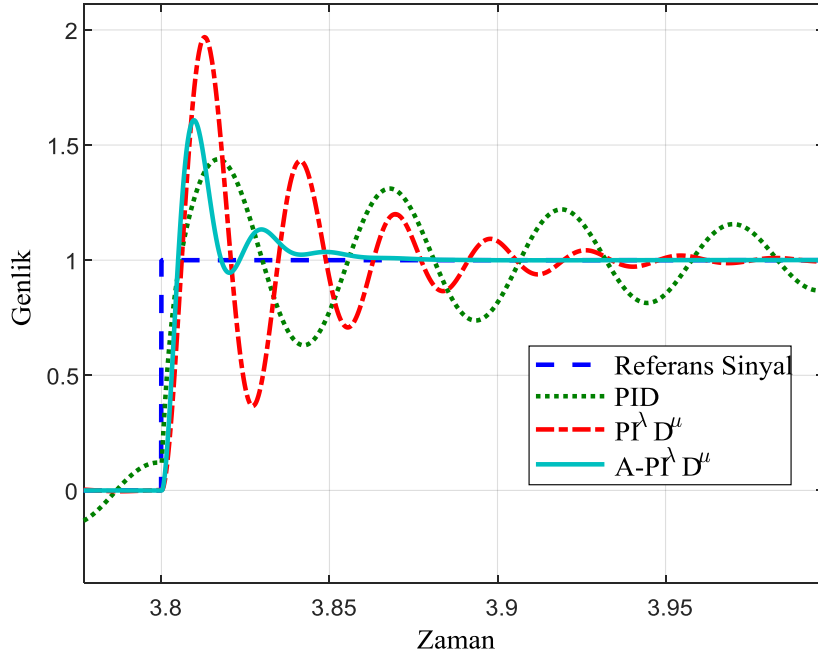
gösterilmiştir. Özellikle  $A-PI^\lambda D^\mu$  denetçinin diğer denetçilere karşı üstünlüğü açık bir şekilde görülmektedir. Modelin kat sayılarının değişmesi durumunda RPVO algoritması önerilen optimizasyon yöntemi ile oturma ve yükselme zamanı gayet iyi olan hızlı tepkiler verebilen denetçiler üretmiştir. Bu da RPVO algoritmasının sistem parametrelerinin değişmesi durumunda ilgili modele rahatlıkla denetçi tasarlayabileceğini göstermiştir. Bu durum algoritmanın adaptif yeteneklere sahip olduğunun en önemli göstergesidir.



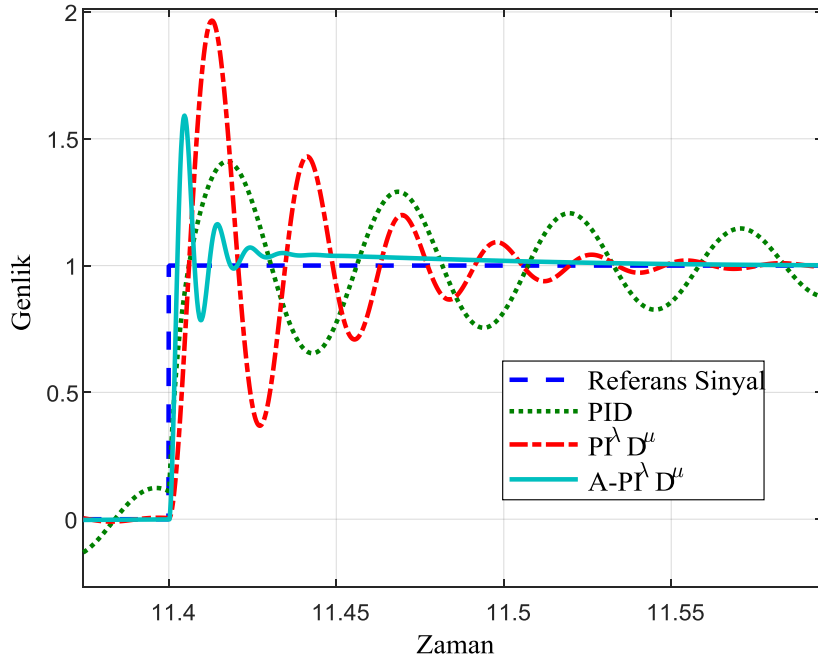
**Şekil 4.14** Denetçilerin 0.2. saniyedeki birim basamak cevapları

Kullanılan yöntemin avantajını göstermek için adaptif bir optimizasyon algoritması ile denetçi tasarlanmış bir modele RPVO algoritması kullanılarak denetçi tasarlanarak sonuçlar karşılaştırmalı olarak verilmiştir. Denklem 4.11 kullanılan zaman gecikmeli modeldir [103].

$$P_1(s) = \frac{e^{-0.2s}}{s-1} \quad (4.11)$$



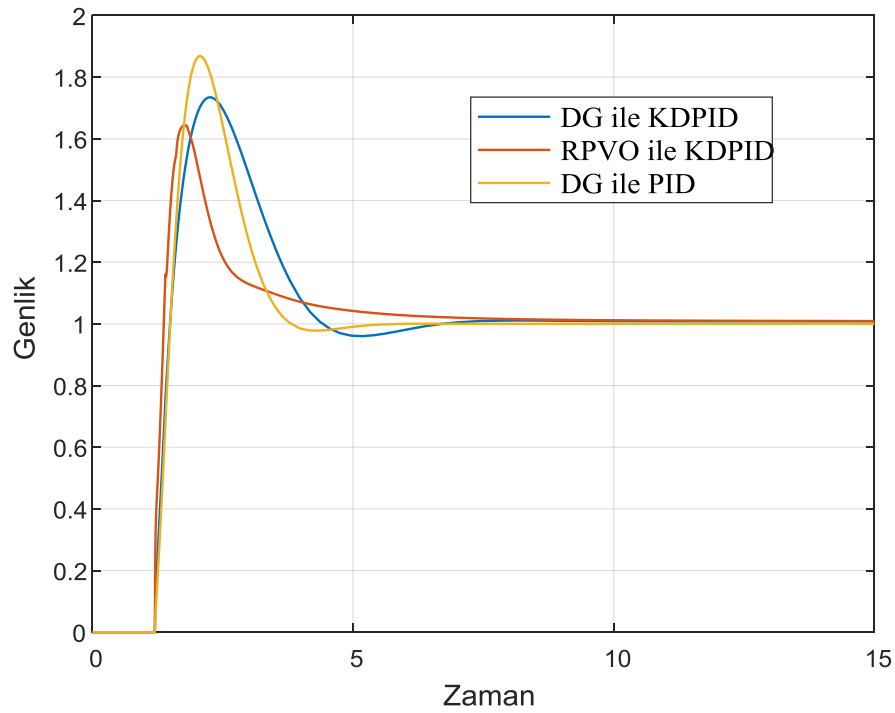
Şekil 4.15 Denetçilerin 3.8. saniyedeki birim basamak cevapları



Şekil 4.16 Denetçilerin 11.4. saniyedeki birim basamak cevapları

$P_1(s)$  matematiksel modeli için Diferansiyel gelişim algoritmasıyla elde edilen PID katsayıları  $k_p = 2.725339$ ,  $k_i = 1.624656$  ve  $k_d = 0.026691$  şeklindedir [103]. Aynı model için aynı algoritmayla elde edilmiş kesir dereceli PID

katsayıları  $k_p = 2.425323$ ,  $k_i = 1.365907$ ,  $k_d = 0.181526$ ,  $\lambda = 0.989938$  ve  $\mu = 0.828449$  şeklindedir. Aynı model için RPVO algoritmasıyla elde edilmiş olan kesir dereceli PID katsayıları  $k_p = 3.973686$ ,  $k_i = 2.144711$ ,  $k_d = 0.134777$ ,  $\lambda = 0.932582$  ve  $\mu = 1.251988$  olarak bulunmuştur. Şekil 4.17’de ise elde edilen tüm denetçilerin  $P_1(s)$  modeli için üretmiş oldukları birim basamak cevapları gösterilmiştir. RPVO algoritması diğer yöntemlere göre daha iyi sonuçlar üretmiştir.

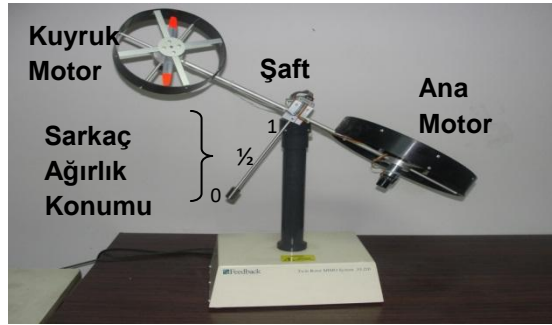


**Şekil. 4.17** Klasik PID, Diferansiyel Gelişim Algoritmasıyla Elde edilen  $PI^\lambda D^\mu$  ve RPVO algoritmasıyla elde edilen  $PI^\lambda D^\mu$  denetçilerinin birim basamak cevaplarının karşılaştırılması

#### 4.1.6 RPVO Algoritması İle Deneysel Olarak Denetçi Tasarımı

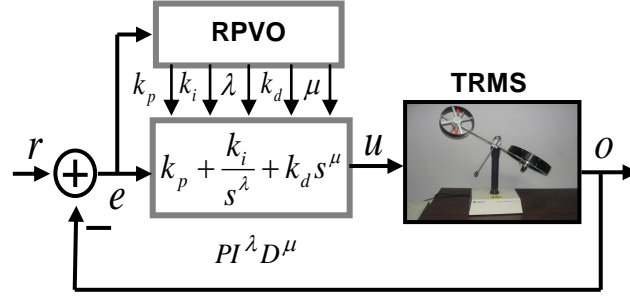
Şekil 4.18’de çift motorlu çok girişli çok çıkışlı helikopter prototipini (TRMS) laboratuvar ortamındaki kurulumu gösterilmiştir. TRMS iki DC motora sahip ortadaki şaft etrafında dönebilen uçuş kontrol simülatörüdür. Bu deney setiyle uçuş kontrol uygulamaları yapılabilmektedir. Sistem ana pervaneyi ve kuyruk pervanesini döndüren iki adet DC motora sahiptir. Ortada bulunan ağırlığın değiştirilmesiyle sistemin modeli büyük ölçüde değiştirilebilmekte ve adaptif kontrol uygulamaları yapılabilmektedir.

TRMS sistemi doğrusal olmayan bir modele sahip olduğundan dolayı kontrol edilmesi güç bir sistemdir. Çünkü ana ve kuyruk motorları da birbirini etkilemektedir. Bu durum motorların birbirini etkilemesine (coupling effect) sebep olur. Deney seti bu bileşenlerden dolayı helikopter sistemlerini laboratuvar ortamında karakterize etmektedir [66, 108].



Şekil 4.18 TRMS deney setinin laboratuvar ortamındaki kurulumu

Şekil 4.19’da RPVO algoritmasıyla TRMS sistemi için  $A-PI^{\lambda}D^{\mu}$  denetçi tasarımının yapıldığı sistemin blok şeması verilmiştir. Dikey ve yatay açısal hareketler TRMS sistemi üzerinde gerçekleştirilerek sonuçlar MATLAB ortamında görüntülenmektedir. TRMS sisteminin modeli MATLAB ortamında oluşturulmuştur. Daha sonra bu model ile eş zamanlı çalışabilen RPVO algoritması MATLAB derleyici de kodlanarak ilgili Simulink dosyasıyla eş zamanlı olarak çalıştırılmıştır [108]. Her iterasyonun sonunda hesaplanan hata değerine göre algoritma parametre vektör uzayını taramaktadır. Önceki bölümlerden de anlatıldığı gibi kesir dereceli denetçi CFE yöntemiyle 4. dereceden eşdeğeri alınarak elde edilmiştir [104].



**Şekil 4.19** RPVO algoritmasıyla TRMS için kesir dereceli denetçi tasarımı sağlayan blok şema

#### 4.1.6.1 Deneysel Sonuçlar

Genellikle literatürdeki çalışmalarda geliştirilen yöntem ile klasik yöntemin TRMS sistemi üzerinden elde edilen sonuçları karşılaştırılarak sunulur. Örneğin [109]'da dayanıklı ölü band denetçi yapısı TRMS sistemine uygulanarak klasik PID ile karşılaştırılmıştır. Genetik algoritmayla TRMS için oluşturulan denetçi PID denetçiyle karşılaştırılması [110]'da sunulmuştur. Küçük insansız hava araçları için tasarlanan kesir dereceli PI ve Ziegler–Nichols PID'nin karşılaştırılması [111]'de verilmektedir.

Bu tezde ise RPVO yöntemiyle laboratuvar ortamında TRMS uçuş kontrol sistemi için adaptif ve adaptif olmayan kesir dereceli denetçiler tasarlanmaktadır. MATLAB Simulink ve derleyici birimleri kullanılarak Şekil 4.5'de verilen RPVO yönteminin akış şeması kullanılarak elde edilen kod ile TRMS sistemine gerçek zamanlı olarak denetçiler tasarlanmıştır.

Bilindiği üzere MATLAB ortamında oluşturulan Simulink modeli sürekli zamanda çalışır. Ama gerçek hayatta sistemler zaman tabanında çalışır. Bilgisayar ile TRMS deney seti arasında dönüşüm “Advantech PCI 1711” kartıyla sağlanmıştır. Tüm deneysel sonuçlar oda şartlarında elde edilmiştir.

Algoritmanın başlangıcında ilgili denetçi parametrelerinin tarama işlemini yaptığı aralıklar her parametre için belirlenmiştir. Bu parametre aralıkları  $0 < k_p \leq 20$ ,  $0 < k_i \leq 20$ ,  $0 < k_d \leq 20$ ,  $0.5 < \lambda \leq 1.5$ ,  $0.5 < \mu \leq 1.5$  şeklindedir. Bu aralıklar sezgisel olarak belirlenmiştir.

RPVO algoritmasıyla TRMS sistemi için elde edilmiş olan  $A-PI^\lambda D^\mu$  denetçi sonuçları klasik PID sonuçlarıyla ve adaptif olmayan  $PI^\lambda D^\mu$  sonuçlarıyla karşılaştırılmaktadır. Böylece önerilen yöntemin üstünlüğü çok daha açık ortaya koyulmuştur.

Önerilen RPVO algoritması rassal başlangıç değerleriyle başlayıp ilgili sistem için istenilen yapıda denetçi tasarlayabilecek yetenektedir. Fakat bunun bazı sakıncaları bulunur. Öncelikle rassal başlangıç koşullarıyla başlanması halinde tahmin edileden daha fazla optimizasyon iterasyonunu harcamak gerekebilir. Tabi bu gerçek zamanlı denetçi tasarımı yapılan çalışmalar için istenmeyen bir durumdur. Diğer yandan gerçek zamanlı bir sistemi optimize ederken rassal değerlerle başlamak optimizasyon süresince tahmin edilemeyen ve istenmeyen sonuçlar üretebilir. Optimize edilen sistem uçuş kontrol sistemi ise kaza kırma uğraması söz konusu olabilir. Bunun için şöyle bir yöntem izlenmektedir. Rassal başlangıç değerleri kullanılarak sistemin simülasyon modeli üzerinde optimizasyon işlemi yapılmıştır. Elde edilen sonuçlar gerçek zamanlı sisteme başlangıç değeri olarak verilmesiyle gerçek zamanlı sistemde kendi dinamiklerine göre kaza kırma uğramadan ilgili denetçileri tespit edebilmektedir. Nitekim bu çalışmada da bu yöntem izlenmiştir. Öncelikle TRMS simülasyon modeli üzerinde optimizasyon başlatılarak bazı sonuçlar elde edilir. Daha sonra bu sonuçlar gerçek zamanlı TRMS sistemine başlangıç değeri olarak verilerek RPVO algoritmasıyla uygun adaptif kesir dereceli denetçiler tasarlanır.

TRMS sisteminde kullanılan PID değerleri firma tarafında sistemin üzerinde hazırda bulunan değerlerdir [66]. İlgili PID değerleri Denklem 4.12'de verilmiştir.

$$C_{PID}(s) = 5 + \frac{8}{s} + 10s \quad (4.12)$$

Adaptif olmayan kesir dereceli  $PI^\lambda D^\mu$  katsayıları TRMS sisteminin simülasyon modeli üzerinden elde edilmektedir. Denetçi parametreleri elde edilirken firmanın göndermiş olduğu Simulink modeli kullanılmıştır [66]. RPVO algoritmasıyla elde edilmiş adaptif olamayan  $PI^\lambda D^\mu$  denetçi katsayıları Denklem 4.13'de verilmektedir.

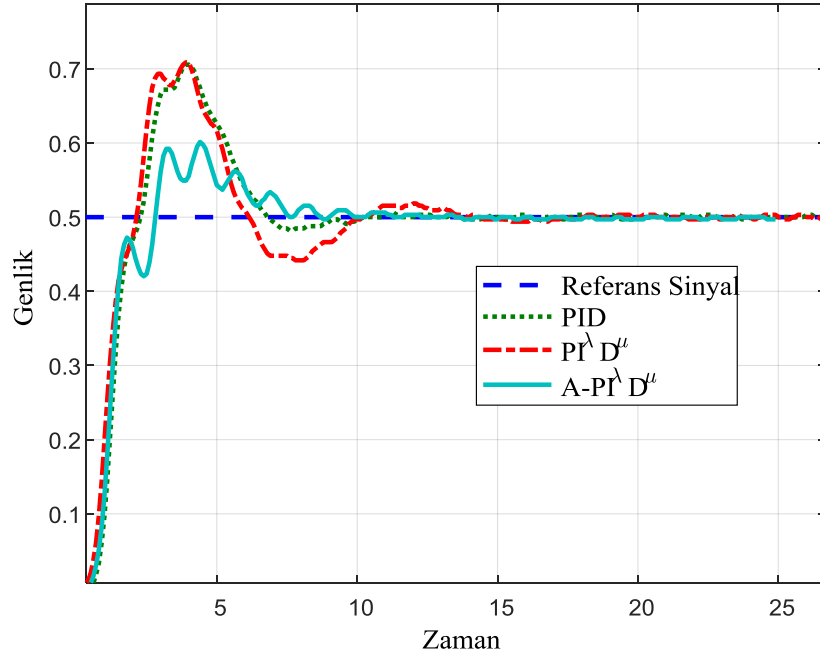
$$C_{PI^\lambda D^\mu}(s) = 9.6052 + \frac{7.7863}{s^{1.0053}} + 11.8976s^{1.0788} \quad (4.13)$$

RPVO algoritmasıyla elde edilen denetçilerin kontrol performanslarını görmek için Klasik PID, adaptif olmayan  $PI^\lambda D^\mu$  ve adaptif  $PI^\lambda D^\mu$  denetçileri TRMS sistemi üzerinde kullanılarak sonuçlar karşılaştırılmıştır.

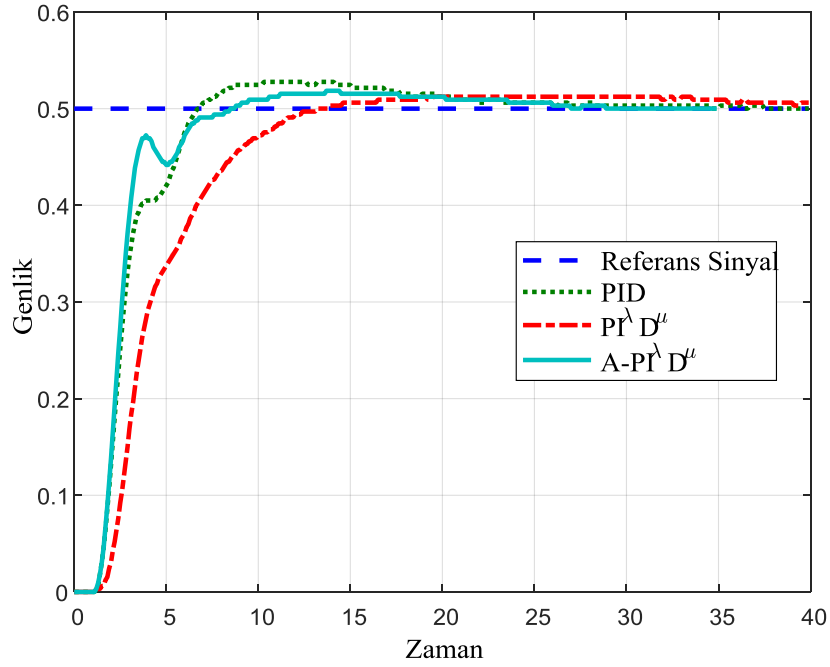
İlgili algoritmayla sonuçlar şöyle elde edilir. TRMS sisteminin Simulink modeliyle RPVO algoritması çalıştırılır. Bu optimizasyon işlemi sonucunda adaptif olmayan  $PI^\lambda D^\mu$  denetçi elde edilir. Adaptif olmamasının sebebi çevresel olarak oluşabilecek bozululara karşı dayanıklı olmamasıdır. Daha sonra bu değerler gerçek zamanlı TRMS modeli için başlangıç değeri olarak verilerek RPVO algoritması tekrar çalıştırılarak adaptif  $PI^\lambda D^\mu$  denetçisi elde edilir. Bu denetçi gerçek zamanlı sistemde çalıştığı için denetçi tasarımı yapılırken hesaplanan hata değerine her iterasyonda bozululara eklenir. Bundan dolayı adaptif bir yapı ortaya çıkar.

Diğer taraftan TRMS sisteminin orta şaftında dengeyi sağlayan bir ağırlık aparatı bulunur. Bu aparatın yerinin değişmesiyle sistem modeli de büyük ölçüde değişir. Bu tez çalışmasında bu ağırlık aparatı başa ortaya ve sona alınarak farklı TRMS modelleri için RPVO algoritmasıyla denetçiler tasarlanmıştır. Bu da kullanılan algoritmanın değişik problemlere karşı adaptif çözümler üretebilme kabiliyetini gösterir. Bilindiği gibi TRMS’de ana hareketi ve kuyruk hareketini sağlayan iki motor bulunur. Bundan dolayı bu tez çalışmasında TRMS sisteminin ana ve kuyruk motoruna ilişkin birim basamak cevapları ayrı ayrı sunulmaktadır.

Şekil 4.20 ve Şekil 4.21’de sırasıyla ana motor ve kuyruk motoru için önerilen adaptif denetçiler, klasik PID denetçiler ve adaptif olmayan  $PI^\lambda D^\mu$  denetçilerin birim basamak cevapları karşılaştırmalı olarak verilmektedir. Şekillerden de anlaşıldığı gibi adaptif denetçiler diğer denetçilere karşı daha üstün kontrol performansı üretebilmektedir. Özellikle A- $PI^\lambda D^\mu$  denetçi TRMS sistemi için düşük aşımalar vererek kısa yerleşme zamanına sahip olduğu görülmektedir [82].



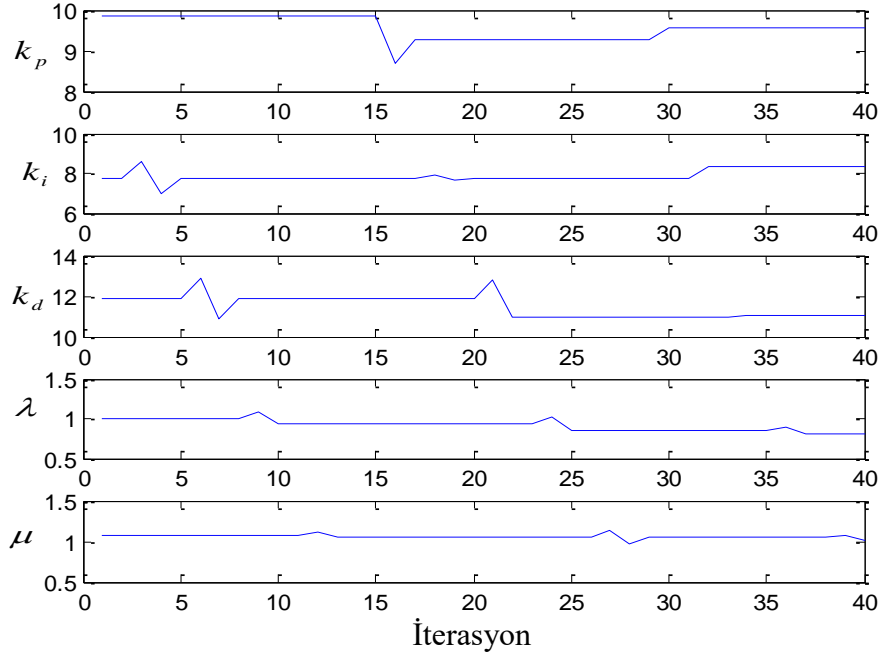
Şekil 4.20 TRMS ana motoru için Klasik PID,  $PI^\lambda D^\mu$  ve  $A-PI^\lambda D^\mu$  denetçilerinin birim basamak cevapları



Şekil 4.21 TRMS kuyruk motoru için Klasik PID,  $PI^\lambda D^\mu$  ve  $A-PI^\lambda D^\mu$  denetçilerinin birim basamak cevapları

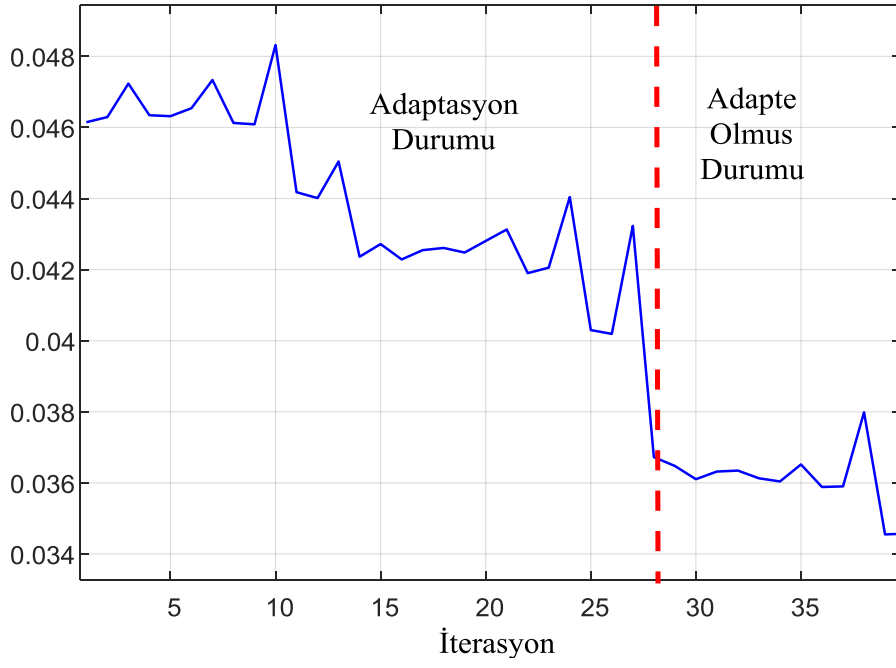
Şekil 4.22'de optimizasyon esnasında kesir dereceli denetçinin parametre değişimi gösterilmiştir. Şekil 4.23'de ise optimizasyon süresince hata fonksiyonun

değişimi sunulmuştur. Şekillerden de görüldüğü gibi RPVO algoritması her zaman hatayı düşürme yönünde hareketler yapmaktadır. Optimizasyon esnasında hatayı büyütecek yönde hareketler olmasına karşın takip eden iterasyonlarda bu yönlerden dönülerek hatayı daha fazla düşürülmeye çalışılmaktadır. Şekilden de görüldüğü üzere algoritma 28. iterasyonda adaptasyon sağlamıştır. Algoritma takip eden iterasyonlarda ise hassas ayarlar yapmaktadır.

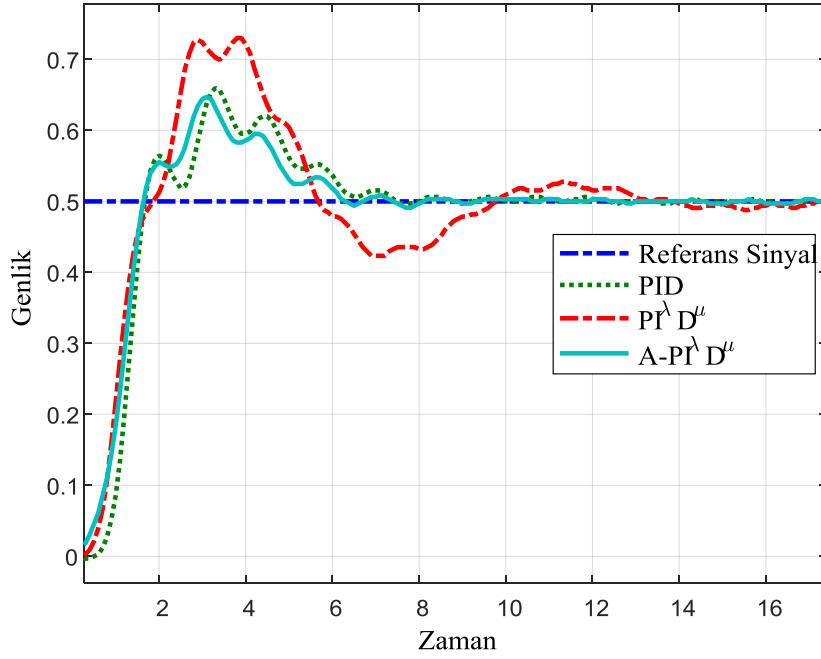


**Şekil 4.22** TRMS ana motoru için tasarlanan  $A-PI^\lambda D^\mu$  denetçinin optimizasyon süresince parametre değişimi

RPVO algoritmasının adaptif sonuçlar üretebildiğini göstermek için TRMS deney setinin orta kısmında bulunan ağırlık aparatının yeri belli aralıklarla değiştirilmiştir. Bu ağırlık değişimlerinde sonra algoritma tekrar çalıştırılmıştır. Her seferinde algoritma şartlar değişse bile optimal sonuçlar üretebilir. Şekil 4.24'de ağırlık aparatının orta da olma durumunda RPVO algoritmasıyla elde edilen denetçinin birim basamak cevapları sunulmaktadır.



Şekil 4.23 Optimizasyon süresince elde edilen hata fonksiyonunun değişimi



Şekil 4.24 Ağırlık aparatının ortada olma durumunda elde edilen birim basamak cevabı

Çizelge 4.1’de ağırlık aparatının değişik pozisyonlardaki durumlarına ilişkin RPVO algoritmasıyla elde edilen kesir dereceli denetçi katsayıları sunulmaktadır. Çizelge 4.2’de ise ağırlık aparatının değişik konumlarına ilişkin hata değerlerinin ortalamaları bulunmaktadır. Bu değerlerin incelenmesiyle de hangi denetçinin daha iyi sonuç ürettiği görülür. Bu uygulamada adaptif KD denetçi yapısı diğer denetçilere göre daha iyi bir kontrol performansı üretebilmektedir.

**Çizelge 4.1** Ağırlık aparatının farklı konumlarına göre RPVO algoritmasıyla elde edilen  $A-PI^\lambda D^\mu$  denetçi katsayıları

Denge Konumu	Aparatının	$k_p$	$k_i$	$k_d$	$\lambda$	$\mu$
0		9.8965	7.1349	11.8178	0.8136	1.1356
1/4		8.8939	7.6710	11.2421	1.0053	1.0082
1/2		9.8939	8.3063	11.6235	0.9352	1.0135
3/4		8.8881	8.0648	11.8976	0.9139	1.0155
1		9.5436	8.3829	11.0123	0.8053	1.0138

**Çizelge 4.2.** Ağırlık aparatının farklı konumlarına ilişkin PID  $PI^\lambda D^\mu$  ve  $A-PI^\lambda D^\mu$  denetçilerin 100 saniye için üretmiş oldukları hata değerleri

Denge Konumu	Aparatının	Hata Değerleri		
		PID	$PI^\lambda D^\mu$	$A-PI^\lambda D^\mu$
0		0.0261	0.0205	0.0202
1/4		0.0193	0.0179	0.0172
1/2		0.0196	0.0183	0.0153
3/4		0.0199	0.0231	0.0176
1		0.0195	0.0200	0.0176

Tezdeki bu çalışmayla gerçek zamanlı sistemlere adaptif kesir dereceli denetçi tasarlayabilen RPVO yöntemi önerilmektedir. Bu yöntem özellikle denetçi tasarımında çok etkin bir şekilde çalıştığı yapılan akademik çalışmalarla ispatlanmıştır. Tezde tekrar olmaması açısından sadece konunun ana fikrini anlatan çalışmalara yer verilmektedir. RPVO yöntemi literatürce kabul görmüş ve konuyla ilgili birçok akademik çalışmalar yapılmıştır [82, 107, 112, 113]. Bunun yanı sıra bu yöntemle yapılan yayınlar yüksek oranda atıf almaktadır. Tezin bu bölümüyle ilgili SCI- indekse giren dergilerde 2 adet makale ve birçok konferans çalışması yapılmıştır. Tezin takip eden bölümünde yine RPVO algoritmasıyla farklı bir kontrol yapısına ilişkin denetçi tasarımı gerçekleştirilecektir.

#### **4.2. RPVO Algoritmasıyla İki Serbestlik Dereceli Kontrol Yapıları İçin Kesir Dereceli Denetçi Tasarımı**

Tezin bu bölümünde önceki bölümde ayrıntısıyla verilmiş olan RPVO algoritması kullanılarak literatürde bulunan iki serbestlik dereceli (2DOF-Two Degree of Freedom) kontrol yapılarına KDPID'ler tasarlanmıştır. RPVO algoritması bir önceki bölümde detaylı olarak anlatıldığı için bu bölümde sadece uygulamanın nasıl gerçekleştirildiği gösterilecektir [112, 113].

Bu bölümde de uygulama platformu olarak TRMS sisteminin matematiksel modeli kullanılarak ilgili kontrol yapılarının kontrol performansına olan etkileri tespit edilmektedir.

Genellikle tek serbestlik dereceli sistemler (One Degree of Freedom) sistemdeki parametre değişimlerine ve sisteme uygulanan bozululara anlık tepki gerektiren durumlara hızlı cevap veremeyebilir. Fakat iki serbestlik dereceli sistemler kontrol yapılarından dolayı bu tarz etkilere hızlı tepki verebilir [112-114].

İki serbestlik dereceli (2SD) kontrol yapıların ilk olarak 1948 yılında Horowitz tarafından önerilmiştir [114]. İki serbestlik dereceli sistemlerin ana avantajı kalıcı hal hatasının dayanıklı bir şekilde sifıra zorlamasıdır. Bu durum kontrol performansı açısından ciddi bir avantaj olarak gösterilir [115, 116]. Araki ve Taguchi tarafından iki serbestlik dereceli kontrol yapısı ile klasik PID'nin kullanıldığı bu alandaki çok önemli bir çalışma [116]'da sunulmaktadır. Bunun yanı sıra literatürde farkı iki serbestlik dereceleriyle PID denetçinin kullanıldığı birçok çalışma bulunmaktadır [117-120].

Bundan dolayı iki serbestlik dereceli kontrol yapılarının çalışmalarda kullanılması literatürde halen geçerliliğini korumaktadır. Bilindiği üzere kesir dereceli türev ve integral hassas ayarlamalar yapmaya fırsat verdiği için literatürde kontrol performansını iyileştirmeye dönük birçok çalışmada kullanılır [41-43, 121]. Feng ve arkadaşları kesir dereceli PID ile tam sayı dereceli PID'yi 2SD kontrol yapısı içerisinde kullanarak yaptıkları karşılaştırmaları [122]'de sunmaktadır.

Bunun yanı sıra 2SD-KD kontrol yapısının otomatik üretim kontrolünde kullanılması [123]'de gösterilmektedir.

Bilindiği gibi kesir dereceli denetçi fikri ilk olarak 1999 yılında Podlubny tarafından önerilmiştir [47]. Bunun yanı sıra literatürde KDPID parametrelerinin belirlendiği birçok çalışmada bulunmaktadır [94, 124-126].

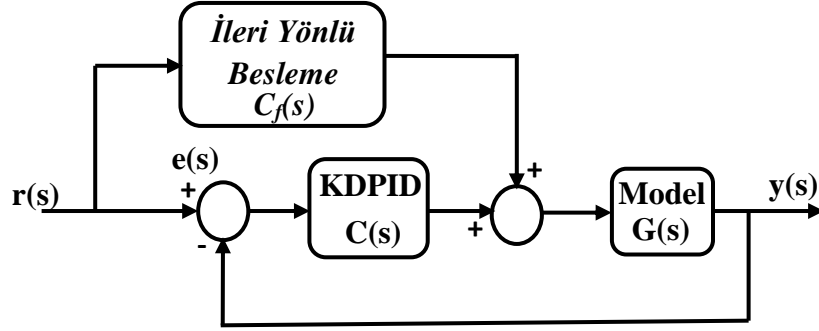
Tezin bu bölümünde ise 2SD-KDPID kontrol yapısının parametreleri RPVO yöntemiyle belirlenmektedir [82, 83, 127].

Bu çalışmada 5 farklı iki serbestlik dereceli kontrol yapısı kullanılmıştır. Bunlar Geri Besleme (FeedBack (FB)), İleri Yönlü (Feed-Forward (FF)), Bileşenlerine Ayrılmış (Component Separated (CS)), Ayar Noktası Filtresi (Set-Point Filter (SPF)) Türev Öncesi Filtreleme (Filter Preceded-Derivative (FPD)) gibi iki serbestlik dereceli kontrol yapılarıdır [116]. Bu kontrol yapıları KDPID ile modifiye edilerek ilgili sistem üzerinde kontrol performansları karşılaştırılmaktadır. İki serbestlik dereceli kontrol yapıları incelendiğinde optimize edilmesi gereken 2 parametre bulunur. Bunu yanı sıra KDPID'inde optimize edilmesi gereken 5 parametresi bulunur. Yani toplamda 7 parametrenin optimize edilmesi gerekir. Bu yedi parametre RPVO algoritmasıyla optimize edilmiştir. Kapalı döngü kontrol sisteminde TRMS uçuş kontrol deney setinin matematiksel modeli kullanılmaktadır.

#### **4.2.1. İki Serbestlik Dereceli Kesir Dereceli Kontrol Döngüsünün Optimizasyonu**

Bu bölümde 2SD kontrol yapılarının KDPID ile kullanımı matematiksel olarak anlatılmaktadır. Bu işlem 2SD kontrol yapılarında bulunan klasik PID'ler KDPID'ler ile değiştirilmektedir. Bu değişimden sonra 2SD sistemlerin denetçi formülasyonları da matematiksel olarak değiştirilir. Elde edilen bu modeller TRMS sistemi üzerinde kullanılarak RPVO algoritmasıyla optimize edilir. Şekil.

4.25’de İleri yönlü (FF tipi) 2SD kontrol yapısı görülmektedir. Bu yapı KDPID denetçinin çıkışına ileri beslemenin eklenmesiyle elde edilmiştir.



Şekil 4.25 İleri Yönlü İki Serbestlik Dereceli KDPID yapısı

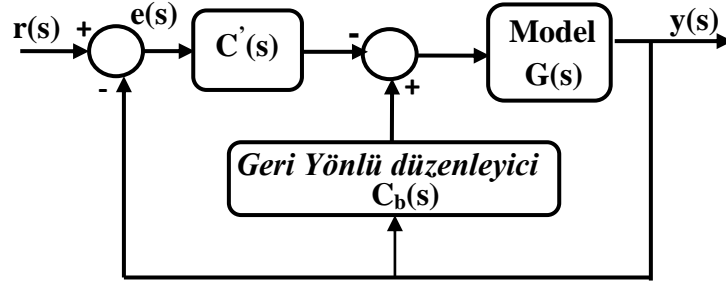
İleri yönlü beslemenin matematiksel formu KD türev ve integral ile aşağıdaki gibi elde edilir [116].

$$C_f(s) = -k_p \alpha - \beta k_d s^u \quad (4.14)$$

Denklemdaki  $\alpha$ , oransal kazancı gösterir ( $0 < \alpha < 1$ ).  $\beta$ , türevsel kazancı gösterir ( $1 < \beta < 2$ ).

Şekil 4.26’da Geri Besleme tipi (FB) 2SD kontrol yapısını gösterir.  $C_b(s)$ , geri besleme dengeleyicisi olarak adlandırılır. Denklem 4.15’de  $C'(s)$ , denetçinin matematiksel formunun KD integro diferansiyel ifadelerle güncellenmiş hali bulunmaktadır [116].

$$C'(s) = (1 - \alpha)k_p + \frac{k_i}{s^2} + (1 - \beta)k_d s^u \quad (4.15)$$

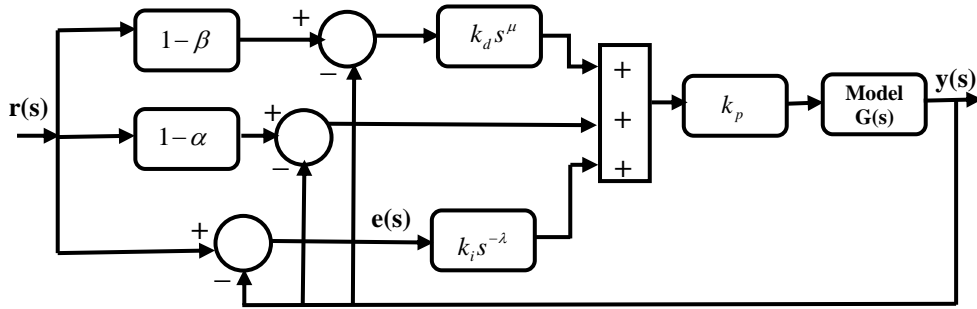


Şekil 4.26 Geri Yönlü İki Serbestlik Dereceli KDPID yapısı

Denlem 4.16'da geri besleme dengeleyicisi olan  $C_b(s)$ 'nin KD integro diferansiyel ifadelerle güncellenmiş hali bulunmaktadır.

$$C_b(s) = k_p \alpha + k_d \beta s^\mu \quad (4.16)$$

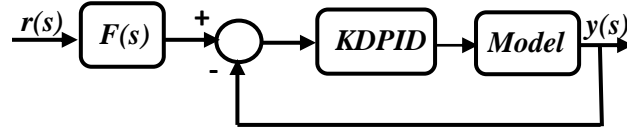
Şekil 4.27'de ise CS tipi 2SD kontrol yapısının KD integro diferansiyel ifadelerle güncellenmiş hali bulunur [116].



Şekil 4.27 CS tipi 2SD KDPID yapısı

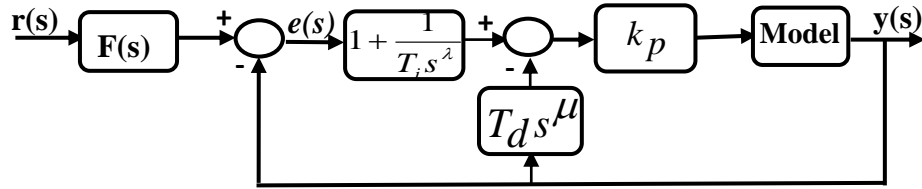
SPF tipi 2SD kontrol yapısının KDPID ile güncellenmiş hali Şekil 4.28'de bulunur [116].

$$F(s) = \frac{1 + (1 - \alpha)T_i s^\lambda + (1 - \beta)T_i T_d s^\mu}{1 + T_i s^\lambda + T_i T_d s^\mu} \quad (4.17)$$



Şekil 4.28 SPF tipi İki Serbestlik Dereceli KDPID yapısı

FPD tipi 2SD kontrol yapısının KDPID ile güncellenmiş hali Şekil 4.29’da bulunur.



Şekil 4.29 FPD tipi 2SD KDPID yapısı

$F(s)$ , KD türevle modifiye edilerek denklem 4.18 elde edilir.

$$F(s) = \frac{1 + (1 - \alpha)T_i s^\lambda + (1 - \beta)T_i T_d s^\mu}{1 + T_i s^\lambda} \quad (4.18)$$

Denklemler ve modellerden de anlaşılacağı üzere 2SD KDPID denetçi yapısında optimize edilmesi gereken 7 parametre bulunur.

KDPID denetçi kullanıldığında karşımıza çıkan diğer problem ise KD denetçinin pratik olarak gerçekleştirilme zorluğudur. Bilindiği gibi literatürde kesir dereceli PID’nin pratik olarak gerçekleştirilmesini sağlayan birçok yöntem bulunur [48, 64, 106]. Fakat bu çalışmada Valerio’nun MATLAB için geliştirmiş olduğu KD araç kutusu kullanılmaktadır [64]. Valerio’nun araç kutusu literatürde bulunan birçok yaklaşım metodunu bünyesinde bulundurduğundan dolayı gerçekleştirme işlemi daha kolay yapılabilmektedir. Bunun yanı sıra diğer bir avantaj ise bu araç kutusu MATLAB simulink ortamında rahatlıkla kullanılabilir.

#### 4.2.2 İki Serbestlik Dereceli Kesir Dereceli PID Kontrol Döngüsünün Çift Motorlu Helikopter Prototipi Üzerinde Uygulanması

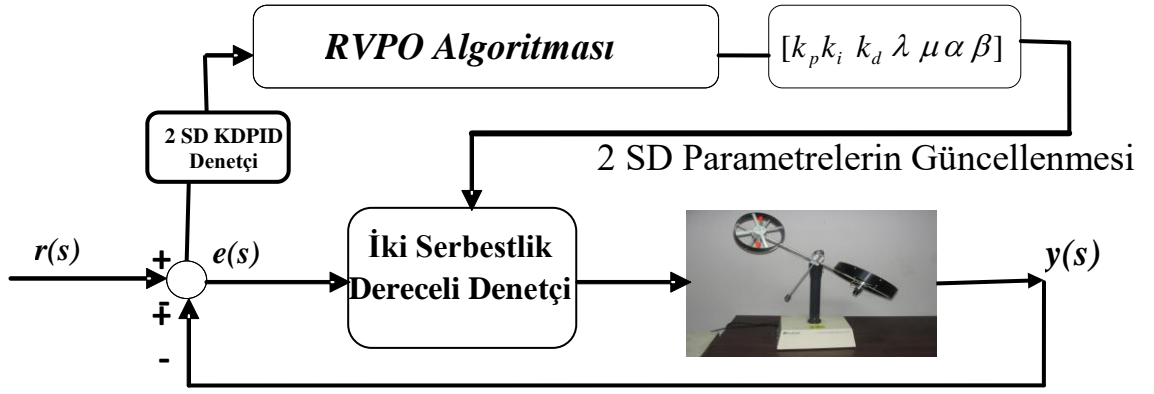
Tezin bu bölümünde uygulama platformu olarak TRMS helikopter prototipi kullanılmaktadır. Çalışmada önerilen kontrol yapılarının simülasyon ortamında test edilebilmesi için deney setinin MATLAB ortamında matematiksel modeli çıkarılmıştır. Bu matematiksel model için RPVO algoritmasıyla 7 parametrenin optimizasyonu gerçekleştirilmektedir. Elde edilen sonuçlar gerçek zamanlı TRMS'nin modeli üzerinde kullanılarak karşılaştırmalı olarak sunulmaktadır.

#### 4.2.3 TRMS Sisteminin Matematiksel Modelinin Çıkarılması

Bu çalışmada uygulamalar TRMS sisteminin ana motoru üzerinden gerçekleştirilmiştir. Ana motora çoklu sinusoidal dalga 100 sn süresince 0.1 örnekleme aralığında uygulanmıştır. MATLAB ortamında uygulanan girişe ilişkin çıkışlar kaydedilmiştir. Elde edilen bu veriler MATLAB System Identification araç kutusuna verilerek sistemin doğrusal parametrik model yapısına göre modeli Denklem 4.19'da gösterildiği gibi elde edilmiştir.

$$G_{TRMS}(s) = \frac{0.06725 + 1.359s}{s^2 + 0.7906s + 3.666} \quad (4.19)$$

Daha sonra elde edilen bu model önceki başlıklarda tanımlanmış olan 5 farklı iki serbestlik dereceli kontrol yapısında sistemin modeli olarak kullanılmıştır. Her model için RPVO algoritması aynı şartlarda çalıştırılarak sisteme ilişkin optimal denetçiler elde edilmiştir. Şekil 4.30'da optimizasyon süresince kullanılan yapının blok şeması gösterilmektedir.



Şekil 4.30 Optimizasyon süresince kullanılan sistemin blok şeması

#### 4.2.4 Simülasyon Sonuçları

Denklem 4.19’da verilen eşitlik ikinci dereceden TRMS ana motorunun matematiksel modelini karakterize etmektedir. Şekil 4.23 – Şekil 4.27 arasında verilen FF, FB, CS, SPF ve FPD tipi 2SD KDPID kontrol yapıları Şekil 4.30’da verilen sistem içerisinde sırasıyla kullanılarak optimizasyon işlemi gerçekleştirilmiştir. En iyi amaç fonksiyonu optimizasyon süresince elde edilmektedir. Simülasyon esnasında simulink modeli 15 sn. boyunca MATLAB ode45 çözümleyicisi kullanılarak elde edilmektedir. Diğer bir önemli konu ise optimizasyon esnasında kullanılan amaç fonksiyonudur. Bu fonksiyon sistemin iyileşmenin bir ölçüsüdür.

Çalışma boyunca kullanılan amaç fonksiyonu sistemde üretilen hata fonksiyonudur. İlgili hata fonksiyonu denklem 4.20’de verilmektedir.

$$J(\theta) = \min \frac{1}{N} \sum e^2 \quad (4.20)$$

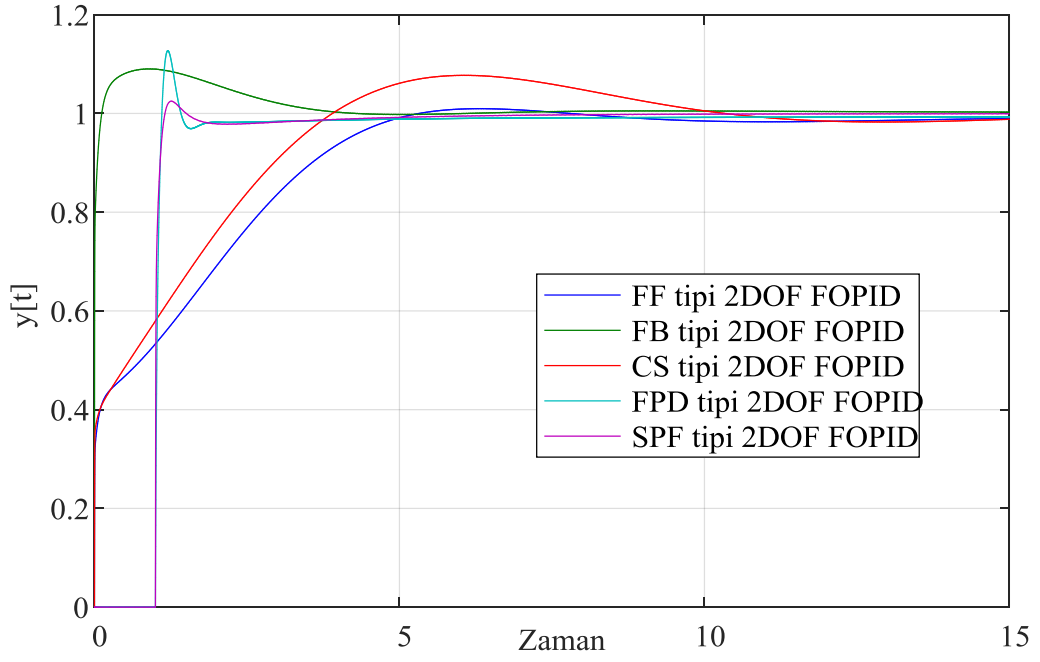
Denklemdeki  $e$  gerçek hata değerini göstermektedir. Bu hata değeri kullanılarak optimizasyonla elde edilen FF, FB, CS, SPF ve FPD tipi 2SD KDPID kontrol yapılarının değerleri Çizelge 4.3 de verilmektedir.

**Çizelge 4.3** 5 Farklı tip 2SD KDPID denetçilerin RPVO algoritmasıyla elde edilen değerleri

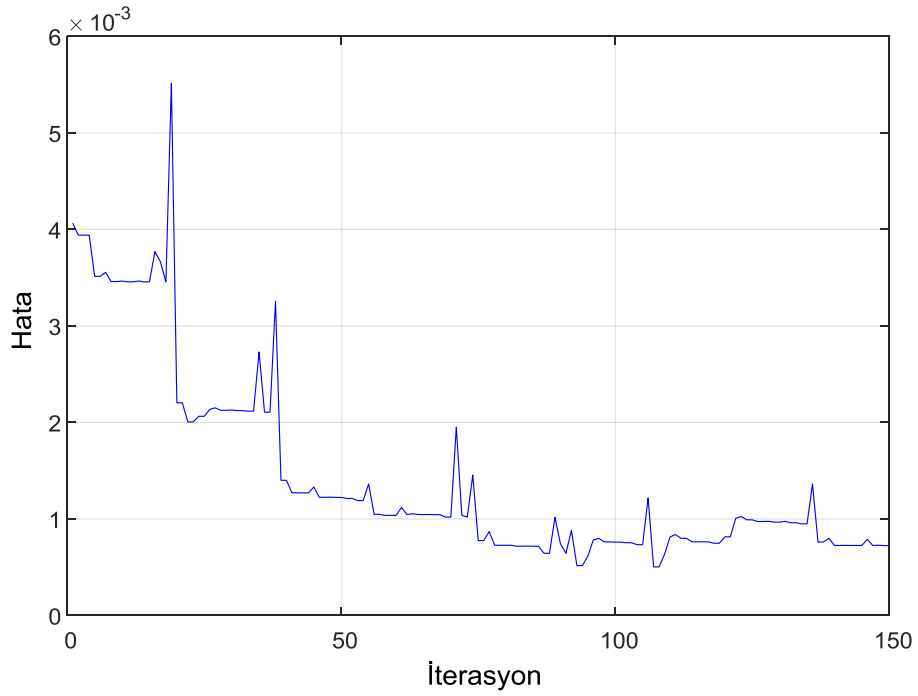
	<b>FF Tipi 2SD KDPID</b>	<b>FB Tipi 2SD KDPID</b>	<b>CS Tipi 2SD KDPID</b>	<b>SPF Tipi 2SD KDPID</b>	<b>FPD Tipi 2SD KDPID</b>
$k_p$	7.9888	10.8318	15.9728	79.8708	63.0741
$k_i$	3.9993	3.3221	3.9337	31.8713	48.3935
$k_d$	13.3757	12.5241	12.2003	24.4112	33.1517
$\lambda$	0.9698	0.9374	0.5690	0.9334	0.5323
$\mu$	1.2454	1.2845	0.7529	0.9722	0.5953
$\alpha$	0.3169	0.3077	0.1356	2.1217	0.9906
$\beta$	0.5422	0.1300	0.6106	0.7555	0.3057
<b>Amaç Fonksiyonu (e)</b>	0.0037	$5.03 \cdot 10^{-5}$	0.0119	0.0082	$1.62 \cdot 10^{-4}$

Şekil 4.31’de Çizelge 4.3’de verilmiş olan FF, FB, CS, SPF ve FPD tipi 2SD KDPID kontrol yapılarının TRMS’nin ana motor modeliyle oluşturulan kontrol döngüsünün birim basamak cevapları karşılaştırmalı olarak verilmektedir. Şekil 4.32’de optimizasyon esnasında hata fonksiyonun değişimini gösteren grafik sunulmaktadır.

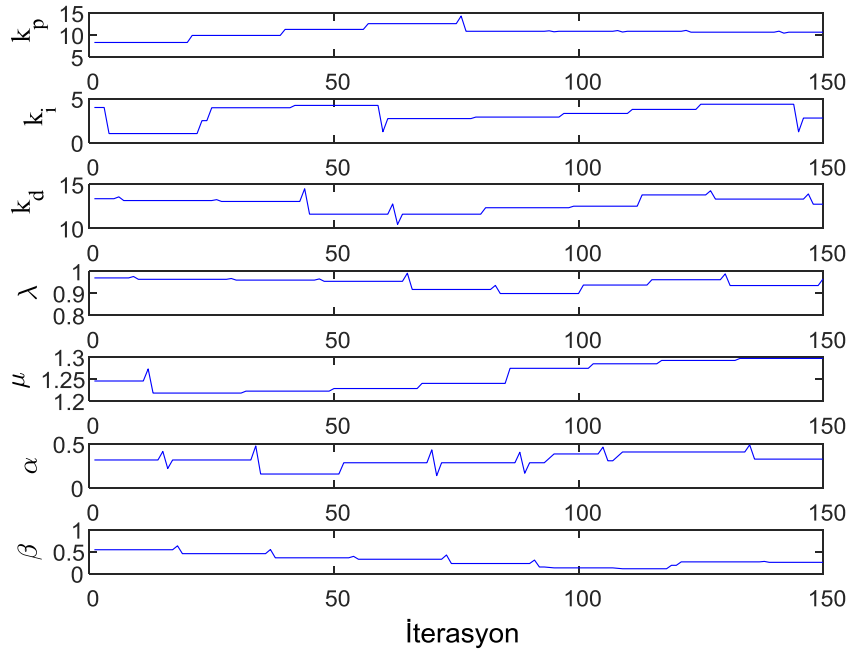
Şekil 4.33’de FB tipi 2SD KDPID kontrol döngüsünün optimizasyon süresince oluşan parametrelerin değişimleri gösterilmektedir. Şekil 4.34’de ise tüm 2SD kontrol yaklaşımlarının ürettiği hata değerleri karşılaştırmalı olarak verilmektedir. Bu grafiğe göre TRMS’nin ana motorunun matematiksel modeli için en iyi sonucu FB tipi kontrol yapısının ürettiği gösterilmektedir.



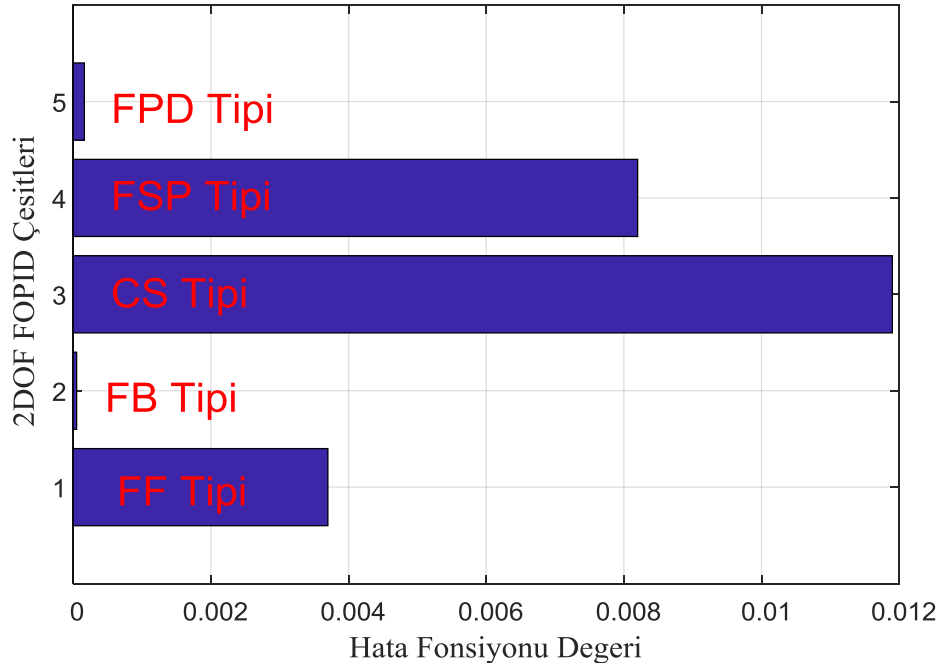
**Şekil 4.31** TRMS ana motoru için FF, FB, CS, SPF, FPD tipi 2SD KDPID denetçilerin birim basamak cevapları.



**Şekil 4.32** Optimizasyon süresince FB tipi kontrol yapısı için elde edilen hata fonksiyonunu değişimi



**Şekil 4.33** Optimizasyon süresince FB 2SD KDPID denetçinin parametre değişimi

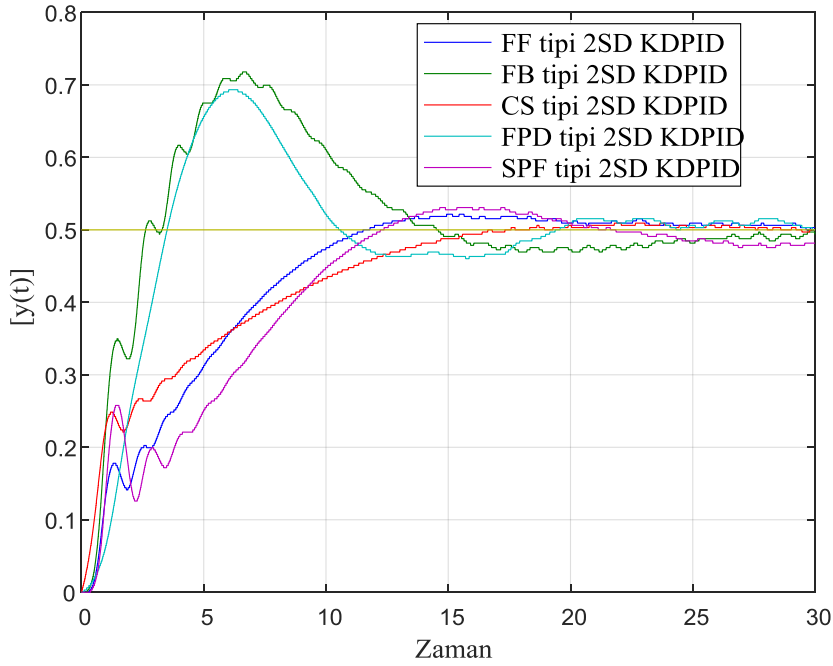


**Şekil 4.34** Üretilen hata değerlerinin karşılaştırılması

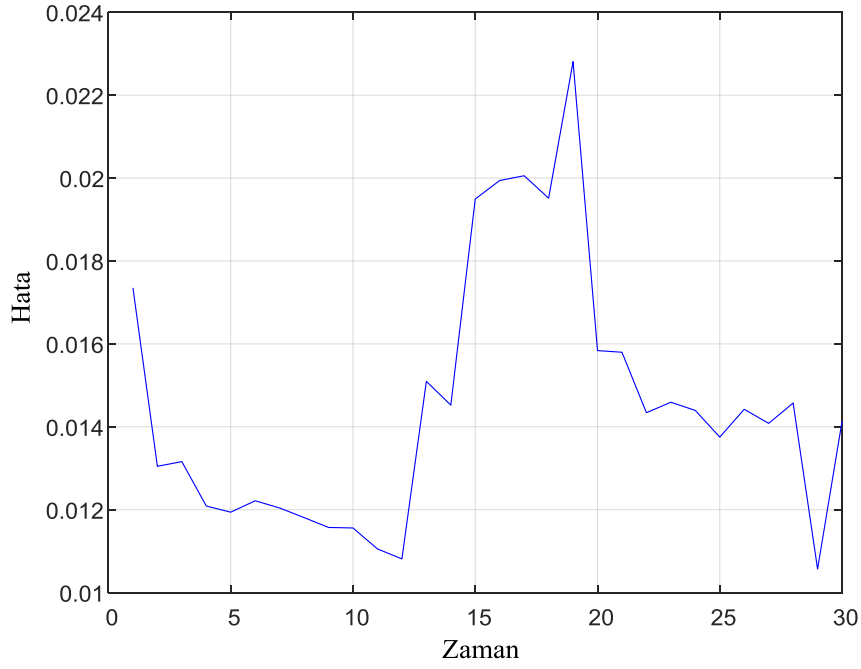
#### 4.2.5 Deneysel Çalışma

Bu bölümde RPVO algoritmasıyla 2SD KD kontrol yapılarının parametreleri TRMS'nin gerçek zamanlı modeli üzerinden elde edilmiştir. Algoritma önceki bölümlerde bahsedildiği gibi kullanılan kontrol yapısı TRMS gerçek zamanlı sisteme uygulanarak her yapı için 7 parametrenin optimizasyonu gerçekleştirilmiştir. Gerçek zamanlı sistemin optimizasyonu ile elde edilen sistem parametreleri Çizelge 4.4'de verilmiştir. Şekil 4.35'de TRMS'nin gerçek zamanlı modeli için optimizasyon ile elde edilen sonuçlar karşılaştırmalı olarak sunulmuştur.

Bu şekilden de anlaşılacağı üzere FF tipi 2SD sistem yaklaşımı TRMS gerçek zamanlı modeli için en iyi sonucu üretmektedir. Şekil 4.36'da ise FF tipi yaklaşımın optimizasyon süresince hata değişim grafiği verilmektedir.



Şekil 4.35 5 tip iki serbestlik dereceli kesir dereceli denetçi yapısının birim basamak cevaplarının karşılaştırılması



**Şekil 4.36** Optimizasyon süresince FF tipi 2SD KDPID denetçi yapısının hata fonksiyonu değişimi.

**Çizelge 4.4** Gerçek zamanlı TRMS modeli için RPVO algoritmasıyla üretilen 5 farklı 2SD PID denetçi parametreleri

	FF Tipi 2SD KDPID	FB Tipi 2SD KDPID	CS Tipi 2SD KDPID	SPF Tipi 2SD KDPID	FPD Tipi 2SD KDPID
$k_p$	5.458272	8.918985	0.959310	1.814724	0.951462
$k_i$	1.323994	1.770751	1.604100	0.800000	0.871423
$k_d$	18.885357	18.071423	16.640595	13.253974	3.198259
$\lambda$	1.145667	1.064638	1.124488	1.046451	1.032212
$\mu$	1.127315	1.050513	1.093880	1.071601	1.151325
$\alpha$	0.830327	0.866294	0.529064	0.519508	0.816883
$\beta$	5.000000	5.000000	4.929064	4.972150	4.904051
<b>Amaç Fonksiyonu (e)</b>	0.0106	0.0232	0.0132	0.1196	0.0125

Tezin bu bölümünde literatürde önerilmiş olan RPVO algoritmasıyla literatürde daha önceden denenmemiş kontrol yapıları için daha iyi kontrol performansı üretebilen denetçilerin tasarlanabileceği gösterilmiştir. Bunun yanı sıra RPVO algoritmasının en belirgin özelliği ise gerçek zamanlı sistemlerde de optimizasyon işlemini gerçekleştirebilmesidir. Böylece sistemlere adaptif denetçiler tasarlanabilmektedir.

Bir sonraki bölümde önceden kesir dereceli PID tasarımı için kullanılmamış bir yöntem olan Tabu Arama Algoritmasından (Tabu Search Algorithm-TSA) faydalanılarak literatürdeki bazı matematiksel modellere KDPID tasarımı yapabilen yeni bir yöntem önerilecektir. Önerilen yöntem ile Tabu Arama Algoritmasının temel denklemlerinden faydalanmakla birlikte optimizasyon performansını artırmak için yeni tanımlamalar yapılacaktır.

## 5. TABU ARAMA ALGORİTMASI TABANLI BİR OPTİMİZASYON YÖNTEMİYLE KESİR DERECELİ DENETÇİ TASARIMI

Bu bölümde Tabu Arama Algoritması (TAA- Tabu Search Algorithm- TSA) modifiye edilerek literatürde olan sistemler için KDPID denetçiler tasarlanmıştır. Aslında literatürde TAA ile bazı matematiksel sistem modelleri için klasik PID denetçilerin tasarlandığı çalışmalar bulunmaktadır. TAA'nın temel felsefesine bağlı kalmakla birlikte yeni modifikasyonlar yapılarak KDPID tasarımında kullanılmıştır. Önerilen bu yöntemi sadece TAA şeklinde adlandırmak yerine Tabu Arama Algoritmasını temel alan bir yöntem olarak nitelendirmek ve adlandırmak daha doğru olacaktır [128].

Kesir dereceli matematiksel ifadelerin pratik olarak gerçekleştirilme ihtimali ortaya çıktıktan sonra araştırmacılar tarafından birçok alanda sıklıkla kullanılmaktadır. Özellikle kesir dereceli matematiğin denetçi yapılarında kullanılmasıyla klasik PID denetçilere olan üstünlükleri gösterilmiştir [91-131].

Yeroğlu ve Tan'ın KDPID parametrelerini ayarladıkları bir çalışma [106]'da bulunmaktadır. Yeroğlu ve Kavuran'ın kayan kipli denetim yapısı ile sistem kontrolünü sağladıkları yeni bir yöntem [132]'de sunulmuştur. Alagöz ve arkadaşları tarafından KD referans modele göre optimizasyon ile PID parametrelerinin elde edildiği çalışma [83]'de verilmektedir.

Optimizasyon yöntemleri etkili kontrol performansı veren denetçileri tasarlamamızı sağlar. Özellikle iteratif sezgisel yöntemlerle sürekli lokal optimum noktalardan kaçılarak, global optimum noktalar taranır [14,128]. Tezin bu bölümünde TAA tabanlı bir yöntem kullanılarak KDPID parametreleri optimizasyonla elde edilmektedir. [141]'de Klasik PID denetçi parametrelerinin TAA ile optimize edildiği çalışma gösterilmektedir. Sonuç olarak TAA ile KDPID parametresini optimize etmek amacıyla yapılan bir çalışma literatüre katkı sağlayacak niteliktedir.

Meta sezgisel algoritmalar zeki arama algoritmalarının modern formudur. Meta-sezgisel algoritmalar 3 temel prensibe göre çalışır. Bunlar adaptif hafızanın kullanımı, komşulukların keşfi ve elde edilen çözümün bir sonraki iterasyona aktarılmasıdır [14, 133-137].

Litertürde bulunan birçok optimizasyon problemi için TAA ile hibrit çözümler sunulmuştur [138-141].

Tezin bu bölümünde TAA'nın temel bileşenleri ve bu bileşenlerin nasıl yorumlandığına ilişkin bölümler sunulmaktadır. Daha sonra optimizasyon işleminin nasıl gerçekleştirildiği ayrıntılarıyla anlatılmaktadır.

### **5.1. Tabu Arama Algoritması**

TAA'nın temel prensipleri ilk olarak 1989 yılında Fred Glover tarafından önerilmiştir [133]. TAA'nın optimizasyon problemleri için meta stratejisi [134]'de sunulmaktadır. Daha sonra F. Glover ve M. Laguna 1997 yılında TAA'nın kullanıldığı alanları ve uygulamalarını gösteren kitabı yayınlamışlardır [14]. TAA, telekomünikasyon, paralel hesaplama, ulaşım, sürekli ve rassal optimizasyon, üretimlerin finansal analizi, planlama, takvimleme gibi birçok alanda kullanılır [14]. Birçok araştırmacı [14,133,134]'de belirtilen TAA'nın temel prensiplerine göre farklı uygulamaları gerçekleştirmiştir. Birçok güncel çalışmada halen TAA ile başka optimizasyon algoritmaları birleştirilerek optimizasyon problemleri çözümlenmektedir.

Örneğin, [138]'de karınca koloni algoritmasıyla TAA birleştirilerek yeni bir yöntem önerilmiştir. TAA ile genetik algoritma birleştirilerek üretim model planı problemi [139]'de çözümlenmektedir. Gürültü ortamının parametrelerini optimizasyonu genetik algoritma ve TAA kullanılarak [140]'da gösterilmektedir.

TAA ile parçacık sürü optimizasyon algoritmasının birleştirilmesiyle bulanık denetçinin oluşturulduğu çalışma [142]'de sunulmaktadır. Global sürekli zamanlı optimizasyon probleminin TAA ve Nelder-Mead arama algoritmasıyla çözümlendiği çalışma [143]'de gösterilmektedir. [144]'de ise araçların yönlendirme problemleri TAA ile çözümlenmektedir. Dağıtım ağının optimizasyonunun TAA ile yapıldığı çalışma [145]'de sunulmaktadır.

TAA zeki ve sezgisel bir optimizasyon algoritmasıdır. Tümleşik optimizasyon problemlerinin çözümünde kullanılır. Bundan dolayı TAA adaptif bir optimizasyon algoritması olarak adlandırılabilir.

Tümleşik optimizasyon problemleri, Denklem (5.1)'de verilen amaç fonksiyonuna göre çalışır. Genellikle optimizasyon problemlerinde bu amaç fonksiyonu hata fonksiyonudur ve bunu minimize etmeye dönük aramalar yapılır.

$$\min f(x) : x \in R \quad (5.1)$$

TAA yöntemi minimizasyon ve maksimizasyon problemleri için çözümler üretir. Özellikle  $f(x)$  fonksiyonu problem çözümlenirken doğrusal ve doğrusal olmayan fonksiyonlar şeklinde olabilir.  $x$  değişkeninin değerine göre  $f(x)$  fonksiyonu minimize veya maksimize edilebilir [14, 133, 134].

Tezin bu bölümünde TAA'nın temel bileşenleri yeniden düzenlenerek KDPID parametrelerini optimize edebilecek bir yöntem önerilmiştir. TAA'nın temel bileşenleri ile problemin özgün şartları doğrultusunda ve algoritmanın dinamiklerinin esnekliği çerçevesinde yeni bir yöntem sunulmuştur. Örneğin TAA'nın temel bileşenleri olan komşuluk araması (neighborhood exploration), Tabu listesi (Tabu list), Tabu görev süresi (Tabu tenure), aspirasyon kriteri (aspiration criteria) ve durdurma kriteri (stopping criteria) gibi bileşenleri yeniden yorumlanarak TAA tabanlı bir yöntem önerilmiştir.

### **5.1.1 Tabu Arama Algoritmasının Temel Bileşenleri**

**5.1.1.1 Komşuluk (Neighbourhood):** Komşuluk mekanizması TAA'nın temel bileşenlerindedir. Komşuluk matrisi olası tüm aday çözümleri bünyesinde bulundurur. İlgili başlangıç değerlerine göre belli bir yöntem ile komşuluk matrisi oluşturulur. Komşuluğun oluşturulma yöntemi algoritmanın esnek yanlarında biridir. Araştırmacı kendi probleminin dinamiklerine göre komşuluk matrisini oluşturabilir [14,133,134,100].

**5.1.1.2 Tabu Listesi (Tabu List):** Tabu listesi algoritmanın başlangıcında boştur. Takip eden iterasyonlarda komşuluk matrisinin en iyi elemanları Tabu listesinde kaydedilir. Bu listede en iyi çözümler bulunur. Böylece algoritma test etmiş olduğu parametre vektör uzaylarına bir daha uğramamaktadır. Bu da algoritmanın lokal noktalara takılmasının önüne geçer.

**5.1.1.3 Tabu Koşulları (Tabu Conditions):** Komşuluk matrisindeki her eleman her iterasyonda bu koşula göre test edilir. Komşuluk matrisinin büyüklüğünü de bu koşul belirler. Bu koşullara göre tabu listesi güncellenir.

**5.1.1.4 Aspirasyon Kriteri (Aspiration Criterion):** Aspirasyon kriteri ise olası iyi çözümleri hafızada tutulmasını sağlayan bir kriterdir. İlgili çözüm tabu listesinin bir elemanı olsa bile bu kriter ile hafızada tutularak test esnasında problem için olası iyi sonuçların tutulmasını sağlar. Bu kriterin amacı önceden test edilmiş bölgelerin bir daha test edilmemesini sağlamaktır. Böylece algoritma optimizasyon işlemi süresince farklı parametre vektör uzaylarını tarayabilir.

**5.1.1.5 Durdurma Kriteri (Stopping Criterion):** Durdurma kriteri algoritmanın ne zaman durdurulacağını belirleyen bir kriterdir. Bu kriter probleme göre değişebilir. Ama genellikle algoritmanın başlangıcında belirlenen maksimum iterasyon sayısıdır.

**5.1.1.6 Tabu Görev Süresi (Tabu Tenure):** Tabu görev süresinde ilgili hareketin Tabu listesinde kalma süresini belirler. Bu süre, ilgili problemin yapısına göre değişiklik gösterir. Çözümlenen problemin yapısına göre problemin başlangıcında belirlenmesi gerekir.

## **5.2 Çok Parametrelili Fonksiyon İçin Tabu Arama Algoritmasının Düzenlenmesi**

$f(A, x)$  doğrusal veya doğrusal olmayan bir fonksiyon olsun  $A = [a_1, a_2, a_3, a_4, \dots, a_n]$ , parametre matrisi  $x = [x_1, x_2, x_3, x_4, \dots, x_n]$  ise bağımsız değişkenlerdir.  $f(A, x)$  fonksiyonu parametre matrisinin optimizasyonu ile minimizasyon ve maksimizasyon problemleri çözülür.

### **5.2.1 Önerilen Optimizasyon Yöntemi İçin Komşuluk Matrisi**

Tabu Arama Algoritmasında komşuluk matrisinin belirlenmesi, problemin yapısına göre farklılık gösterir.  $f(A, x)$  fonksiyonundaki her parametre  $a_i^n \pm s$  şeklinde iki yönlü olarak hareket ettirilir.  $s$  hareketin miktarını gösterir. İlgili parametreler hareket ettirilirken diğer parametreler sabit tutulur. Optimal çözümü

bulmak için her iterasyonda Denklem 5.2'deki konfigürasyona göre komşuluk matrisi oluşturulur.  $(N)$  olası tüm komşulukların bulunduğu matrisdir.  $N^n$  ise  $n$ . iterasyondaki komşuluk matrisini gösterir.

$$N^n = \begin{bmatrix} a_1 + s_1 & a_2 & a_3 & \cdots & \cdots & \cdots & \cdots & a_k \\ a_1 - s_1 & a_2 & a_3 & \vdots & \vdots & \vdots & \vdots & a_k \\ \vdots & a_2 + s_2 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & a_2 - s_2 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & a_k + s_k \\ a_1 & a_2 & a_3 & \cdots & \cdots & \cdots & \cdots & a_k - s_k \end{bmatrix} \quad (5.2)$$

Denklem 5.3'de komşuluk matrisindeki her satırda fonksiyonunun hesaplanan değeri tutulur.

$$S^n = \begin{bmatrix} f_1([a_1 + s_1, a_2 \cdots a_k], x) \\ f_2([a_1 + s_1, a_2 \cdots a_k], x) \\ \vdots \\ f_3([a_1 + s_1, a_2 \cdots a_k], x) \\ \vdots \\ f_k([a_1 + s_1, a_2 \cdots a_k], x) \end{bmatrix} \quad (5.3)$$

Minimizasyon problemleri için komşuluk matrisinin en iyi hareketi, maliyet matrisinin en küçük değerini veren satırıdır ve  $S_j^n = \min(S^n)$  denklemiyle hesaplanır. Maksimizasyon problemlerinde ise  $S_j^n = \max(S^n)$  denklemi kullanılır.

### 5.2.2 Önerilen Optimizasyon Algoritması İçin Tabu Listesi

Tabu listesi, minimizasyon veya maksimizasyon problemlerinde daha önceden test edilen çözümlerin bir daha test edilmesini önler.  $n^{\text{th}}$  iterasyonun en iyi

parametre konfigürasyonu  $A_j^n = [a_1^n a_2^n \dots a_j^n \pm s_j \dots a_k^n]$  olsun. Bu çözüm Tabu listesine kaydedilir. Bir sonraki iterasyonda bulunan bu en iyi değer yeni başlangıç değeri olarak kullanılır. Bu çalışma da Tabu listesinin etkin büyüklüğü  $2k$  olarak belirlenmiştir.  $k$  optimize edilen parametre sayısıdır.

### **5.2.3 Önerilen Optimizasyon Algoritması İçin Tabu Koşulu**

Önerilen algorithmada üretilen sonuçlar tabu koşuluna göre test edilir. Her iterasyondaki maliyet fonksiyonunun değeri tabu koşulu için çok önemlidir. Bundan dolayı Tabu listesinin güncellenmesinde çok kritik rol oynar.

### **5.2.4 Önerilen Optimizasyon Algoritması İçin Tabu Yaşam Süresi**

Algoritmanın etkin çalışabilmesi için Tabu listesine alınan hareketin kısıtlı bir sürede Tabu listesinde kalması gerekir. Önerilen algorithmada Tabu yaşam süresi  $2k$  olarak belirlenmiştir.  $k$  ise optimize edilen parametre sayısıdır. Bu değeri aştıktan sonra ilgili hareket tabu listesinde çıkarılır. Bu kriter Tabu listesinin aşırı büyümesini ve algoritmanın yavaşlamasını engellemektedir.

### **5.2.5 Önerilen Optimizasyon Algoritması İçin Aspirasyon Kriteri**

Önerilen algorithmada en iyi hareket en küçük maliyet değerine sahip olan harekettir. Her iyi hareket bu değerle karşılaştırılır. Bulunan değer aspirasyon kriterinden daha iyi ise bu değer güncellenir. Değil ise bu değer kriter olarak kalmaktadır. Böylece algorithmada bulunan en iyi değer her zaman muhafaza edilir.

### **5.2.6 Önerilen Optimizasyon Algoritması İçin Yeniden Başlatma Prosedürü**

Optimizasyon süresince maliyet fonksiyonu belli bir iterasyon süresince değişmiyorsa bunun iki anlamı vardır. Birincisi, algoritma optimal noktayı bulmuştur, ikincisi ise algoritma lokal bir noktaya takılmıştır. Bu durumda ilgili değerden başlamak kaydıyla algoritmanın komşuluk matrisi üretilirken kullanılan adım aralığı değiştirilerek testlere devam edilir. Belli bir iterasyon sonunda

maliyet fonksiyonunda iyileşme var ise algoritma bu noktadan süreci devam ettirir. Fakat iyileşme yok ise algoritmanın takıldığı yer ilgili optimizasyon süresi için global bir nokta olarak adlandırılır ve algoritma sonlandırılır. Bu süreç yeniden başlatma prosedürü olarak adlandırılır.

### 5.2.7 Önerilen Optimizasyon Algoritması için Durdurma Kriteri

Önerilen yöntemde durdurma kriteri maksimum iterasyon sayısı olarak belirlenmiştir. Bu çalışmada algoritma 100 iterasyon çalıştırılmıştır.

### 5.3 Önerilen Optimizasyon Algoritması

Klasik TAA tekli hareket ile optimizasyon işlemini yürütür. Önerilen yöntemde ise TAA çoklu hareket ile komşuluk matrisini oluşturur.  $A = [a_1, a_2, a_3, a_4, \dots, a_n]$ , parametre vektörünü  $S = [s_1, s_2, s_3, s_4, \dots, s_n]$  ise parametre vektöründeki her parametre için hareketleri gösteren vektördür. Önerilen yöntemde her parametre için farklı bir adım aralığı kullanılmıştır. Çünkü her parametre için aynı adım aralığında hareket yapmak algoritmayı lokal noktalara götürebilir. Fakat her parametre için farklı adım aralıklarının kullanılması algoritmanın lokalden kurtulup global çözüme ulaşmasını sağlar.

Bu yaklaşımla Denklem 5.2'de verilen komşuluk matrisi oluşturulmuştur. Denklem 5.3'de ise komşuluk matrisindeki her satır için hata fonksiyon değeri hesaplanıp ilgili matrise kaydedilmiştir.

Daha sonra oluşturulan her komşuluk matrisi tabu listesiyle karşılaştırılmıştır. Eğer komşuluktaki hareket tabu listesinde var ise bu hareket silinir. Böylelikle test edilen bölgeler bir daha test edilmez. İşlemin sonunda en iyi değer tabu listesine kaydedilir. Tabu her iterasyonda ilgili en iyi değeri aspirasyon kriteri ile karşılaştırır. Eğer o değerden daha iyi bir değer söz konusu ise ilgili kriter güncellenir.

Her hareket tabu listesine kaydedilirken Tabu listesinin çok fazla büyümemesi ve algoritmanın arama bölgesini kısıtlamaması için Tabu listesindeki hareketin kalma süresi  $2k$  olarak belirlenmiştir.  $k$  optimize edilen parametre sayısıdır.

Tabu listesi ( $T$ ) başlangıçta boştur. İterasyonlar ilerledikçe Denklem 5.4'e göre güncellenir.

$$T = \begin{bmatrix} a_{11n} & a_{21n} & \cdots & a_{k1n} \\ a_{21n} & a_{21n} & \cdots & a_{k1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{12kn} & a_{21n} & \cdots & a_{k2kn} \end{bmatrix} \quad (5.4)$$

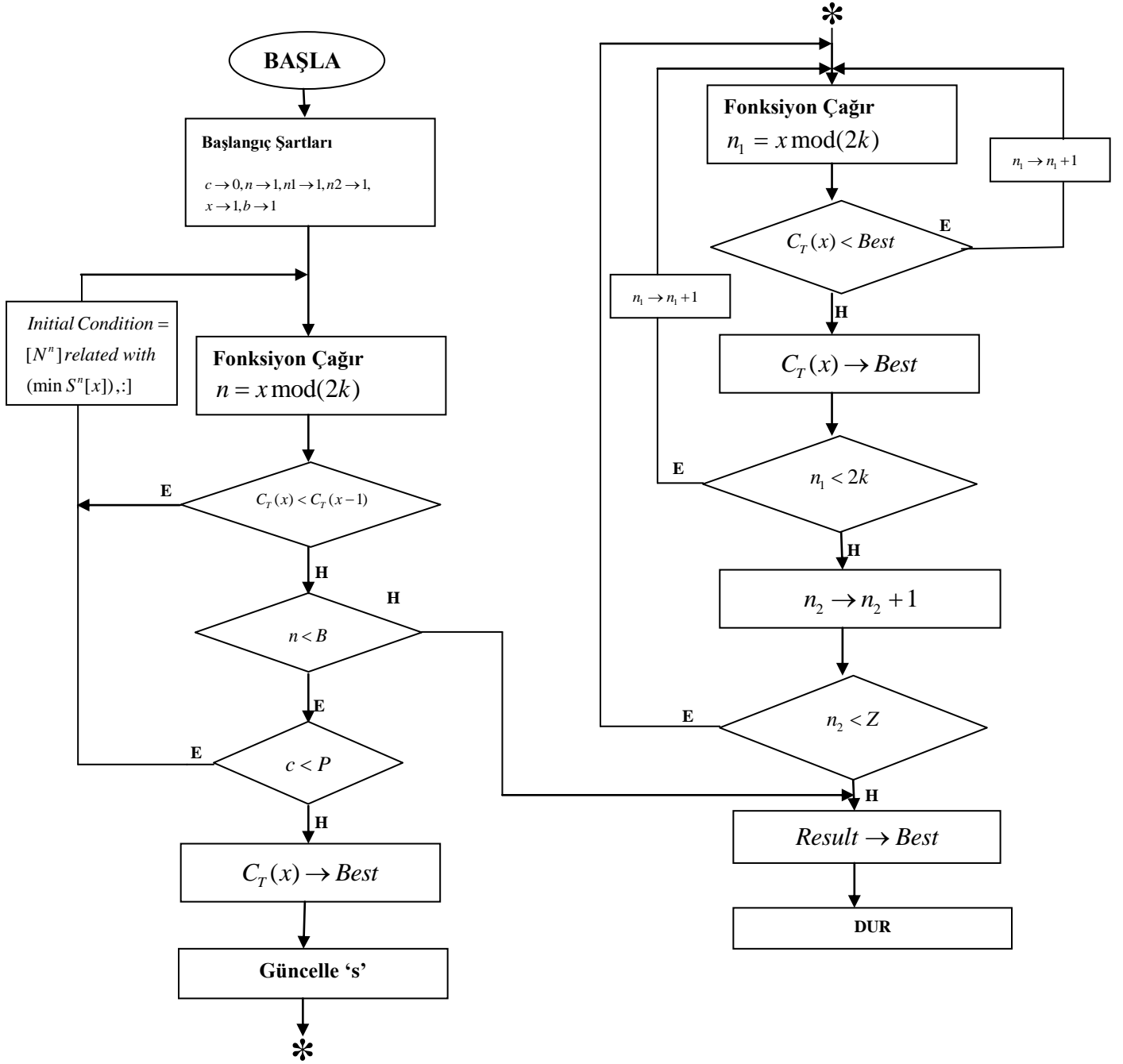
Denklemde  $i = 1, 2, \dots, k$  ,  $j = 1, 2, \dots, k$  ve  $n$  ise iterasyon sayısıdır. Hata fonksiyon değerleri Denklem 5.5'de bulunan ( $C_T$ ) matrisine kaydedilir;

$$C_T = \begin{bmatrix} f_1([a_{1n}, a_{2n}, \dots, a_{kn}], x) \\ f_2([a_{1n}, a_{2n}, \dots, a_{kn}], x) \\ \vdots \\ f_3([a_{1nk}, a_{2nk}, \dots, a_{knk}], x) \end{bmatrix} \quad (5.5)$$

Aday listesi yani komşuluk matrisi her iterasyonda  $S^n \cap C_T$  işlemine tabi tutulur. Yani aday listesiyle tabu listesinin ortak elemanı olup olmadığı test edilir. Eğer ortak elemanlar var ise bunlar aday listesinde çıkarılır. Böylece algoritmanın lokal noktalara takılması engellenmiş olur. Daha sonra komşuluk matrisinde kalan diğer elemanlar için hata fonksiyonu hesaplaması devam eder. Hata fonksiyonunun en düşük komşuluk matrisi kofigürasyonu indisi en iyi kofigürasyon olarak adlandırılır. Komşuluk matrisinde bu indise karşılık gelen parametre kofigürasyonunda en iyi parametre kofigürasyonu olarak tanımlanır. Bu değer öncelikle aspirasyon kriteri ile karşılaştırılır. Eğer bu değerden daha iyi ise güncellenmekte değilse değer değişikliğe uğramamaktadır. Daha sonra elde edilen bu değer tabu listesinde ömrünü tamamlayan değer yerine yazılacaktır. Aynı zamanda bu değer yeni iterasyonun başlangıç değeri olarak alınır.

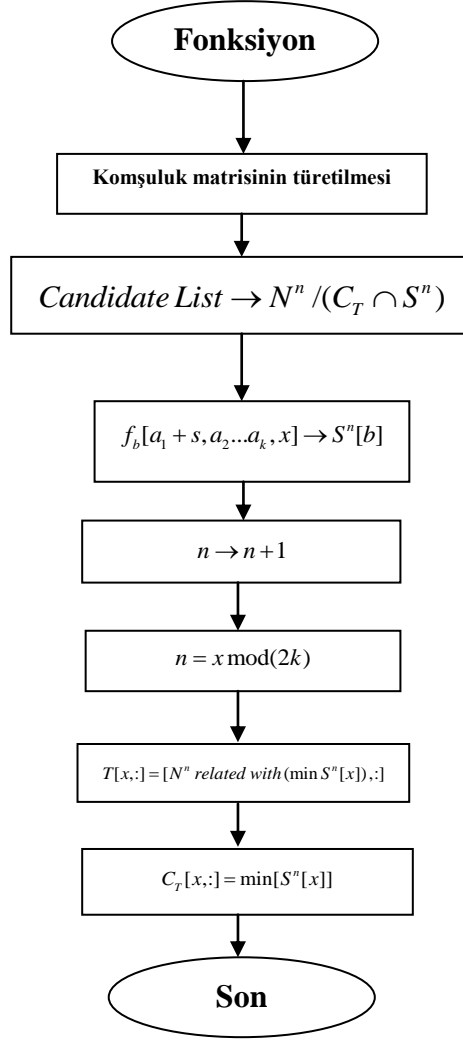
Algoritma arama işlemini sürdürürken belli bir iterasyonda hata fonksiyonu sabit ise algoritmanın lokal bir noktaya takıldığı veya global sonucu bulduğu anlamına gelir. Böyle durumda algoritma en son değerini alır ve algoritmanın başlangıcında

belirlediđi yeni hareket deęeri " $s$ " e gore algoritma kaldıđın yerden devam eder. Belli bir iterasyon sonra hata fonksiyonunda azalma meydana geliyorsa algoritma lokal deęerden kurtulmuştur. Eđer hata fonksiyonunda yine deęişim yoksa ilgili deęer algoritmanın global deęeri olarak kabul edilir ve bu deęer algoritmanın nihai sonucu olarak tanımlanarak optimizasyon sureci sonlandırılır. Algoritma bu işlemlere durdurma kriterini yani maksimum iterasyon sayısını sađđlayana kadar devam eder. Őekil 5.1'de onerilen TAA akış grafiđi gosterilmektedir. Őekil 5.2'de ise Őekil 5.1'de çağrılan ozel fonksiyonun akış grafiđi verilmektedir.



$c, n, n_1, x, b$  sistem sayıcıları  
 $B$ : birinci döngünün iterasyon sayısı  
 $Z$ : ikinci döngünün iterasyon sayısı  
 $P$ : Problemin tipine göre seçilen sezgisel değer

Şekil 5.1 Önerilen TAA Akış Şeması



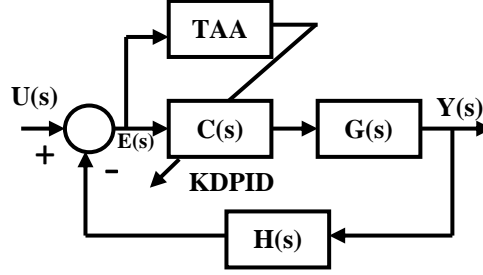
Şekil 5.2 Önerilen TAA'nın fonksiyon akış şeması

#### 5.4 Tabu Arama Algoritması Tabanlı Optimizasyon Yöntemiyle Kesir Dereceli Denetçi Tasarımı

Şekil 5.1 ve 5.2'de önerilen optimizasyon algoritması KDPID parametrelerini  $(k_p, k_i, k_d, \lambda, \mu)$  Denklem 5.6'da verilen hata fonksiyonunu minimize ederek belirlemektedir. Hata fonksiyonu sistemin kapalı döngü birim basamak cevabından birim basamak fonksiyonunun çıkarılmasıyla elde edilir [48].

$$J = \min \left\{ \sum (Y_{step}(s) - Y_p(s))^2 \right\} \quad (5.6)$$

Denklemdaki  $Y_{step}(s)$  birim basamak fonksiyonu,  $Y_p(s)$  ise birim basamak cevabıdır. Şekil 5.3’de ise optimizasyon işleminin blok şemasıdır.



Şekil 5.3 Sistem optimizasyon modeli

Algoritmanın başlangıcında KD denetçi parametrelerine  $(k_p, k_i, k_d, \lambda, \mu)$  ait farklı hareket miktarları  $s_1, s_2, s_3, s_4, s_5$  şeklinde tanımlanmıştır. Çünkü her parametre optimizasyon sürecini farklı etkilemektedir.

Algoritmanın bir sonraki adımı ise belirlenen hareket miktarlarına göre denetçi parametrelerinin  $(k_p, k_i, k_d, \lambda, \mu)$  komşuluk matrisinin oluşturulması ve komşuluk matrisindeki her satır için hata değerinin hesaplanıp kaydedilmesidir. KDPID parametrelerinin belirlenen hareket miktarlarına göre oluşturulan komşuluk matrisi Şekil 5.4’de ayrıntılarıyla gösterilmiştir. Komşuluk matrisinin her elemanı Tabu listesiyle karşılaştırılır. Tabu listesiyle aynı parametre konfigürasyonlarının olduğu hareketler komşuluk matrisinden çıkarılır. Böylece algoritma, denemiş olduğu iyi noktalara bir daha gitmeyerek farklı parametre vektör uzaylarını tarayabilir. Algoritmanın başlangıcında Tabu listesi boştur. Her iterasyonun başlangıç değeri tabu listesine kaydedilmekte ve her iterasyon için yeni başlangıç değeri olarak kullanılmaktadır. Aday listesindeki her parametre konfigürasyonu Şekil 5.5’de verilen simulink modeline uygulanarak sistemin birim basamak cevabı hesaplanır. Denklem 5.6’da verilen amaç fonksiyonuna göre her parametre konfigürasyonunun değeri bulunur. Şekil 5.4’de gösterildiği gibi hesaplanan hata

değerleri  $[S^n]$  matrisine kaydedilir. Her iterasyonda en küçük hata değerine sahip parametre konfigürasyonu en iyi değer olarak adlandırılır. Bu konfigürasyon Tabu listesine kaydedildiği gibi bir sonraki iterasyonun başlangıç değeri olmaktadır. Böylece denenmiş olan hareketlerin bir daha denenmesinin önüne geçilir. Tabu listesinin büyüklüğü algoritma açısından çok önem taşır. Çünkü bu liste gereğinden fazla büyüdüğünde, algoritma arama yapamaz hale gelir. Tabu listesinin büyüklüğünü belirleyen faktör ise Tabu yaşam miktarı kriteridir. Bu çalışmada 5 parametrenin optimizasyonu gerçekleştirildiğinden bir parametre konfigürasyonu 10 iterasyonun sonunda Tabu listesinden çıkar.

KDPID denetçinin optimizasyon süresince algoritmanın yeniden başlatılma prosedürü aşağıdaki gibidir.

*for n=1:iterasyon sayısı*

*for n1=1:2k*

*c=0*

*Bul ( $N^n$  related with  $\min[S^n]$ )*

*$T[n,:]=[N^n \text{ related with } \min[S^n]]$*

*$C_T[n,:]=\min[S^n]$*

*If ( $(C_T[n-1,:] > C_T[n,:]) \parallel (C_T[n-1,:]=C_T[n,:])$ )*

*c=c+1*

*end*

*If c>P*

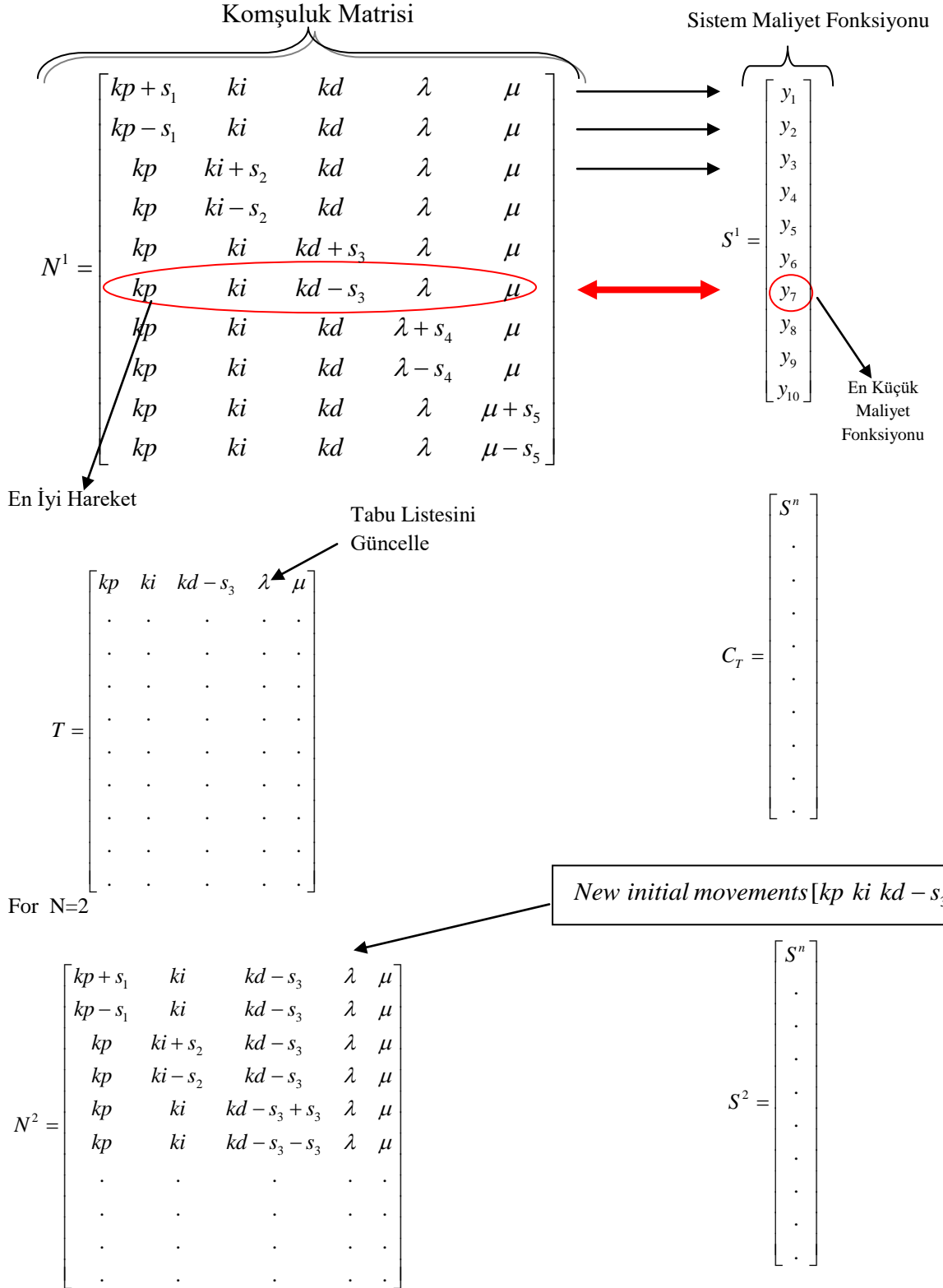
*Yeniden Başlama Prosedürü Uygula*

*s hareket miktarı değerini güncelle*

*end*

*end*

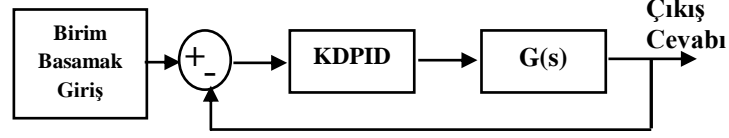
*end*



**Şekil 5.4** İlk iki iterasyon için TAA'nın matris konfigürasyonu

Burada,  $k$  –KDPID denetçi parametre sayısı,  $n$  – maksimum iterasyon sayısı,  $n1$  – Tabu listesini güncelleyen iterasyon sayısı,  $c$  – sayıcı miktarı,  $T$  – Tabu

listesi,  $S^n$  – Maliyet fonksiyonun değerlerinin listesi,  $C_T$  – Tabu listesine konulan maliyet fonksiyonlarının listesi and  $P$  – Problem tipine göre seçilen sezgisel sayı.



Şekil 5.5 Matlab Simulinkte Kullanılan Simülasyon Modeli

## 5.5 Simülasyon Sonuçları

Bu bölümde 3 farklı matematiksel model için TAA tabanlı optimizasyon algoritmasıyla KDPID denetçiler tasarlanmıştır. Elde edilen sonuçlar aynı matematiksel modeller için TAA ile elde edilmiş klasik PID denetçilerin birim basamak cevapları ile karşılaştırılmıştır. İlgili matematiksel modeller [100]'den alınmıştır. Daha sonra önerilmiş olan algoritma literatürde [30]'da önerilmiş olan geliştirilmiş diferansiyel gelişim algoritmasıyla karşılaştırılmıştır.

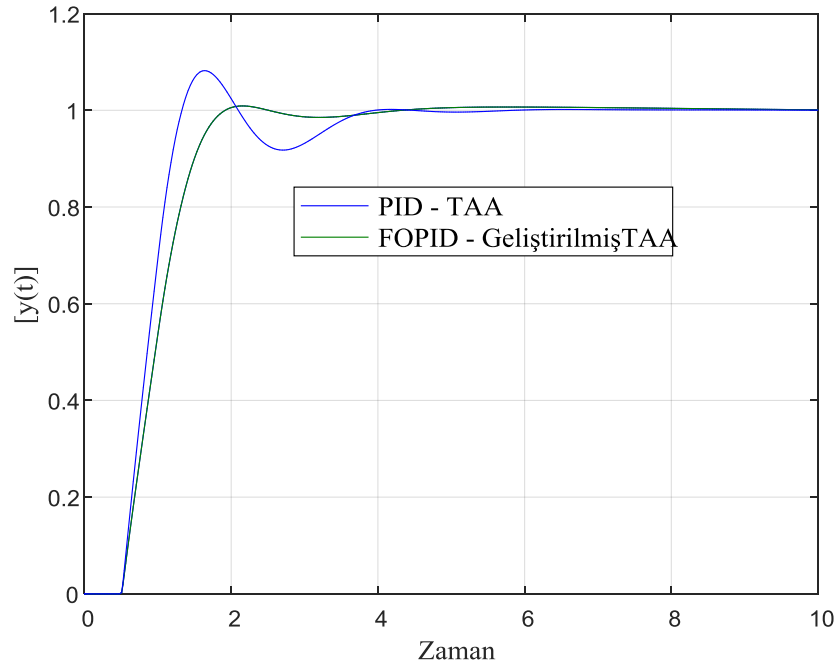
**5.5.1. Simülasyon Örnek 1:** Denklem 5.7'de verilmiş olan sistemi göz önünde bulunduralım [100];

$$G_1(s) = \frac{e^{-0.5s}}{(s+1)^2} \quad (5.7)$$

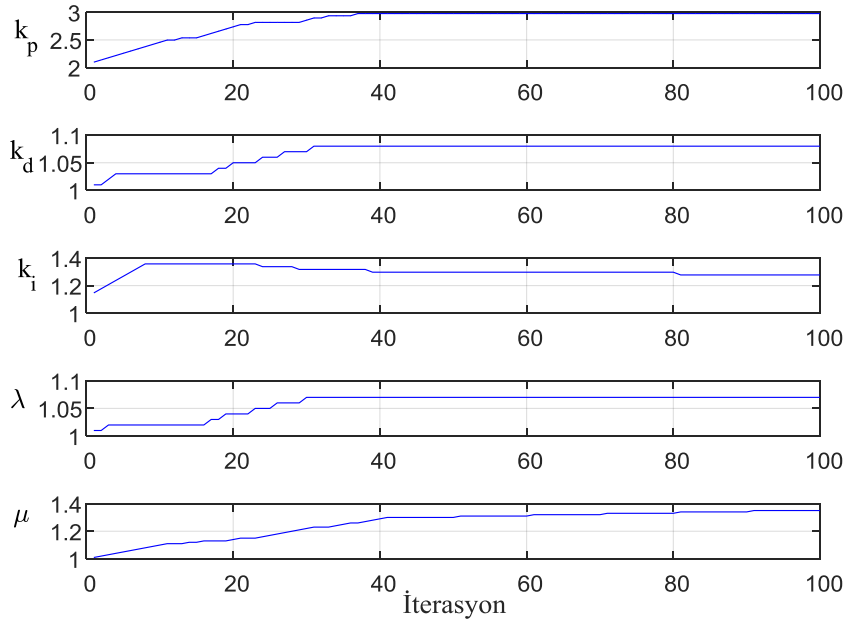
Denklem 5.7'de verilen matematiksel model ve KDPID yapısı Şekil 5.5'de verilmiş olan simülasyon modeline uygulanmıştır. Daha önceki bölümlerde de bahsedildiği gibi KDPID Valerio'nun önermiş olduğu ninteger MATLAB simulink araç kutusuyla gerçekleştirilmiştir [64]. Şekil 5.5'de verilen simulink modeli MATLAB ortamında oluşturulan kodun içine gömülerek her iterasyonda sistem cevabını tespit etmek için rekürsif olarak çağrılmıştır. Simülasyon çalışması MATLAB simulink ortamında MATLAB ode45 çözümleyicisi

arayıcılığı ile yapılmaktadır. Çizelge 5.1’de önerilen optimizasyon algortiması ile elde edilen KD denetçi katsayıları bulunmaktadır. Şekil 5.6’da ilgili matematiksel model için önerilen yöntemle elde edilen KDPID ile [100]’de verilen klasik PID’nin birim basamak cevaplarının karşılaştırmaları önerilen yöntemin avantajını gösterir.

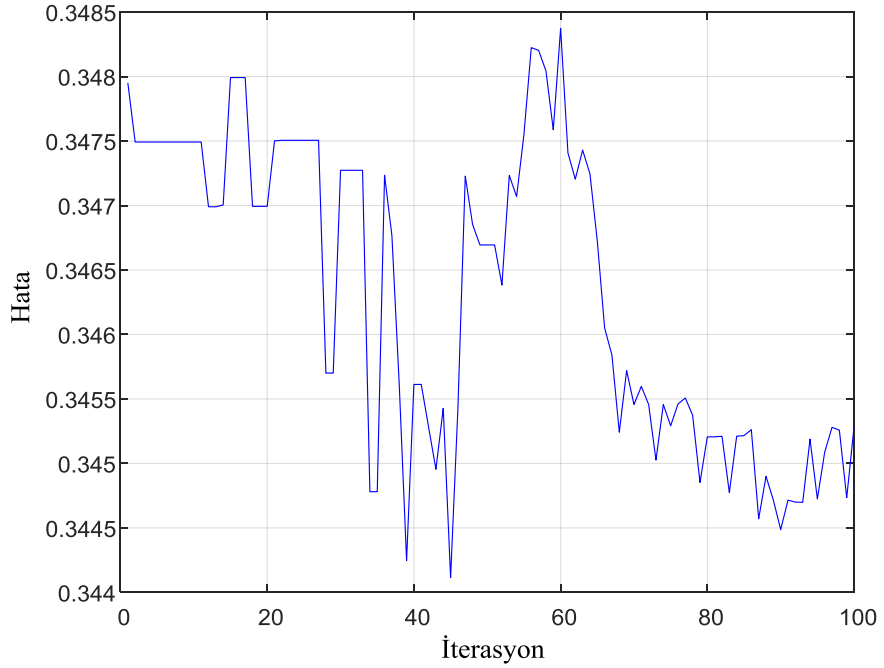
Şekil 5.7’de optimizasyon süresince KDPID’nin parametre değişimleri, Şekil 5.8’de ise optimizasyon süresince hata fonksiyonunun değişimi görülmektedir. Algoritma 100 iterasyon çalıştırılmıştır. Şekillerden görüldüğü gibi 70 iterasyonda optimal sonuca ulaşabilmektedir.



**Şekil 5.6** Denklem 5.7’de verilen sistem için PID ve KDPID’nin birim basamak cevapları



**Şekil 5.7** Optimizasyon süresince KDPID parametrelerinin değişimi

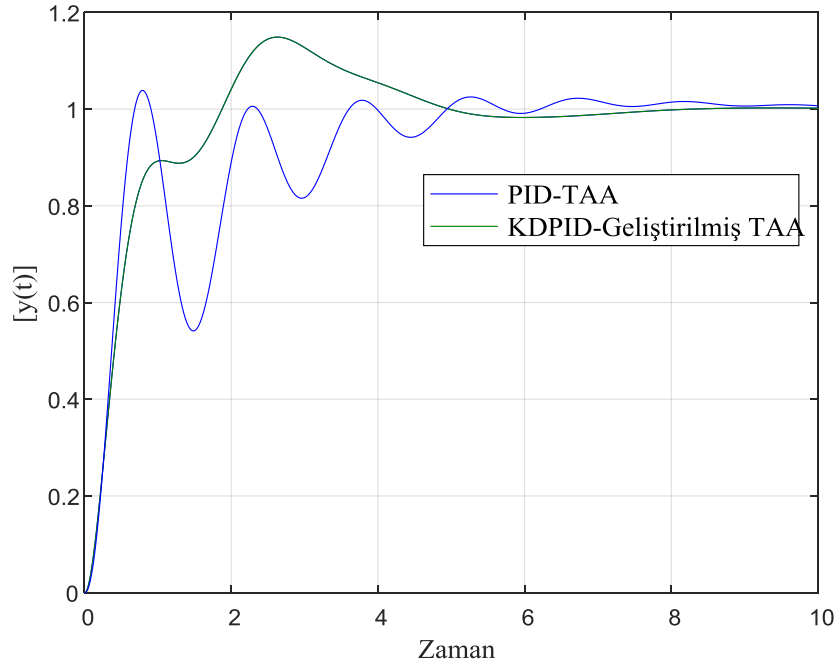


**Şekil 5.8** Optimizasyon süresince hata fonksiyonunun değişimi

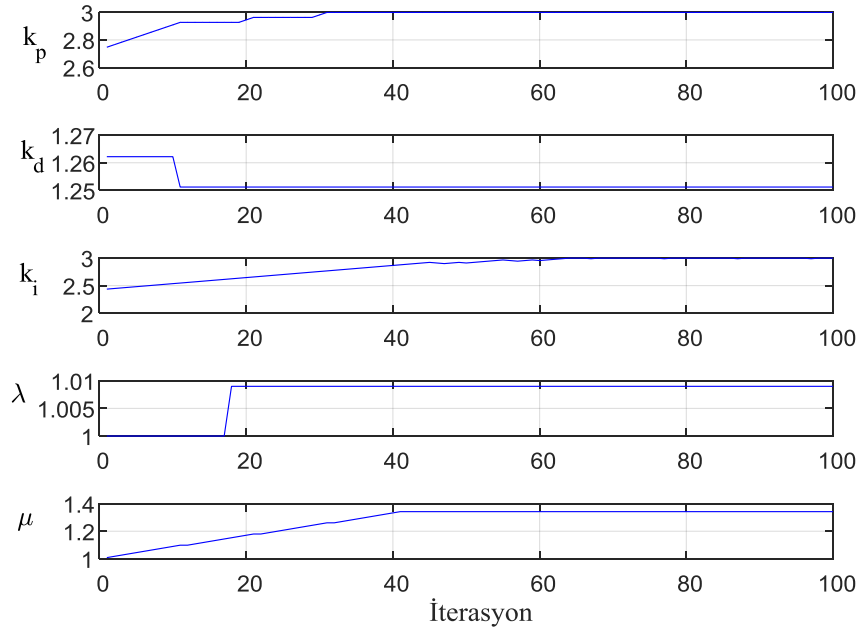
**5.5.2. Simülasyon Örnek 2:** Bu örnekte KDPID Denklem 5.8’de verilen matematiksel model için elde edilmiştir[100]:

$$G_2(s) = \frac{4.228}{(s + 0.5)(s^2 + 1.64s + 8.456)} \quad (5.8)$$

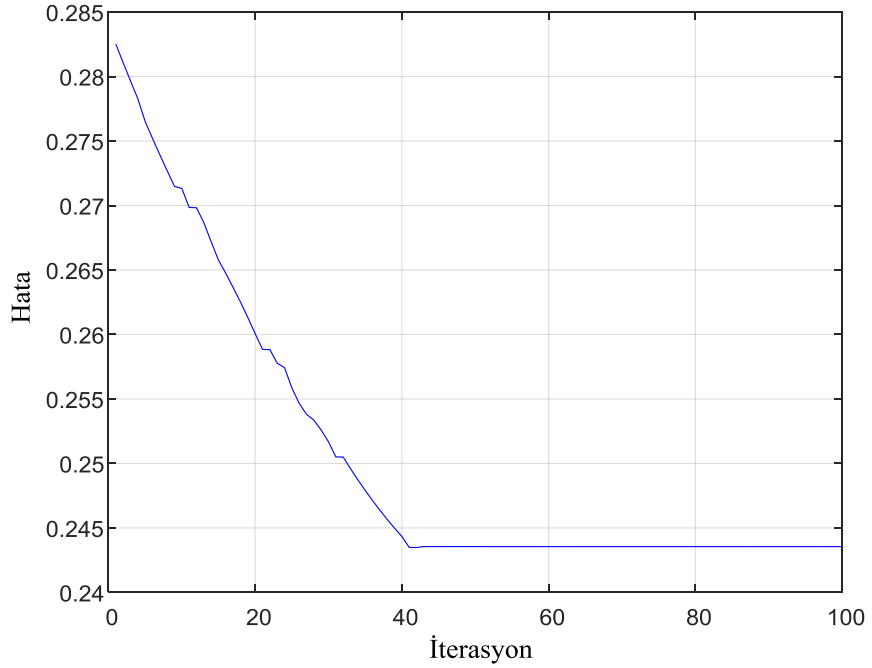
İlk örnekteki prosedürün aynısı bu sistem içinde uygulanmıştır. Çizelge 5.1’de bu sistem için üretilen KDPID parametrelerinin değerleri bulunmaktadır. Şekil 5.9’da KDPID’nin klasik PID’ye olan üstünlüğü, Şekil 5.10’da optimizasyon esnasında parametrelerin değişimi, Şekil 5.11’de ise optimizasyon süresince hata fonksiyonunun değişimi gösterilmektedir. Optimizasyon algoritması 100 iterasyon çalıştırılmıştır. Fakat şekillerden görüleceği gibi algoritma 40. iterasyonda ilgili optimizasyon döngüsünde global değere yakınsamaktadır.



**Şekil 5.9** Denklem 5.8’de verilen sistem için PID ve KDPID’nin birim basamak cevapları



**Şekil 5.10** Optimizasyon süresince KDPID parametrelerinin değişimi

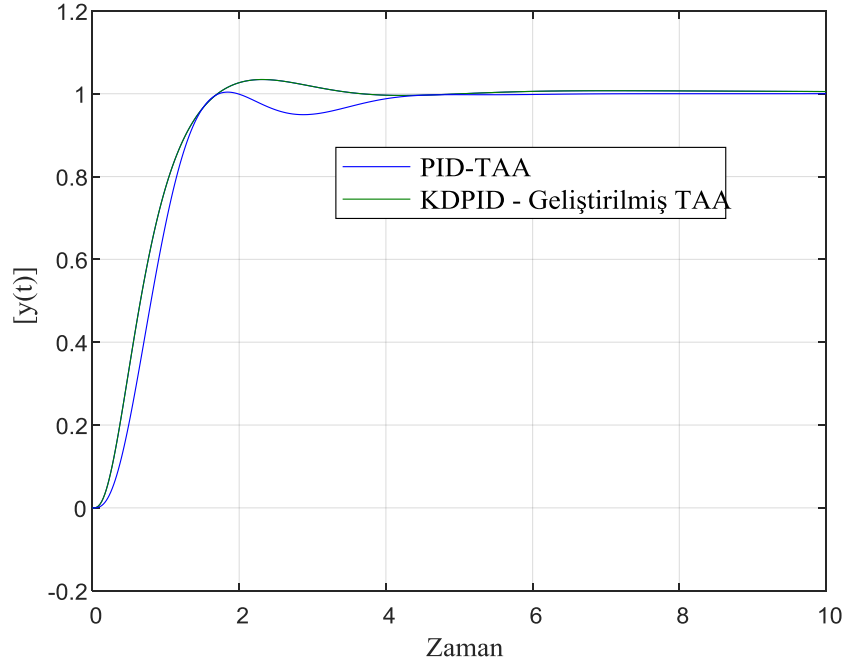


**Şekil 5.11** Optimizasyon süresince hata fonksiyonunun değişimi

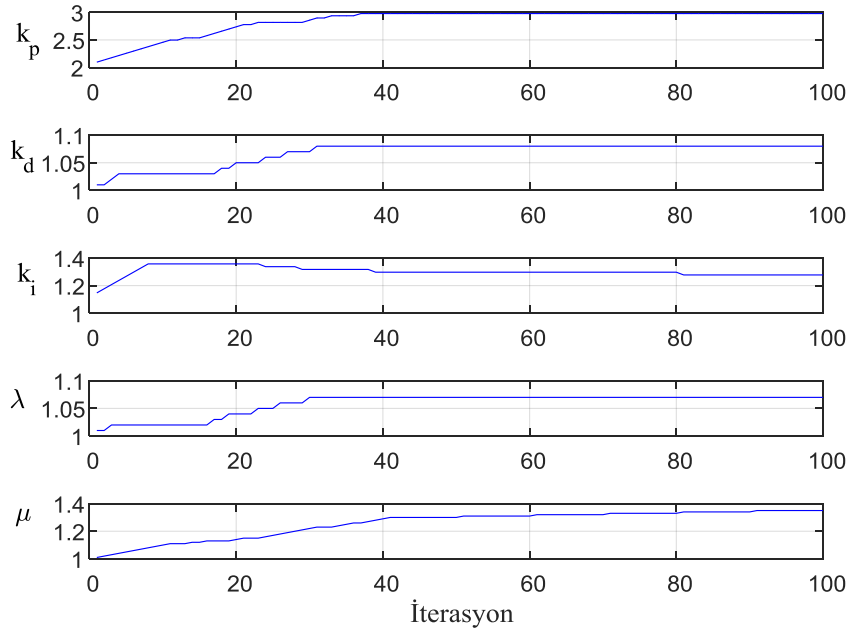
**5.5.3. Simülasyon Örnek 3:** Bu örnekte [100]'den alınan 4. dereceden model kullanılmıştır.

$$G_3(s) = \frac{27}{(s+1)(s+3)^3} \quad (5.9)$$

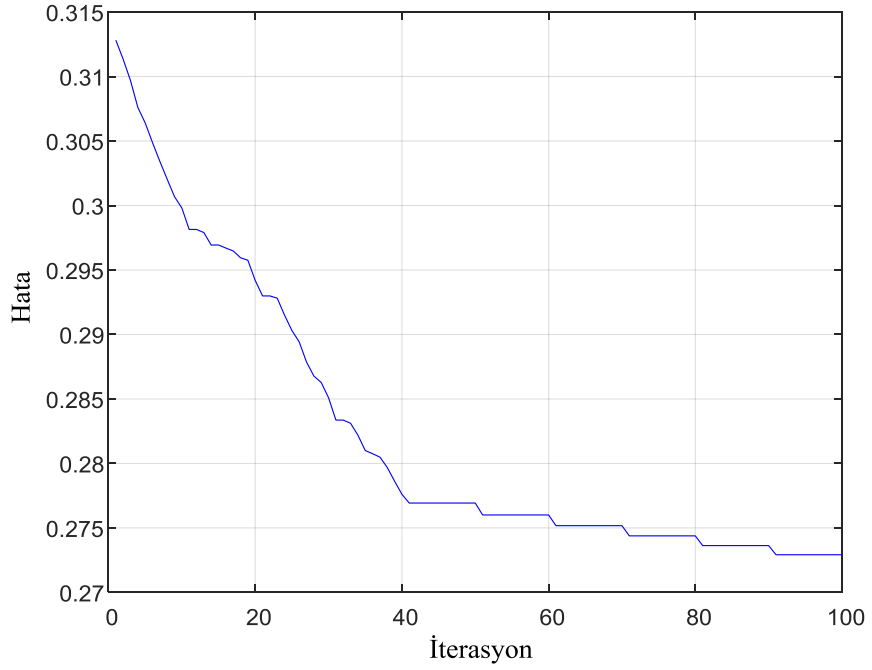
Önerilen optimizasyon yöntemi ile elde edilen KDPID katsayıları Çizelge 5.1'de verilmektedir. Önerilen yöntemin avantajları Şekil 5.12, 5.13 ve 5.14'de sunulmuştur. Şekil 5.12'de birim basamak cevapları karşılaştırılmaktadır. Şekil 5.13'de optimizasyon süresince parametrelerin değişimleri, Şekil 5.14'de ise optimizasyon süresince hata fonksiyonun değişimi gösterilmiştir. Bu örnek için de algoritma 100 iterasyon çalıştırılmıştır.



**Şekil 5.12** Denklem 5.9'da verilen sistem için PID ve KDPID'nin birim basamak cevapları



**Şekil 5.13** Optimizasyon süresince KDPID parametrelerinin değişimi



**Şekil 5.14.** Optimizasyon süresince hata fonksiyonunun değişimi

**Çizelge 5.1** Önerilen algoritmaya göre elde edilen KDPID parametreleri

	<i>Model 1 (Denk.5. 7)</i>	<i>Model 2 (Denk.5. 8)</i>	<i>Model 3 (Denk.5. 9)</i>
<i>kp</i>	2.1698	2.9890	2.9370
<i>kd</i>	1.0470	1.2412	1.080
<i>ki</i>	0.9021	2.5069	1.2770
$\lambda$	1.0740	1.0090	1.0700
$\mu$	1.0452	1.4930	1.3400

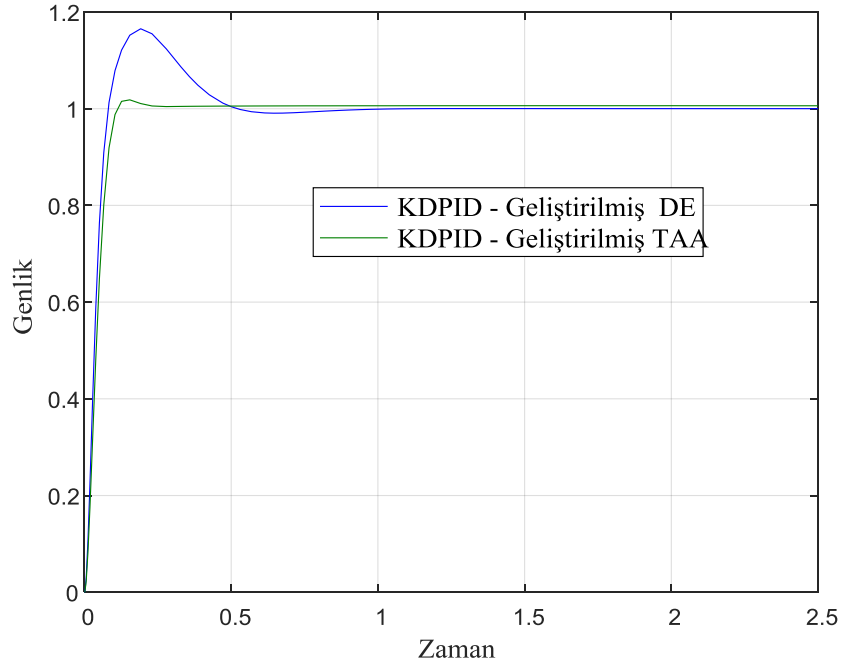
**5.5.4. Simülasyon Örnek 4 :** Önceki 3 örnekte literatürde TAA ile elde edilmiş klasik PID ile önerilen yöntem ile elde edilen KDPID karşılaştırılmıştır. Bu örnekte ise farklı algoritmayla elde edilen KDPID ile önerilen yöntemle elde edilen KDPID karşılaştırılacaktır. Denklem 5.10’da verilmiş olan matematiksel model [30]’dan alınmıştır. [30]’da bu sistem için geliştirilmiş diferansiyel gelişim optimizasyon algoritmasıyla kesir dereceli denetçi tasarlanmıştır. Önerilen yöntemin üstünlüğünü göstermek için aynı sistem TAA tabanlı optimizasyon yöntemiyle yeniden optimize edilerek KDPID elde edilmiştir. Sonuçlar Çizelge 5.2 de verilmektedir.

$$G_4(s) = \frac{400}{s^2 + 50s} \quad (5.10)$$

Şekil 5.15’de görüldüğü gibi önerilen yöntem literatürde ki yöntemden daha iyi birim basamak cevabı üretmiştir.

**Çizelge 5.2** Önerilen TAA ve geliştirilmiş diferansiyel gelişim algoritmasına (GDGA) göre elde edilen KDPID denetçi katsayıları

	<i>kp</i>	<i>kd</i>	<i>ki</i>	$\lambda$	$\mu$
GDGA ile KDPID	0.349	1.009	19.287	0.949	0.322
Önerilen TAA ile KDPID	2.94	0.011	0.4141	1.2401	1.002



**Şekil 5.15** Denklem 5.10’da verilen sistem için TAA ile üretilmiş KDPID ile Geliştirilmiş DE ile üretilmiş KDPID’nin birim basamak cevapları

Optimizasyon süresince algoritma yine 100 iterasyon çalıştırılmaktadır. 40. iterasyonda algoritma kendi optimizasyon süreci içerisinde global değere yakınsamıştır.

Takip eden diğer bölümde ise Yapay Fizik Algoritması kullanılarak kesir dereceli ve tam sayı dereceli sistemlere denetçi tasarlayabilecek bir optimizasyon algoritması önerilecektir. Algoritmanın tüm kurallarının denetçi tasarlayabilecek yeteneğe ulaştırılması amaçlanmaktadır.

## 6. DENETÇİ PARAMETRELERİNİN YAPAY FİZİK OPTİMİZASYON ALGORİTMASI İLE BULUNMASI

Bu bölümde literatürde olan fakat denetçi tasarımında kullanılmamış bir yöntem olan Yapay Fizik Optimizasyon (YFO) Algoritmasıyla literatürden alınmış sistemlere KD ve tamsayı dereceli denetçiler tasarlanmıştır. Algoritmaya öncelikle denetçi tasarımı yapabilecek yetenek kazandırılmıştır. Bunun için algoritmanın tüm bileşenleri denetçi tasarımı yapabilecek şekilde değiştirilmiştir. Ayrıca algoritmada başlangıç değer problemini çözmek için Temel optimizasyon algoritmasından (TOA) (Base Optimization Algorithm- BaOA) faydalanılarak aday çözümlerin oluşturulması sağlanmıştır [146].

Genellikle mühendislik problemlerinin çözümünde nümerik optimizasyon algoritmaları tercih edilir. Özellikle meta-sezgisel, akıllı hesaplama (computational intelligent), zeki sürü gibi stokastik yöntemler, problemler için global veya global çözüme en yakın çözümler üretebilir. Stokastik yöntemler global çözüm aramasında uygun zaman diliminde farklı parametre vektör uzaylarını arama yeteneği nedeniyle daha uygundur. Bunlara ek olarak stokastik yöntemler analitik yöntemlerin ihtiyaç duyduğu sürekli maliyet fonksiyonu ve değişkenlere ihtiyaç duymaz. Literatürde farklı optimizasyon problemlerini çözen birçok stokastik algoritma bulunmaktadır. Örneğin literatürde bilinen bazı stokastik yöntemler tezin önceki bölümlerinde gösterilmiştir [2,13-21]. Literatürde bulunan birçok stokastik optimizasyon algoritması doğada bulunan fiziksel kurallardan esinlenerek oluşturulmuştur. Bu tarz yeni optimizasyon algoritmalarından biri olan Fizik tabanlı algoritmalar fizik biliminin temel kurallarında faydalanılarak oluşturulmuştur. Fizik bilimi maddenin incelenmesini, maddenin uzaydaki hareketini, zamanını, enerjisini ve kuvvetini içerir [21, 147]. Uygulamaya bağlı olarak, algoritmik süreçteki parçacıklar buldukları çevre içerisinde belirli bir problemi çözmeye doğrultusunda hareket eden robotlar gibi davranır [146-150].

Yapay Fizik Optimizasyon (YFO) algoritmasında maddenin fiziksel davranışı olan “kütle (mass)”, “kuvvet (force)” ve “hız (velocity)” kavramlarından yararlanır. Algoritmadaki her parçacık fiziksel bir birey olarak kabul edilir. Çekici itici kuralı

(attractive-repulsive rule) düşük maliyet fonksiyonu değerine sahip bölgeleri bulmada kullanılır.

Diğer bir önemli durum ise stokastik optimizasyon algorimaları optimizasyon süresi başladığında aday çözümleri bünyesinde bulunduran sürüler oluşturur. Bu sürüler belli bir yöntemle göre genellikle doğadan esinlenerek elde edilir. Bu sayede optimal çözüme ulaşılır. Sürülerin doğal özelliklerinin avantajları, kendi kendini düzenleme, dayanıklılık, kendini onarma, ölçeklenebilirlik, gürültüye karşı tolerans ve yerel iletişim olarak ortaya çıkar [150].

Modern bilgisayar biliminde sürü literatürü zeki sürüler (swarm intelligence), davranış tabanlı (behavior-based), kural tabanlı (rule-based), kontrol teorisi (control-theoretic) ve fizik tabanlı (physics-based techniques) teknikler gibi alt grublara ayrılabilir [150]. Sürü tabanlı teknikler doğal yapılarından dolayı görev paylaşımında ve görevin gerçekleştirilmesinde ciddi başarı sağlar [150, 151]. Davranış Tabanlı (Behavior-based) ve kural tabanlı (rule-based) sürü optimizasyon yöntemleri belirli bir sürü davranışını sergilemede daha başarılı olabilir [152-154]. Bundan dolayı sürünün oluşturulma tekniği optimizasyon probleminin çözümünde çok kritik bir öneme sahiptir. Basit ve zeki sürü üretme yönteminin kullanıldığı bir optimizasyon algoritmasında, algoritma ilgili problemi hızlı bir şekilde daha az matematiksel hesaplamayla çözebilir. Örneğin Temel Optimizasyon algoritması basit aritmetik işlemler yaparak sürü oluşturur [1].

Tezin bu bölümünde çoklu parametre optimizasyonu için Yapay fizik optimizasyon algoritmasıyla Temel optimizasyon algoritması birleştirilerek literatüre etkin bir optimizasyon yöntemi önerilmektedir. TOA Salem tarafında önerilmiş hesapsal zeki (computational intelligence) bir yöntemdir [1]. Bu algoritmanın en önemli avantajı temel matematiksel operatörleri (-,+,\*./) kullanarak aday çözümleri oluşturmasıdır.

Zeki bir sürü optimizasyon algoritması olan YFO algoritması ise “physicomimetics” in avantajlarından faydalanarak global çözüm bulmada ciddi üstünlüğü bulunan bir algoritmadır [155-158].

Bu çalışmada yapay fizik algoritması ile temel optimizasyon algoritmasının üstün olduğu tarafların birleştirilmesiyle Gelistirilmiş Yapay Fizik Optimizasyon

(GYFO) algoritması (Modified Artificial Physic Optimization (MAPO)) çoklu parametre optimizasyonu için önerilmektedir.

Algoritma başlangıçta Temel optimizasyon algoritmasından faydalanarak ve temel matematiksel operatörler kullanarak sürü matrisini oluşturur. Daha sonra sürü matrisinde her satırda bulunan parametre konfigürasyonu için Yapay Fizik Algoritması kuralları kullanarak ilgili hareketlerin kütle, hız ve kuvvet değerleri hesaplanır. Algoritma başka bir iterasyona başlarken ilgili iterasyondaki kütle, hız ve kuvvet değerleri kullanılarak hesaplanan yeni parametre konfigürasyonunu alarak hareket eder.

Bu çalışmada önerilen geliştirilmiş yapay fizik optimizasyon algoritmasıyla, literatürden alınmış sistemlere KD ve tam sayı dereceli denetçiler tasarlanmıştır. Bilindiği gibi denetçi parametrelerinin belirlenmesi bilim dünyası açısından son derece önemli ve güncel bir konudur. Birçok çalışmada denetçi parametreleri analitik yöntemler [106, 132] ve optimizasyon yöntemleri [82, 83, 125-128] kullanılarak belirlenir. GYFO algoritması ile tam sayı dereceli ve KD sistem modellerine KD ve tam sayı dereceli denetçiler tasarlanmıştır. Önerilen algoritmanın üstünlüğü literatürde aynı sistemler için türetilmiş denetçi parametrelerinin birim basamak cevaplarıyla karşılaştırılmıştır. Böylece önerilen algoritmanın avantajı gösterilmiştir.

## **6.1 Yapay Fizik Optimizasyon Algoritması ve Temel Optimizasyon Algoritması**

Tezin bu bölümünde Yapay Fizik ve Temel Optimizasyon algoritmalarının temel yapılarından sırasıyla bahsedilmektedir.

### **6.1.1 Yapay Fizik Optimizasyon Algoritması**

Yapay Fizik optimizasyon algoritması “Physicomimetics” yaklaşımından faydalanarak global optimizasyon problemlerinin çözebilen zeki bir optimizasyon yöntemidir [146, 147]. Yapay fizik optimizasyon algoritmasındaki her çözüm bir birey olarak kabul edilir. Bu bireyler kütle, konum ve hız değerlerine sahiptir.

YFO algoritmasının genel kuralları diğer evrimsel gelişim algoritması kurallarına physicomimetics kurallarının etki ettirilmesiyle oluşturulur. YFO algoritması, Physicomimetics veya Yapay fizik (Artificial Physics-AP) yaklaşımını kullanır. Yapay Fizik çoklu robot sürülerinin kontrolünde kullanılan bir metafordur [146, 148-150]. Bireysel robotlar, oluşturulan yapay fizik çevresinde, çevresel parametreleri algılayıp algoritmik olarak cevap üretir. Bu konudaki temel yaklaşım şudur: Temel fizik kurallarına göre oluşturulan her sistem modern fizikte ve mühendislikte kullanılan yöntemler ile ampirik, analitik ve teorik olarak çözümlenebilir.

Yapay fiziğin temel yaklaşımına göre robotlar konuma, kütle ve momentuma sahip fiziksel parçacıklar olarak değerlendirilir. Physicomimetic içerisinde oluşturulmuş sanal kuvvetler aynen gerçek kütlelerin harici olarak uygulanan bir kuvvete vermiş olduğu tepki hareketi gibi parçacıkların hareket etmesini sağlar. Yapay fiziğin bu itici gücü, yer çekimi kuvveti olarak tanımlanır [155-158]. Benzer olarak Spear ve arkadaşları Physicomimetics kurallarını [156]'da sunmaktadır.

Sanal fiziksel kuvvetler çoklu robot sistemlerini istenen bir konfigürasyona ve duruma sürükler. İstenen konfigürasyon genel sistemin potansiyel enerjisini minimize eden konfigürasyondur. Yani sistem bir moleküler dinamik gibi davranır.

En yaygın kuvvet kuralı Denklem 6.1'de dir.

$$F = G \frac{m_i m_j}{r^\xi} \quad (6.1)$$

Denklemdaki  $G$  yer çekimi sabitidir (gravitational constant),  $F$ ,  $i$  ve  $j$  parçacıkları arasındaki kuvvet,  $r$  iki parçacık arasındaki mesafe  $\xi$  kullanıcının tanımladığı güçtür [155-158].

Yapay fiziğin genel kurallarında robotlar iki veya üç yöne sahip fiziksel parçacıklar olarak davranır. Her bir parçacık kütle "mass" ( $m$ ), konuma "position" ( $\mathbf{x} = (x_1, x_2, x_3)$ ) ve hıza "velocity" ( $\mathbf{v} = (v_1, v_2, v_3)$ ) sahiptir.

Parçacıkların sürekli zamanlı davranışlarına, ayırık zamanlı sapmalarla yakınsanır. Her zaman adımında her bir parçacığın sapması mevcut durumda bulunan hızı temel almaktadır [155-158]. Örneğin  $\Delta x = v\Delta t$ , hızın değişimi ( $\Delta v$ ) parçacığın kuvvetiyle kontrol edilir  $\Delta v = F\Delta t / m$ . İfadede ki  $F$  parçacığın kuvvetini,  $F_{\max}$  ve  $v_{\max}$  sırasıyla ilgili parçacığın maksimum kuvvet ve hızını gösterir.

Yapay fizik yaklaşımı, robot oluşumu, engel önleme ve kapsama görevleri gibi dağıtılmış robot sürülerinin kontrolünde kullanılır [2, 147-150].

### 6.1.2 Temel Optimizasyon Algoritması

Temel optimizasyon algoritması (TOA) temel matematiksel operatörleri kullanılıp parçacıkların hareketini sağlayarak optimum çözüm üretebilen bir yöntemdir. [1] Temel optimizasyon algoritmasının çalışma prensibi kısaca aşağıda sunulmaktadır.

i. Algoritmanın başlangıcında rassal olarak ( $S_1, S_2, \dots, S_n$ ) aday noktaları  $d_i$  yönü için oluşturulur ( $i = 1, 2, \dots, n$ ).

ii. Oluşturulan her parametre konfigürasyonunun maliyet fonksiyonu hesaplanır. Daha sonra her bir aday çözüm, algoritmanın başlangıcında belirlenen hareket miktarı kadar temel matematiksel operatörler (+, -, \*, /) kullanılarak hareket ettirilir.

Elde edilen yeni aday kümesinin hepsinin maliyet fonksiyonu hesaplanarak en küçük değere sahip konfigürasyon seçilip yeni iterasyonun başlangıç değeri olarak alınır.

### 6.2. Geliştirilmiş Yapay Fizik Optimizasyon Algoritmasıyla Çoklu Parametrelerin Optimizasyonu

$f(\mathbf{p})$   $\mathbf{p} = [p_1, p_2, p_3, \dots, p_n]$  bağımsız değişkenleriyle oluşturulan doğrusal veya doğrusal olmayan bir fonksiyon olsun.  $f(\mathbf{p})$  fonksiyonunun parametrelerinin optimizasyonu, minimizasyon veya maksimizasyon problemi olabilir. Denklem 6.2'de verilmiş olan kriter çoklu parametre optimizasyonunda kullanılan minimizasyon kriteri olarak kabul edilir.

$$\min \{f(p_1, p_2, \dots, p_n) : p \in \Omega \subset \mathbb{R}^n\}, f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^l \quad (6.2)$$

Denklemden  $\Omega = \{p \mid p_k^{\min} \leq p_k \leq p_k^{\max}, k = 1, \dots, n\}$  dir.

Denklemin 6.2'de verilen problem  $n$  yönlü bir optimizasyon problemidir ve  $p_k^{\min}$ , her yönün alt sınırını  $p_k^{\max}$ , her yönün üst sınırını karakteriz eder.  $f(p_{k\dots n})$  ise minimize edilen fonksiyonun işaretçisidir. Bu çalışmada önerilen Geliştirilmiş Yapay Fizik Optimizasyon Algoritması 3 adımdan oluşmaktadır. Bunlar, i) Başlangıç (Initialization), ii) Kuvvetin hesaplanması (Force Calculation), iii) Hareket (Motion) [157,158]. Başlangıç adımı olarak Temel optimizasyon algoritmasına göre rassal adım aralığı ile başlangıç sürüsü aşağıda belirtilen yönteme göre Denklem 6.3 kullanılarak oluşturulur.

for i=1:1:n

Swarm Matrix (i,:)=  $[p_i + \delta_i \quad p_i - \delta_i \quad p_i * \delta_i \quad p_i / \delta_i]$

end

$$f(\mathbf{p}) = [p_i + \delta_i \quad p_i - \delta_i \quad p_i * \delta_i \quad p_i / \delta_i] \quad i = 1..n \quad (6.3)$$

GYFO algoritması sürü matrisindeki her satırın maliyet fonksiyonunu yani hata değerini hesaplayarak konum vektörünü elde eder.  $i^{th}$  yani ilgili iterasyonda konum vektöründeki en küçük hata değerine sahip parçacık  $p_{best}$  en büyük hata değerine sahip parçacık ise  $p_{worst}$  olarak adlandırılır. Önerilen algoritmadaki sürü matrisinin her bir elemanı, bileşen (individual) sürü matrisinin her bir elemanı ve satırı ise parçacık, (particles) olarak tanımlanmıştır. GYFO algoritması yapay fizik algoritması [156] kullanılarak elde edilmiş bir algoritmadır. Algoritmanın ikinci adımı ise kuvvetin hesaplanmasıdır. Kuvvet, kütle ve parçacıklar arasındaki mesafe farkı kullanılarak hesaplanır.

Denklem 6.4, 6.5 ve 6.6’da yapay fizik algoritmasında kullanılan farklı kütle hesaplama formülleri verilmektedir. Bu formüller kullanılarak farklı kütle değerleri elde edilir.

$$m_1 = e^{\frac{f(p_{best})-f(p_i)}{f(p_{worst})-f(p_{best})}}, \quad \forall i \quad (6.4)$$

$$m_2 = \tanh(p) = \frac{e^{\frac{f(p_{best})-f(p_i)}{f(p_{worst})-f(p_{best})}} - e^{-\frac{f(p_{best})-f(p_i)}{f(p_{worst})-f(p_{best})}}}{e^{\frac{f(p_{best})-f(p_i)}{f(p_{worst})-f(p_{best})}} + e^{-\frac{f(p_{best})-f(p_i)}{f(p_{worst})-f(p_{best})}}}, \quad \forall i \quad (6.5)$$

$$m_3 = \arctan(p) = \arctan\left(\frac{f(p_{best})-f(p_i)}{f(p_{worst})-f(p_{best})}\right), \quad \forall i \quad (6.6)$$

Denklemlerdeki ;

$p_i$ , ilgili parçacığın mevcut konum vektörünü

$f(p_i)$ , mevcut parçacık vektörünün hata fonksiyonu değerini

$f(p_{best})$ , ilgili iterasyonda ki en küçük hata fonksiyonu değerine sahip parçağı gösterir.

Denklemlerde en iyi ve en kötü değerler  $p_{best} = \min f(p_i)$  ve  $f(p_{worst})$

$p_{worst} = \max f(p_i)$  ifadeleriyle bulunmaktadır.

Üçüncü adımda ise elde edilen kütleler dikkate alınarak her parçacığın kuvvet değeri Denklem 6.7’ye göre hesaplanır.

$$F_{ij,k} = \left\{ \begin{array}{l} Gm_i m_j (p_{j,k} - p_{i,k}) \quad \text{if } f(p_j) < f(p_i) \\ -Gm_i m_j (p_{j,k} - p_{i,k}) \quad \text{if } f(p_j) \geq f(p_i) \end{array} \right\} \quad \text{for } \forall i \neq j \text{ and } i \neq \text{best} \quad (6.7)$$

Denklemdaki  $F_{ij,k}$  ilgili parçacığın kuvvetini,  $p_{i,k}$  ve  $p_{j,k}$  ilgili parçacığın yönünü,  $F_{i,k}$  ise  $k^{th}$  bileşenin toplam kuvvetini gösterir.

Elde edilen değerlere göre Denklem 6.8’de toplam kuvvet hesaplanır.

$$F_{i,k} = \sum_{j=1}^{N_{nop}} F_{ij,k} \quad \forall i \neq best \quad (6.8)$$

Son adım olarak elde edilen kuvvet ve kütle değerleri kullanılıp Denklem 6.9’da verilen ilgili parçacığın hızı hesaplanır. Bu hız ile ilgili parçacığın hareket miktarı belirlenir. Daha sonra Denklem 6.10’da verilen denklem ile ilgili parçacığa hesaplanan hız değeri eklenerek parçacığın yeni değeri elde edilir.

$$v_{i+1,k} = wv_{i,k} + \psi F_{i,k} / m_i \quad \forall i \neq best \quad (6.9)$$

$$p_{i+1,k} = p_{i,k} + v_{i,k} \quad \forall i \neq best \quad (6.10)$$

$v_{i,k}$  ve  $p_{i,k}$  sırasıyla parçacığın hızını ve yeni konumu gösterir.  $\psi$ , [0,1] aralığında değişen rassal bir değerdir.  $w$ ,  $0 \leq w < 1$  aralığında kullanıcı tarafından tanımlanan ağırlık değeridir. Tüm parçacıkların yeni değerleri bulunduktan sonra maliyet fonksiyonu elde edilen bu yeni değerlere göre güncellenir. Kütle ve hareket değerlerinin hesaplanması algoritma başlangıcında belirlenen maksimum iterasyon sayısı kadar devam eder.

Önerilen GYFO algoritması sürü matrisini temel optimizasyon algoritmasına göre oluşturur. Önerilen algoritmada kullanılan kütle, kuvvet ve hareket değerleri de Yapay Fizik Algoritmasının denklemleri kullanılarak elde edilir.

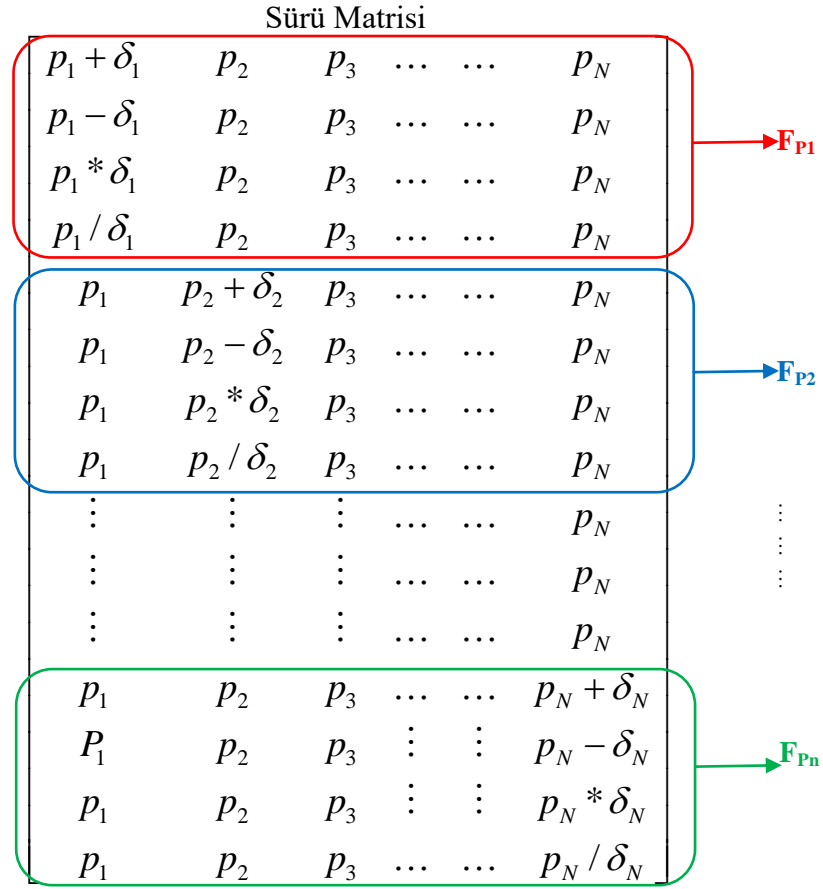
Optimizasyon problemlerinde başlangıç değerlerinin belirlenmesi kritik bir öneme sahiptir. Önerilen GYFO algoritması toplama (+), çıkarma (-), çarpma (\*) ve bölme (/) matematiksel operatörlerini kullanarak verilen başlangıç değerlerinin yakın çevresini ve uzak çevresini inceleyebilir. Böylece başlangıç değerleri rassal olarak verilse bile algoritma her zaman lokal noktalardan kaçarak global çözüme ulaşabilmektedir. Bu yapı algoritmanın avantajlarından biri olarak gösterilebilir.

Şekil 6.1’de verilmiş olan matris yapısı kullanılarak  $[p_1, p_2, p_3, \dots, p_n]$  parametre vektörü için  $[\delta_1, \delta_2, \delta_3, \dots, \delta_n]$  şeklinde verilen rassal hareketler temel matematik operatörleriyle hareket ettirilerek sürü matrisi oluşturulur.

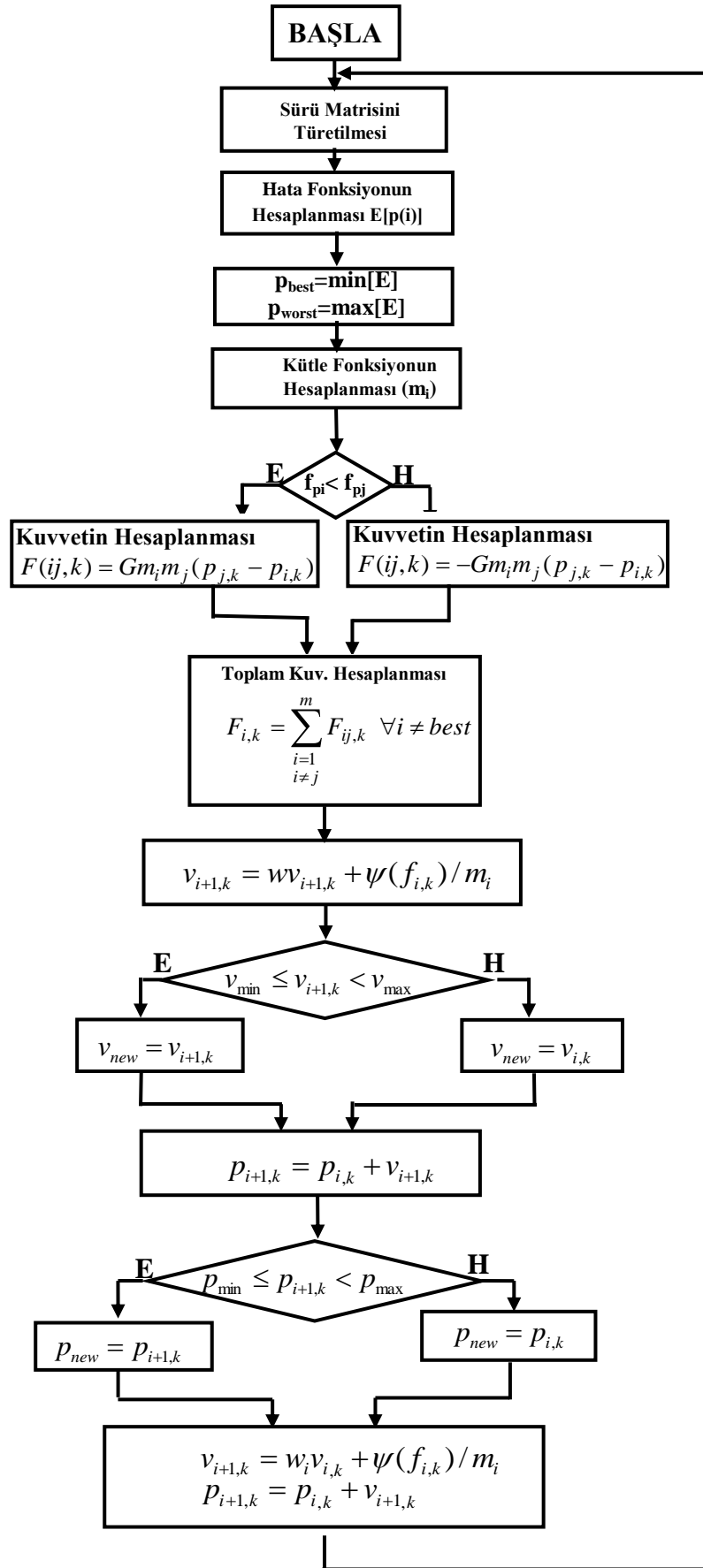
$p_i$ , parametre vektöründeki tüm bileşenler için temel matematiksel operatörler kullanılarak hareketler  $\delta_i$ , rassal değişken değerince yaptırılır. Bir bileşene hareket yaptırılırken diğer bileşenler sabit tutulur. Böylece hangi bileşenin değişimi neticesinde iyi sonucun alındığı tespit edilerek en iyi hareket konfigürasyonu tespit edilir. Bu prosedür her parametre için  $(n \times 4n)$  matris boyutuna ulaşına kadar devam eder.  $n$ , fonksiyondaki parametre sayısını gösterir. Daha sonra Denklem 6.4-6.6’daki kütle fonksiyonları kullanılarak  $(1 \times n)$  boyutundaki kütle matrisi hesaplanır.

Farklı kütle fonksiyonlarının optimizasyon sürecine farklı etkileri vardır. Tezin takip eden bölümlerinde kütle fonksiyon tipinin optimizasyon sürecine olan etkileri detaylarıyla irdelenecektir.

Bir sonraki adımda Denklem 6.7 kullanılarak  $(1 \times n)$  boyutundaki kuvvet matrisi ve Denklem 6.8 kullanılarak her  $p_i$  parametresi için toplam kuvvet hesaplanır. Temel optimizasyon algoritmasına göre 4 farklı şekilde hareket ettirilen her bir parametrenin kuvvet değeri toplam kuvvet değeri hesabında kullanılır. Daha sonra Denklem 6.9 kullanılarak  $(1 \times n)$  boyutunda hız vektörü elde edilir. Sonuç olarak parametrenin yeni değeri algoritmik süreç içerisinde hesaplanan kütle, kuvvet ve hız değerleri kullanılarak elde edilir. Denklem 6.10’da ise parametrenin yeni değerinin hesaplandığı formül bulunur. Şekil 6.2’de gösterilen GYFO yönteminin akış şemasında işlem detayları sunulmuştur.



**Şekil 6.1** Sürü matrisi konfigürasyonu



Şekil 6.2 GYFA'nın akış şeması

### 6.3 Geliştirilmiş Yapay Fizik Optimizasyon Algoritması ile Denetçi Tasarımı

Bu bölümde GYFO algoritması ile klasik PID ve KDPID denetçilerin optimizasyonla tasarlanması amaçlanmıştır. Önerilen algoritmanın tüm kuralları PID ve KDPID tasarlayabilecek şekilde oluşturulmuş ve takip eden iki bölümde klasik PID ve KDPID tasarımı yapabilen optimizasyon süreçleri anlatılmıştır.

#### 6.3.1 PID Denetçi Parametrelerinin Ayarlanması

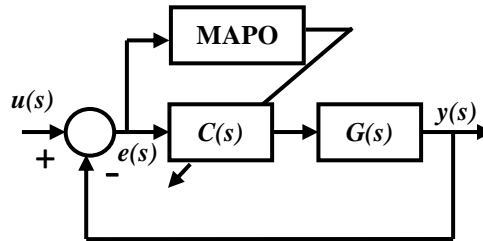
Oransal integral türev (PID) denetçisi özellikle endüstriyel kontrol sistemlerinde yaygın olarak kullanılır [159, 160]. PID denetçi basit kontrol sistemlerinin en erken kontrol stratejisidir. Bundan dolayı özellikle endüstride geniş bir uygulama alanı bulmaktadır. Literatürde birçok PID tipi kontrol metodu bulunur [159]. Bundan dolayı denetçi parametrelerinin ayarlanması sistemler için hayati öneme sahiptir.

Bu bölümde PID denetçinin parametreleri GYFO algoritması ile optimizasyonla elde edilmektedir. PID denetçinin bilindiği gibi 3 adet parametresi  $(k_p, k_i, k_d)$  bulunur. Her parametrenin rassal olarak hareketini sağlamak için her parametreye ayrı ayrı 3 adet  $[\delta_{k_p}, \delta_{k_i}, \delta_{k_d}]$  şeklinde rassal hareket belirlenmiştir. Önceki bölümlerde anlatıldığı gibi Temel optimizasyon algoritması yardımıyla tanımlanan bu üç hareket için sürü matrisi türetilmiştir. Bu matris Denklem 6.3 ve Şekil 6.3'e göre oluşturulmuştur. Sürü matrisindeki her bir satır sırayla Şekil 6.4'de verilen kapalı döngü kontrol sistemine uygulanmıştır. Optimizasyon süresince kapalı döngü sistemde hesaplanan hata değeri minimize edilmeye çalışılmıştır. Şekil 6.4'deki  $G(s)$  sistemin matematiksel modelini,  $e(s)$  hata değerini,  $u(s)$  giriş sinyalini ve  $y(s)$  ise kapalı döngü sistemin çıkışını gösterir. Sürü matrisindeki her bir satır Şekil 6.4'deki simülasyon modeline uygulanarak  $e(s)$  hata değeri hesaplanır. Tezin bu bölümünde gerçekleştirilen bu çalışmada optimizasyon süresince 4 farklı tip hata değeri kullanılmaktadır. Bunlar ISE, MSE, ITAE ve IAE'dir. Çünkü hata fonksiyon tanımları optimizasyon süreçlerini etkilemektedir. Farklı hata fonksiyonlarına göre algoritma tekrar tekrar

çalıştırılarak ilgili optimizasyon problemi için uygun hata fonksiyonu tanımı tespit edilmeye çalışılmıştır.

$$\begin{array}{c}
 \left[ \begin{array}{ccc}
 \textit{PID sürü Matrisi} & & \\
 k_p + \delta_{k_p} & k_i & k_d \\
 k_p - \delta_{k_p} & k_i & k_d \\
 k_p * \delta_{k_p} & k_i & k_d \\
 k_p / \delta_{k_p} & k_i & k_d \\
 k_p & k_i + \delta_{k_i} & k_d \\
 k_p & k_i - \delta_{k_i} & k_d \\
 k_p & k_i * \delta_{k_i} & k_d \\
 k_p & k_i / \delta_{k_i} & k_d \\
 k_p & k_i & k_d + \delta_{k_d} \\
 k_p & k_i & k_d - \delta_{k_d} \\
 k_p & k_i & k_d * \delta_{k_d} \\
 k_p & k_i & k_d / \delta_{k_d}
 \end{array} \right]
 \begin{array}{c}
 \left[ \begin{array}{c}
 E_1 \\
 E_2 \\
 E_3 \\
 E_4 \\
 E_5 \\
 E_6 \\
 E_7 \\
 E_8 \\
 E_9 \\
 E_{10} \\
 E_{11} \\
 E_{12}
 \end{array} \right]
 \end{array}
 \end{array}$$

Şekil 6.3 PID denetçi için oluşturulan sürü matrisi



Şekil 6.4 Optimizasyonda kullanılan sistem modeli

Çalışmada kullanılan hata fonksiyonlarına ilişkin kısa tanımlamalar aşağıda verilmektedir [161].

- Integral Square Error (ISE) Karesel Hatanın İntegrali; Hatanın karesini alarak negatif bileşenleri elimine etmektedir.

- Mean Square Error (MSE) Kareysel Hatanın Ortalaması; hedef değerdeki tüm değişimleri ve sapmaları yansıtmaktadır.
- Integral Time Absoulate Error (ITAE) Mutlak Hata ile Çarpılan Zamanın İntegrali; zamanla hatanın değerine bir ağırlık fonksiyonu atanır. Bu yapıda hatanın ilk değerinin yanı sıra hatanın toplam değeri göz önünde bulundurulur.
- Integral Absolulate Error (IAE) Mutlak Hatanın İntegrali; bu yapı yardımıyla hatadaki negatif bileşenler elenmektedir.

Kullanılan hata fonksiyonlarının (maliyet fonksiyonlarının) matematiksel formu Denklem 6.11' de verilmektedir.

$$J_{ISE} = \int_0^T e^2(t) dt; \quad J_{MSE} = \frac{1}{n} \sum_{i=1}^n (e(t))^2; \quad J_{ITAE} = \int_0^T t|e(t)| dt; \quad J_{IAE} = \int_0^T |e(t)| dt \quad (6.11)$$

Kütle fonsiyon değeri Denklem 6.4-6.6'daki formülasyonlar ile ilgili hareket için bulunan  $f(p_i)$ ,  $f(p_{best})$  ve  $f(p_{worst})$  değerleri kullanılarak hesaplanır. Bu çalışmada her kütle fonksiyonu tanımı için her hata fonksiyon tanımı eşleştirmesi yapılarak tüm konfigürasyonlar incelenmektedir. Yani 3 adet kütle fonksiyonu tanımı 4 adet hata fonksiyonu tanımı bulunmaktadır. Bir optimizasyon problemi için uygun eşleştirmeler ile algoritma aynı şartlarda toplam 12 kez çalıştırılarak algoritma için uygun kütle-hata fonksiyonu ikilisi tespit edilmiştir. Daha sonra hesaplanan değerler kullanılarak kuvvet fonsiyonu değerleri hesaplanarak Denklem 6.12'ye kaydedilmiştir.

$$[F_1 F_2 F_3 F_4 F_5 F_6 F_7 F_8 F_9 F_{10} F_{11} F_{12}]^T \quad (6.12)$$

Denklem 6.8 deki kuvvet hesaplama formülü  $k_p$ ,  $k_i$  ve  $k_d$  PID denetçi parametreleri için güncellenerek Denklem 6.13-6.15 elde edilir. Bu denklemler ile parametreler için ayrı ayrı kuvvet fonksiyonları hesaplanır.

$$F_{i,k}^{k_p} = \sum_{j=1}^4 F_{ij,k} \quad \forall i \neq best \quad (6.13)$$

$$F_{i,k}^{k_i} = \sum_{j=5}^8 F_{ij,k} \quad \forall i \neq best, \quad (6.14)$$

$$F_{i,k}^{k_d} = \sum_{j=9}^{12} F_{ij,k} \quad \forall i \neq best \quad (6.15)$$

Bu kuvvet fonksiyonları yardımıyla Denklem 6.9'da verilen hız denklemleri PID denetçi parametrelerinin hızını tespit etmek için güncellenerek Denklem 6.16-6.18'de verilmekte olan hız denklemleri elde edilir.

$$v_{i+1,k}^{k_p} = \omega v_{i,k}^{k_p} + \psi F_{i,k}^{k_p} / m_i, \quad \forall i \neq best \quad (6.16)$$

$$v_{i+1,k}^{k_i} = \omega v_{i,k}^{k_i} + \psi F_{i,k}^{k_i} / m_i, \quad \forall i \neq best \quad (6.17)$$

$$v_{i+1,k}^{k_d} = \omega v_{i,k}^{k_d} + \psi F_{i,k}^{k_d} / m_i, \quad \forall i \neq best \quad (6.18)$$

Denklem 6.10 denetçi parametreleri için güncellenerek Denklem 6.19-6.21'de bulunan yeni parametre değerleri için elde edilen formüller bulunmaktadır.

$$k_{p_{i+1,k}} = k_{p_{i,k}} + v_{i+1,k}^{k_p}, \forall i \neq best \quad (6.19)$$

$$k_{i_{i+1,k}} = k_{i_{i,k}} + v_{i+1,k}^{k_i}, \forall i \neq best \quad (6.20)$$

$$k_{d_{i+1,k}} = k_{d_{i,k}} + v_{i+1,k}^{k_d}, \forall i \neq best \quad (6.21)$$

Şekil 6.5’de önerilen algoritmanın sözde kodu verilmektedir. Bu kod kullanılarak Şekil 6.4’de verilen simulink modeli arayıcılığıyla belirlenen iterasyonda çalıştırılarak uygun  $k_p$ ,  $k_i$  ve  $k_d$  değerleri bulunmaktadır.

**Start**  
**for**  $i=1$ :Maksimum İterasyon Sayısı  
**Sürü Matrisinin Oluşturulması**  
**Hata fksiyonun hesaplanması [E] ISE, MSE, ITAE, IAE**  
 $f(p_{best}) = \min E[i]$   
 $f(p_{worst}) = \max E[i]$   
**if**  $f(p_i) < f(p_j)$   
 $F(ij, k) = Gm_i m_j(p_{j,k}, p_{i,k})$   
**else**  
 $F(ij, k) = -Gm_i m_j(p_{j,k}, p_{i,k})$   
**end**  

$$Fi_{i,k} = \sum_{\substack{i=1 \\ i \neq j}}^m F(ij, k) \quad \forall i \neq best$$
  
**for**  $n=1$ :Maksimum optimizasyon yapılan parametre sayısı  
 $v_{n+1,k} = wv_{n+1,k} + \psi f_{n,k} / m_i$   
**If**  $v_{min} \leq v_{n+1,k} < v_{max}$   
 $v_{new} = v_{n+1,k}$   
**else**  
 $v_{new} = v_{n,k}$   
**end**  
 $p_{n+1,k} = p_{nk} + v_{n+1,k}$   
**If**  $p_{min} \leq p_{n+1,k} < p_{max}$   
 $p_{new} = p_{n+1,k}$   
**else**  
 $p_{new} = p_{n+1,k}$   
**end**  
**end**  
**end**

**Şekil 6.5** GYFO algoritmasının sözde kodu

### 6.3.2 Geliştirilmiş Yapay Fizik Optimizasyon Algoritması ile Kesir Dereceli PID Denetçi Parametrelerinin Optimizasyonu

Bu bölümde, önerilen optimizasyon algoritmasıyla KDPID parametrelerinin optimizasyon prosedürü anlatılmaktadır.

Birçok endüstriyel kontrol uygulamasında tam sayı dereceli PI, PD veya PID'ler kullanılır [94]. KD integro diferansiyel ifadelerin pratik olarak gerçekleştirilmesini mümkün kılan çalışmalardan sonra KDPID denetçilere ilgi artmıştır. KD denetçinin matematiksel formu  $C_{PI^\lambda D^\mu}(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu$  şeklinde verilmektedir [41]. KDPID optimizasyonu içinde Şekil 6.4'de verilen simülasyon modeli kullanılmıştır.

İlk üç parametre olan  $(k_p, k_i, k_d)$ 'nin optimizasyonu PID denetçinin optimizasyon prosedürü ile aynıdır. Bu bölümde bunlara ek olarak  $(\lambda, \mu)$ 'nün optimizasyonu için gerekli olan denklemler türetilmiştir. Önceki bölümlerde de bahsedildiği gibi KD integro diferansiyel ifadeler yaklaşım yöntemleri kullanılarak gerçekleştirilmektedir. Literatürde CFE [162], Carlson's method [163] and Matsuda's method [163] gibi birçok yaklaşım metodu bulunur. Bu çalışmada ise diğer çalışmalarda da olduğu gibi “ninteger v.2.3” [64] simulink araç kutusu kullanılacaktır.

Kesir dereceli PID denetçide optimize edilmesi gereken 5 parametreye sahip olduğu için  $[5 \times 20]$  büyüklüğünde bir sürü matrisine sahiptir.  $k_p, k_i, k_d, \lambda, \mu$  parametreleri için 5 farklı rassal hareket  $(\delta_{k_p}, \delta_{k_i}, \delta_{k_d}, \delta_\lambda, \delta_\mu)$  tanımlanmıştır. KDPID'nin sürü matrisi Şekil 6.1'de gösterildiği gibi elde edilir. Matrisin ilk 12 satırı PID denetçi için Şekil 6.3'de verilmiş olan yöntem ile benzer şekilde elde edilir. Kalan diğer 8 satır ise kesirli dereceler  $\lambda$  ve  $\mu$  için  $\lambda + \delta_\lambda, \lambda - \delta_\lambda, \lambda * \delta_\lambda, \lambda / \delta_\lambda, \mu + \delta_\mu, \mu - \delta_\mu, \mu * \delta_\mu$  ve  $\mu / \delta_\mu$  şeklinde türetilir. İlk üç parametre için PID denetçinin sunulduğu kısımda türetilen formüller kesir dereceli PID için de geçerlidir.

$\lambda$  ve  $\mu$  parametreleri için ise denklemler şöyledir. Denklem 6.22 ve 6.23'de  $\lambda$ ,  $\mu$  değerleri için kuvvet fonksiyonları verilmektedir.

$$F_{i,k}^{\lambda} = \sum_{j=12}^{16} F_{ij,k} \quad \forall i \neq best \quad (6.22)$$

$$F_{i,k}^{\mu} = \sum_{j=16}^{20} F_{ij,k} \quad \forall i \neq best \quad (6.23)$$

Benzer şekilde Denklem 6.24 ve Denklem 6.25'de  $\lambda$ ,  $\mu$  için hız denklemleri gösterilmektedir.

$$v_{i+1,k}^{\lambda} = w\lambda_{i,k} + \psi F_{i,k}^{\lambda} / m_i, \quad \forall i \neq best \quad (6.24)$$

$$v_{i+1,k}^{\mu} = w\mu_{i,k} + \psi F_{i,k}^{\mu} / m_i, \quad \forall i \neq best \quad (6.25)$$

Denklem 6.26 ve Denklem 6.27'de parametrelerin yeni değerlerinin türetildiği formüller verilmiştir. Parametrelerin yeni değerleri her iki denetçi tasarım metodu içinde geçerli olmak şartıyla  $v_{\max}$  ve  $v_{\min}$  hız değerleriyle sınırlandırılarak daha etkili bir optimizasyon süreci elde edilmiştir.

$$\lambda_{i+1,k} = \lambda_{i,k} + v_{i+1,k}^{\lambda}, \quad \forall i \neq best \quad (6.26)$$

$$\mu_{i+1,k} = \mu_{i,k} + v_{i+1,k}^{\mu}, \quad \forall i \neq best \quad (6.27)$$

## 6.4. Simülasyon Çalışması

Bu bölümde önerilen optimizasyon algoritmasıyla kesir dereceli ve tam sayı dereceli matematiksel modellere KDPID ve klasik PID denetçiler tasarlanmıştır. Elde edilen denetçiler literatürde var olan yöntemlerle karşılaştırmalı olarak sunulmuştur.

GYFO algoritması kütle hesaplaması yaparken 3 farklı kütle fonksiyonu kullanmaktadır. Bunlar  $e^p$ ,  $\tanh(p)$  ve  $\arctan(p)$ 'dir. Kütle fonksiyonun tipi optimizasyon sürecine de etki etmektedir. En iyi kütle fonksiyon tanımı optimize edilen sisteme göre de değişim gösterebilir. Diğer taraftan optimizasyonda kullanılan hata fonksiyonu tanımında optimizasyon sürecini etkiler. Çünkü farklı hata fonksiyon tanımları süreç boyunca farklı değerler türettiğinde algoritma farklı sonuçlar verebilir. Bunda dolayı önerilen GYFO algoritması farklı hata-kütle ikilileri ile çalıştırılmıştır. Bu durumda 12 farklı olasılık mevcuttur. Algoritma 12 farklı konfigürasyonda eş zamanlı çalıştırılarak ilgili problem için en etkili kütle-hata konfigürasyonu bulunmaktadır. Simülasyon çalışmasında 3 farklı  $C(s)G(s)$  kullanılarak optimizasyon gerçekleştirilmiştir. Birinci örnekte tam sayı dereceli sisteme tam sayı dereceli PID, ikinci örnekte tam sayı dereceli sisteme KDPID ve üçüncü örnekte ise KD sisteme KDPID tasarlanmıştır.

**6.4.1 Simülasyon Örnek 1:** Bu örnekte [164]'den alınmış olan ve Denklem 6.28 ile verilmiş olan matematiksel model kullanılmıştır;

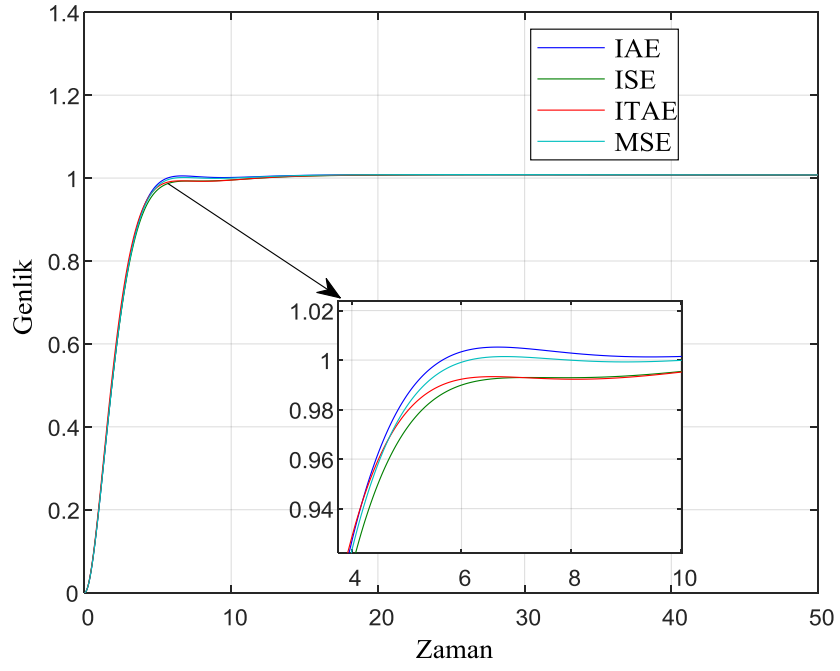
$$G_1 = \frac{0.1}{s(3s + 1)(0.8s + 1)} \quad (6.28)$$

PID denetçi yapısı ve Denklem 6.28'de gösterilmiş model Şekil 6.4'deki sisteme uygulanmaktadır. Bu sistem MATLAB derleyicide oluşturulan optimizasyon algoritması kodu ile eş zamanlı çalışır. İlgili matematiksel model için en iyi PID denetçi konfigürasyonu türetilmiştir. İlgili simülasyon modeli 50 sn. çalıştırılmış ve MATLAB çözücüsü olarak ode45 kullanılmıştır.

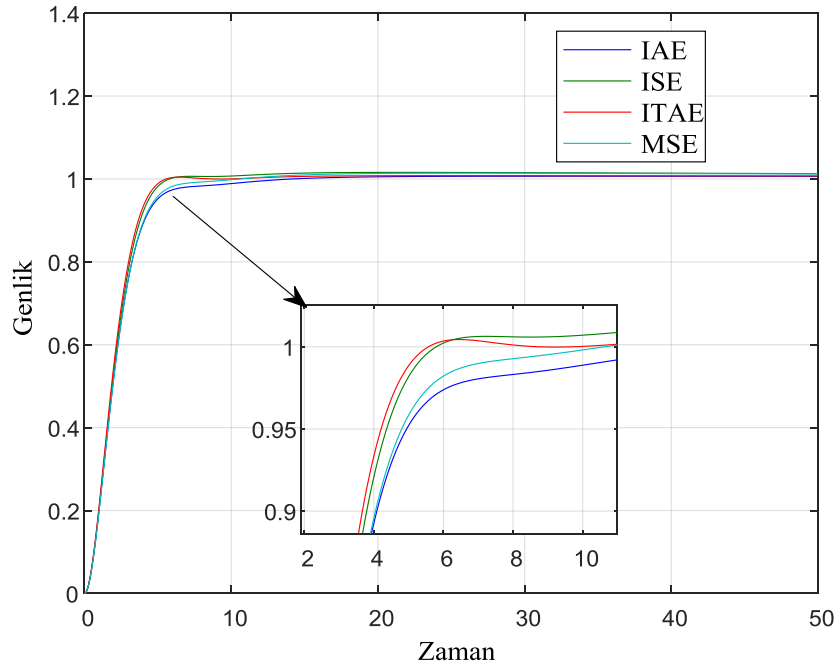
Çizelge 6.1 de  $C_1(s)$  PID denetçisi için 12 farklı kütle-hata konfigürasyonuna göre elde edilen denetçi parametleri bulunmaktadır. 12 farklı optimizasyon algoritması eş zamanlı çalıştırılarak parametreler adil bir karşılaştırmayla elde edilmiştir. Şekil 6.6, 6.7 ve 6.8’de  $C_1(s)G_1(s)$  için kullanılan 3 farklı kütle fonksiyonuna ( $m_1 = e^p$ ,  $m_2 = \tanh(p)$  and  $m_3 = \arctan(p)$ ) göre ayrı ayrı elde edilmiş 4 farklı hata fonksiyonunun birim basamak cevapları karşılaştırılmıştır. Şekil 6.9’da ise en iyi kütle fonksiyonu için 4 farklı hata fonksiyonu tanımlarının karşılaştırması verilmektedir. Çizelge 6.1 ve Şekil 6.9’dan anlaşılacağı üzere  $C_1(s)G_1(s)$  için en iyi kütle-hata konfigürasyonu  $\tanh(p) - ITAE$ ’dir. Şekil 6.10’da optimizasyon süresince hata fonksiyonun değişimi gösterilmiştir. Şekil 6.11’de ise optimizasyon süresince parametrelerin değişim grafiği verilmektedir. Görüldüğü gibi GYFO algoritması 50. iterasyonda ilgili problem için optimal çözüm üretebilmiştir. Bunlara ek olarak Şekil 6.12’de aynı matematiksel model için DEchaos ve SOMAchaos [164] algoritmalarıyla elde edilmiş birim basamak cevaplarıyla önerilen optimizasyon algoritmasının birim basamak cevabı karşılaştırılmıştır. Şekilden de anlaşılacağı üzere önerilen algoritma literatürde olan diğer yöntemlere göre daha iyi sonuçlar üretebilmektedir.

**Çizelge 6.1**  $C_1(s)G_1(s)$  sistemi için kütle-hata ikilisinin optimizasyon sonuçları (Kırmızı renk ile yazılmış sonuçlar ilgili kütle için en iyi değeri, yeşil ile yazılmış sonuçlar ise en iyi kütle-hata ikilisini göstermektedir.)

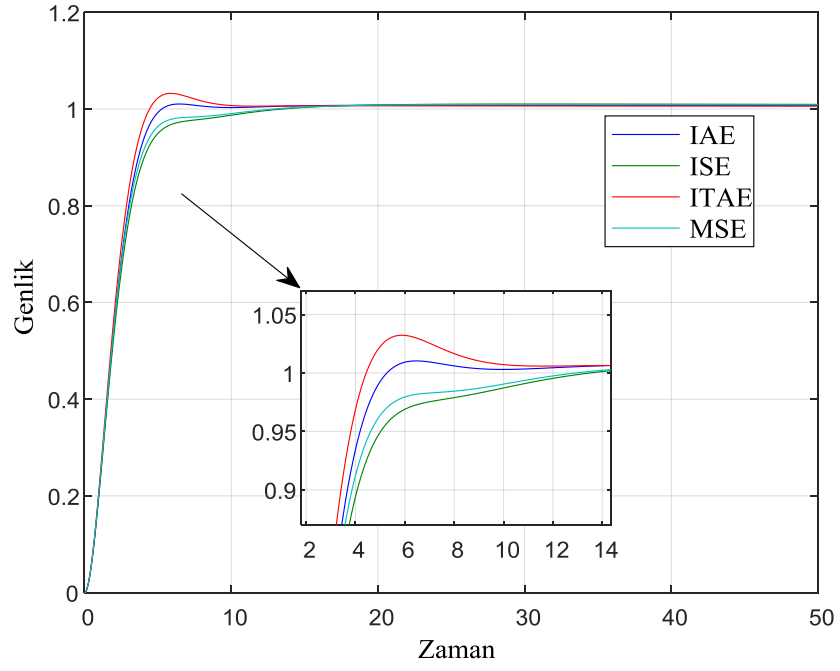
Kütle Fonksiyonu	Hata Fonksiyonu	$J$	$k_p$	$k_i$	$k_d$
$m_1 = e^p$	ITAE	5.4977	4.6947	0.0200	16.3875
	IAE	2.0960	4.7809	0.0200	15.6033
	ISE	1.3524	4.5746	0.0178	15.7805
	<b>MSE</b>	<b>0.0346</b>	<b>4.7105</b>	<b>0.0200</b>	<b>15.6404</b>
$m_2 = \tanh(p)$	<b>ITAE</b>	<b>5.1307</b>	<b>4.8412</b>	<b>0.0200</b>	<b>15.9800</b>
	IAE	2.2404	4.2643	0.0128	15.2892
	ISE	1.3632	4.6682	0.0386	15.4616
	MSE	0.0375	4.3212	0.0289	15.1089
$m_3 = \arctan(p)$	ITAE	5.2004	5.4459	0.0200	16.1804
	<b>IAE</b>	<b>2.0847</b>	<b>4.9002</b>	<b>0.0200</b>	<b>15.6860</b>
	ISE	1.3797	4.2006	0.0200	15.7162
	MSE	0.0363	4.4342	0.0200	16.1929



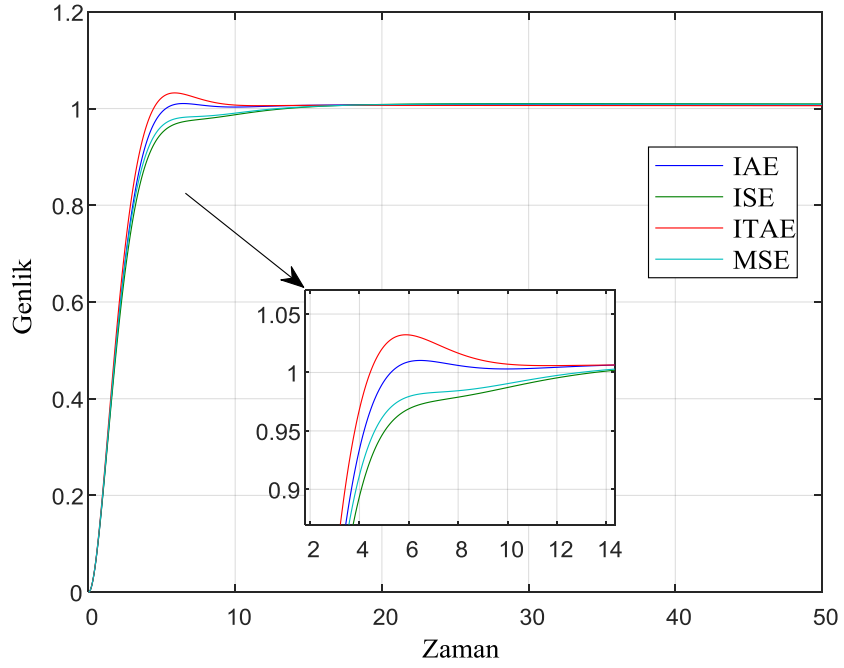
Şekil 6.6  $m_1$  kütle tanımı için farklı hata fonksiyon tanımlarının birim basamak cevapları



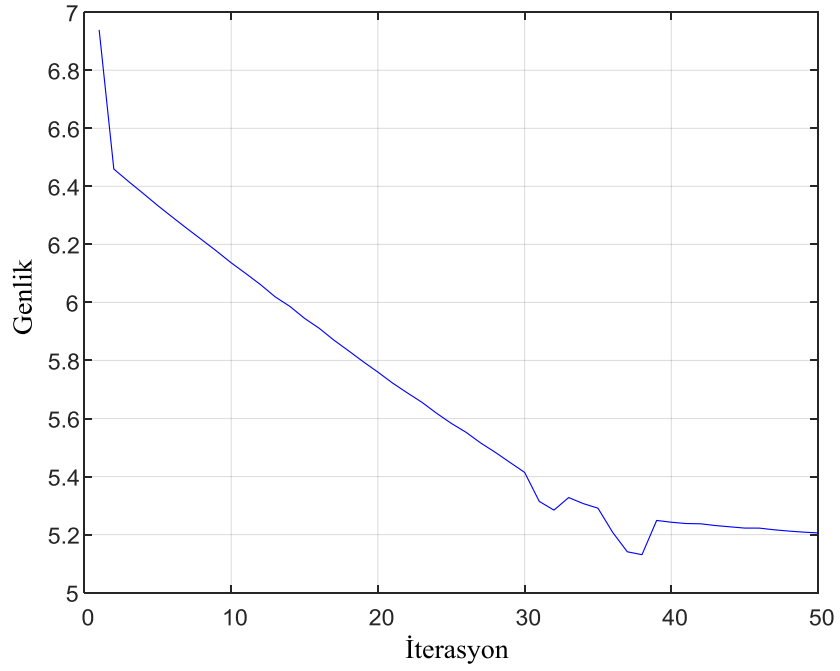
Şekil 6.7  $m_2$  kütle tanımı için farklı hata fonksiyon tanımlarının birim basamak cevapları



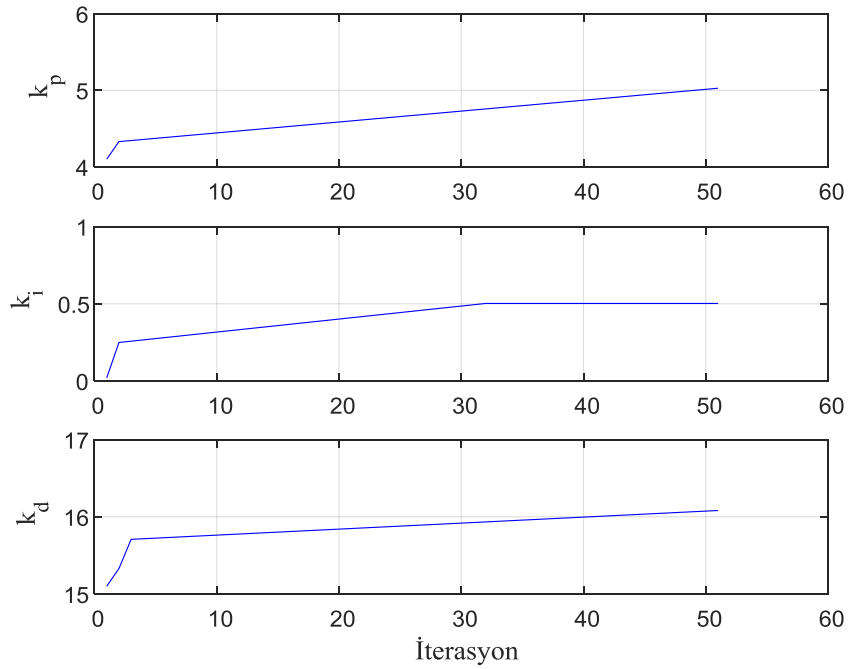
Şekil 6.8  $m_3$  kütle tanımı için farklı hata fonksiyon tanımlarının birim basamak cevapları



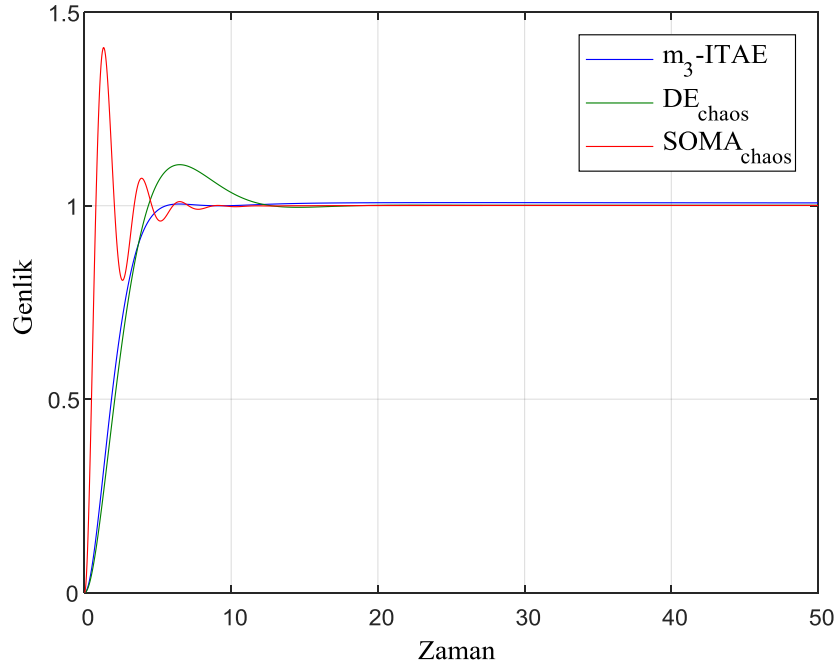
Şekil 6.9 Çizelge 6.1’de verilmiş olan en iyi hata fonksiyon tanımlarının birim basamak cevapları



**Şekil 6. 10** Çizelge 6.1’de verilen en iyi kütle-hata ikilisi olan ( $m_3$ -ITAE) yapısının optimizasyon süresince maliyet fonksiyonunun değişimi



**Şekil 6. 11** Çizelge 6.1’de verilen en iyi kütle-hata ikilisi olan ( $m_3$ -ITAE) yapısının optimizasyon süresince parametre değişimi



**Şekil 6.12** Denklem 6.29 de verilmiş [164]'de alınmış sistem için GYFO Algoritması ile diğer yöntemlerin birim basamak cevaplarının karşılaştırılması.

**6.4.2 Simülasyon Örnek 2:** Bu örnekte [30]'dan alınmış olan tam sayı dereceli matematiksel modele KDPID tasarlanmıştır.

$$G_2 = \frac{s^2}{50s + 400} \quad (6.29)$$

$C_2(s)$  Kesir dereceli PID olsun,  $C_2(s)G_2(s)$  Şekil 6.4'de verilmiş olan simülasyon modelinde kullanılmıştır. 'Ninteger' araç kutusu [64] KDPID'nin pratik olarak gerçekleştirilmesi için kullanılmıştır. Birinci örnekte olduğu gibi en iyi kütle-hata ikilisinin bulunması için 12 konfigürasyon eş zamanlı çalıştırılmıştır. İlgili matematiksel model için en uygun hata kütle konfigürasyonuna sahip KDPID denetçi tasarlanmıştır. Uygulama esnasında simülasyon modeli 20 sn. çalıştırılmıştır.

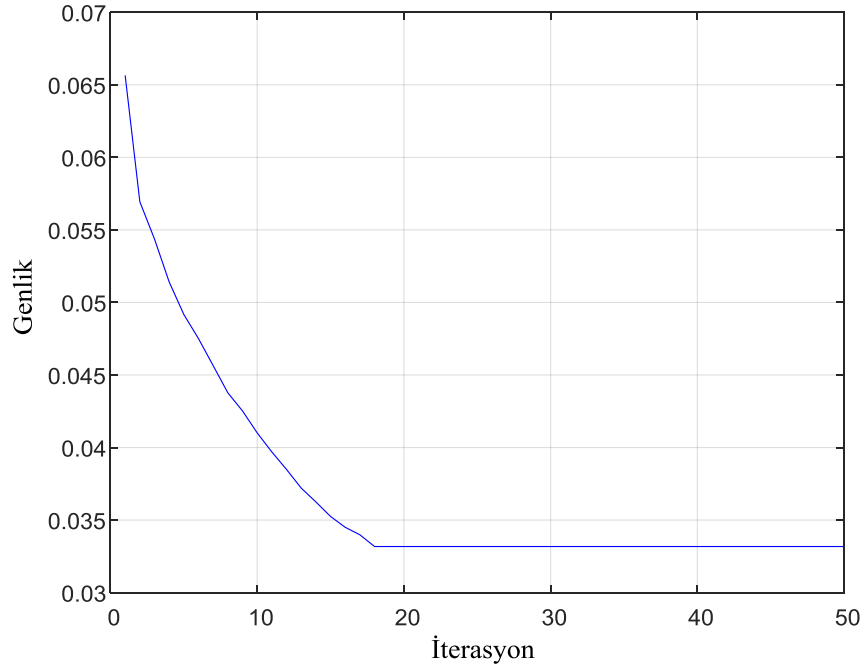
12 farklı kütle-hata konfigürasyonu için elde edilen KDPID denetçi kat sayıları Çizelge 6.2'de verilmektedir. Bu çizelgeye göre  $C_2(s)G_2(s)$  sistemi için en

uygun kütle-hata konfigürasyonu  $e^p - IAE$ 'dir. Şekil 6.13'de optimizasyon esnasındaki hata fonksiyonunun değişimi gösterilmektedir. Şekil 6.14'de ise optimizasyon esnasındaki parametre değişimi verilmiştir. Görüldüğü üzere algoritma 20. iterasyon da optimal çözümü bulmuştur.

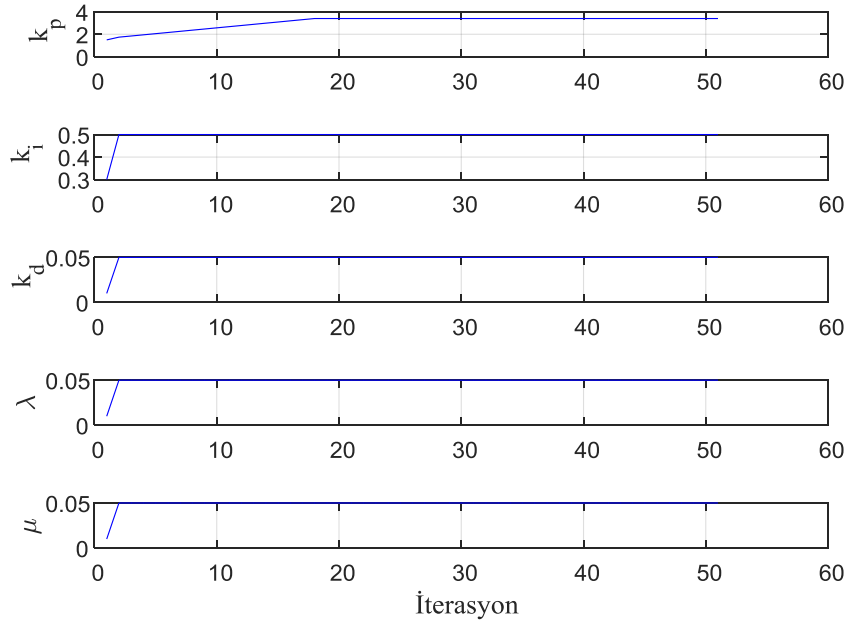
Buna ek olarak aynı matematiksel model için Diferansiyel Gelişim (DE) [30] algoritmasıyla elde edilmiş KDPID'nin birim basamak cevabı ile önerilen yöntemin türetmiş olduğu KDPID'nin birim basamak cevapları Şekil 6.15'de karşılaştırılmıştır. Şekilden de anlaşılacağı üzere önerilen optimizasyon algoritması çok daha iyi kontrol performansı sergileyen sonuçlar üretebilmektedir.

**Çizelge 6.2**  $C_2(s)G_2(s)$  sistemi için kütle-hata ikililerinin optimizasyon sonuçları (Kırmızı renk ile yazılmış sonuçlar ilgili kütlenin en iyi değerini yeşil ile yazılmış sonuçlar ise en iyi kütle-hata ikilisini göstermektedir.)

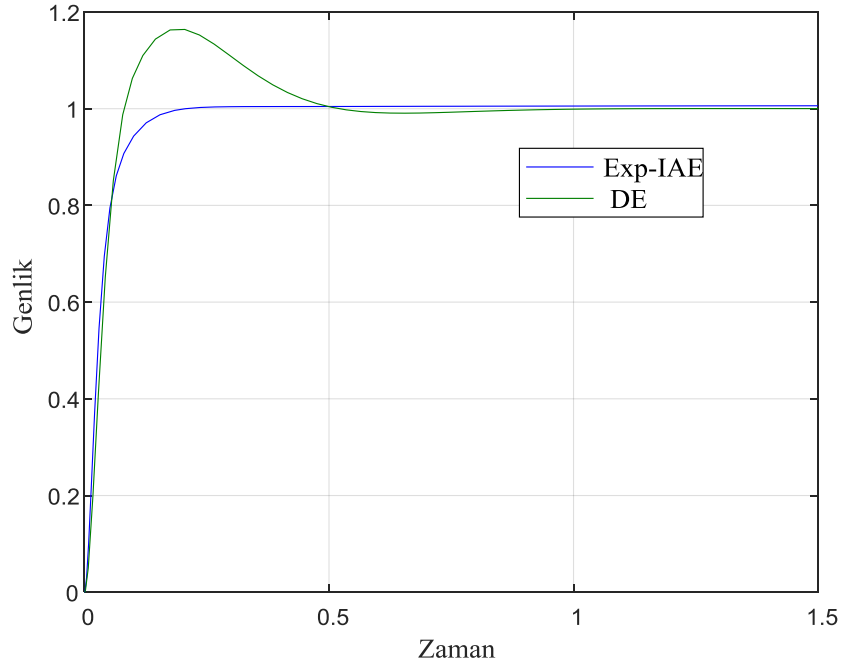
Kütle Fonksiyonu	Hata Fonksiyonu	$J$	$k_p$	$k_i$	$k_d$	$\lambda$	$\mu$
$m_1 = e^p$	ITAE	0.0023	3.4192	0.3000	0.0500	1.3615	1.2222
	<b>IAE</b>	<b>0.0332</b>	<b>3.3968</b>	<b>0.5000</b>	<b>0.0500</b>	<b>1.3765</b>	<b>1.0544</b>
	ISE	0.0143	3.4792	0.4585	0.0500	1.1486	1.2899
	MSE	0.1264	3.4633	0.5000	0.0500	1.2064	1.2950
$m_2 = \tanh(p)$	<b>ITAE</b>	<b>0.0023</b>	<b>3.4593</b>	<b>0.4292</b>	<b>0.0500</b>	<b>1.3956</b>	<b>1.1360</b>
	IAE	0.0326	3.4746	0.4700	0.0500	1.3663	1.1920
	ISE	0.0165	2.9256	0.4639	0.0500	1.1426	1.2465
	MSE	0.1324	2.1518	0.4967	0.0500	1.3747	1.2971
$m_3 = \arctan h(p)$	<b>ITAE</b>	<b>0.0022</b>	<b>3.4840</b>	<b>0.2990</b>	<b>0.0500</b>	<b>1.3241</b>	<b>1.2228</b>
	IAE	0.0346	3.3333	0.4781	0.0408	1.3558	1.2635
	ISE	0.0147	3.4980	0.4927	0.0500	1.3506	1.2558
	MSE	0.1289	3.3831	0.4765	0.0500	1.3777	1.2855



**Şekil 6.13** Çizelge 6.2’de verilen en iyi kütle-hata ikilisi olan ( $m_1$ -IAE) yapısının optimizasyon süresince maliyet fonksiyonunun değişimi



**Şekil 6.14** Çizelge 6.2’de verilen en iyi kütle-hata ikilisi olan ( $m_1$ -IAE) yapısının optimizasyon süresince parametre değişimi



**Şekil 6.15** Denklem 6.29'daki sistem için GYFO Algoritması ile diğer yöntemlerin birim basamak cevaplarının karşılaştırılması.

**6.4.3 Simülasyon Örnek 3:** Bu örnekte de Denklem 6.30'da verilmiş olan KD sistem modeline [165] önerilen optimizasyon algoritmasıyla KDPID tasarlanmıştır.

$$G_3(s) = \frac{1}{5.3538s^{1.0109} + 0.4126s^{0.9}} e^{-s} \quad (6.30)$$

$G_3(s)$ , matematiksel modeli ve  $C_3(s)$  KD denetçisi Şekil 6.4 verilmiş olan optimizasyon modeli içerisinde kullanılmıştır. Birinci ve ikinci örnekteki benzer şekilde  $C_3(s)G_3(s)$  sistemi için uygun kütle hata fonksiyonuna sahip KD denetçiler tasarlanmıştır. Çizelge 6.3'de ilgili tüm konfigürasyonlar verilmektedir. İlgili çizelgeye göre  $C_3(s)G_3(s)$  sistemi için en uygun kütle hata ikilisinin  $\tanh(x)$ -ITAE olduğu belirlenmektedir.

**Çizelge 6.3**  $C_3(s)G_3(s)$  sistemi için kütle-hata çiftlerinin optimizasyon sonuçları (Kırmızı renk ile yazılmış sonuçlar ilgili kütlelerin en iyi değerini yeşil ile yazılmış sonuçlar ise en iyi kütle-hata ikilisini göstermektedir.)

Mass Function	Error Functions		$J$	$k_p$	$k_i$	$k_d$	$\lambda$	$\mu$
$m_1 = e^p$	<b>ITAE</b>	<b>13.2754</b>	<b>1.3802</b>	<b>0.0985</b>	<b>1.1757</b>	<b>1.0847</b>	<b>0.9000</b>	
	IAE	4.1396	1.2033	0.0989	1.1393	1.1832	0.9000	
	ISE	1.8806	1.4646	0.0920	1.2729	1.1723	0.7987	
	MSE	0.0519	1.3965	0.0683	1.2667	1.1764	0.8088	
$m_2 = \tanh(p)$	<b>ITAE</b>	<b>8.6476</b>	<b>1.4712</b>	<b>0.1000</b>	<b>1.2975</b>	<b>1.1987</b>	<b>0.3000</b>	
	IAE	3.8925	1.4974	0.0629	1.2761	1.0874	0.3000	
	ISE	1.7147	1.4845	0.0823	1.2890	1.1960	0.3000	
	MSE	0.0516	1.4943	0.0880	1.2622	1.1944	0.9000	
$m_3 = \arctan h(p)$	ITAE	12.1028	1.4935	0.0954	1.2821	1.1322	0.9000	
	IAE	2.8971	1.4593	0.0867	1.2495	1.1927	0.3000	
	<b>ISE</b>	<b>1.7070</b>	<b>1.4561</b>	<b>0.0956</b>	<b>1.2732</b>	<b>1.1956</b>	<b>0.3000</b>	
	MSE	0.0548	1.3499	0.0912	1.2487	1.1987	0.9000	

Tezin bu bölümünde YFO algoritması TOA ile birleştirilip GYFO algoritması önerilmiştir. Daha sonra farklı sistemler için denetçiler tasarlanmış ve literatürdeki örnekler ile karşılaştırılmıştır. Algoritmanın performansına etki eden kütle ve hata fonksiyon tanımlarına göre farklı konfigürasyonlar için denetçiler tasarlanmış ve en uygun kütle hata konfigürasyonu ilgili problem için tespit edilmiştir.

## 7. ÇOKLU AMAÇ FONKSİYONUNA GÖRE DENETÇİ OPTİMİZASYONU

Tezin önceki bölümlerinde yapılan uygulamalardan da görüleceği gibi optimizasyon süreçlerine etki eden 2 faktör bulunmaktadır. Bunlardan ilki kullanılan optimizasyon algoritması diğeri ise optimizasyon sürecinde kullanılan amaç fonksiyonudur.

Optimizasyon algoritmalarının zekiliği belirlenen amaç doğrultusunda hareket ederken izlenen yolun zeki yöntemlerle belirlenmesi olarak tanımlanabilir. Bundan dolayı optimizasyon probleminin hangi amaç doğrultusunda çözümleneceği algoritmanın başlangıcında belirlenmekte ve problem bu doğrultuda çözümlenmektedir.

Aslında optimizasyon algoritmalarında tüm süreçlere karar veren tek etken amaç fonksiyonudur. Her problem her optimizasyon algoritmasıyla çözümlenemeyeceği gibi her amaç fonksiyonu da problemin çözümlendiği optimizasyon yöntemiyle kullanılamamaktadır.

Bunun için problem çözümüne başlamadan önce problemin ruhunu çok iyi anlamak gerekir. Problem nedir ve nasıl çözülebilir sorularının cevabı bulunduktan sonra ikinci aşamaya geçilmelidir.

İkinci aşama ise ilgili problemin hangi amaç doğrultusunda çözümlerse istenilen sonuçlar üretilebilir sorusudur.

Bahsedilen bu iki durum netleştikten sonra yani problemin ruhu anlaşıldıktan sonra optimizasyon problemi etkili bir şekilde çözülebilir.

Amaç fonksiyonunda ise hangi etkeni koyarsanız, fonksiyon o doğrultuda çalışır. Yani bazen optimizasyon algoritmalarının optimizasyon problemlerinden beklenen sonuçları üretmediği görülmektedir.

Buradaki problem, kullanılan yöntemin yanlışlığından çok kullanılan amaç fonksiyonun yanlışlığıdır. Yani bir problem için ideal olan uygun amaç fonksiyonu belirlenmiş ise problem istenilen birçok optimizasyon algoritmasıyla global çözüme ulaşabilir veya global çözüme çok yakın sonuçlar üretebilir.

Bu tez esnasında yapılan uygulamalarda hata fonksiyonun minimizasyonu amaç fonksiyonu olarak kullanılmıştır. Aslında literatüre bakıldığında da genellikle hata fonksiyonun minimizasyonu amaç fonksiyonu olarak kullanılmaktadır.

Tez kapsamında yapılan çalışmalarda sistemlere denetçiler tasarlanmıştır. Sistemlere denetçi eklemenin amacı giriş sinyali ile çıkış sinyali arasındaki farkı ideal de sıfır yapmadır.

İşin doğası gereği hata fonksiyonunun minimizasyonunun amaç fonksiyonu olarak kullanımı optimizasyonla denetçi tasarım problemleri için ideal bir çözüm olmaktadır. Amaç hata fonksiyonu ile birlikte problemin dinamiğine uygun amaç fonksiyonlarının eklenmesiyle daha etkili çözümler üretilebilir.

Bu tezin önceki bölümlerinde tekli amaç fonksiyonu kullanılmıştır. Fakat tezin bu bölümünde ise yine sezgisel optimizasyon algoritmalarıyla çoklu amaç fonksiyonları kullanılarak denetçi tasarımı yapılacaktır.

Burada hata fonksiyonuna ek olarak literatürden alınmış sistem kontrol performansına etki eden başka fonksiyonlarda kullanılarak optimizasyon süreci daha etkili hale getirilmiştir.

Bu bölümde Tezin 5. bölümünde sunulmuş olan Tabu Arama Algoritması çoklu amaç fonksiyonu ile optimizasyon yapabilecek yeteneğe ulaştırılıp kesir dereceli denetçi tasarımı yapılmıştır.

Diğer bir uygulamada ise Büyük Kırılma Büyük Çöküş (BK-BÇ) algoritması kullanılarak daha önce kullanılmamış ve bozucuları önleyen bir amaç fonksiyonuyla hata fonksiyonu beraber kullanılarak literatürden alınan bir matematiksel model için kesir dereceli denetçi tasarlanmıştır.

## 7.1. Geliştirilmiş Tabu Arama Algoritması ile Çoklu Amaç Fonksiyonu Kullanılarak Kesir Dereceli PID Optimizasyonu

Tabu Arama Algoritmasıyla KD denetçi tasarımı Tezin 5. bölümünde detaylarıyla anlatılmıştır. Tezin bu bölümünde ise önerilen TAA'nın algoritmik yapısı kullanılmaktadır. Yalnız bu çalışmada önerilen optimizasyon algoritmasında çoklu amaç fonksiyonuna göre optimizasyon yapabilecek şekilde düzenlenmiştir. Tezin bu bölümünde TAA'nın kurallarından bahsedilmeyerek doğrudan kullanılan çoklu amaç fonksiyonunun yapısında bahsedilmektedir. TAA'nın çoklu amaç fonksiyonuna göre KDPID'nin optimizasyon süreci sunulmaktadır.

Monje-Vinagre ve arkadaşları KDPID parametrelerini ayarlamak için 5 farklı eşitlik ve eşitsizliklerden oluşan kriterler önerilmiştir. Bu kriterler yardımıyla istenilen sonuçlar elde edilebilir. Bu çalışmada ise bu 5 farklı kriter TAA içerisinde amaç fonksiyonu olarak kullanılarak KDPID parametreleri elde edilmektedir.

### 7.1.1. Monje-Vinagre'nin Kesir Dereceli PID Tasarımında Kullanılan Kriterleri

Monje-Vinagre ve arkadaşları KDPID parametrelerini ayarlama için kullanılan bir yöntem sunmaktadır. Bu yöntem aşağıda ilgili denklemlerde verilmiş olan 5 kriterin sağlanması felsefesine göre hareket eder. Bu kriterler sisteminin geçiş frekansındaki genliği, faz marjı, model belirsizliğine karşı dayanıklılık, yüksek frekansantaki gürültülere karşı dayanıklılık ve hassasiyet fonksiyonu gibi yapıları sağlar. Böylece iyi bir kontrol performansı elde edilebilir. Sistemdeki matematiksel model  $G(s)$  ve kullanılan denetçi KDPID ( $C(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu$ ) 'dir. Monje-Vinagre ve arkadaşları aşağıdaki 5 farklı denklemi kullanarak kesir dereceli sistemin 5 parametresini ayarlamaktadır [106, 166-168].

$$\left| C(jw_{cg})G(jw_{cg}) \right|_{dB} = 0dB \quad (7.1)$$

$$(\text{Arg}(C(jw_{cg})G(jw_{cg}))) = -\pi + \Phi_{pm} \quad (7.2)$$

$$\left( \frac{d(\text{Arg}(C(jw_{cg})G(jw_{cg})))}{dw} \right)_{w=w_{cg}} = 0 \quad (7.3)$$

$$\left| T(jw) = \frac{C(jw_{cg})G(jw_{cg})}{1 + C(jw_{cg})G(jw_{cg})} \right| \leq A \text{ dB} \quad (7.4)$$

$$\left| S(jw) = \frac{1}{1 + C(jw)G(jw)} \right|_{dB} \leq A \text{ dB} \quad (7.5)$$

Denlemede bulunan  $w_{cg}$  geiş frekansını  $\Phi_{pm}$  ise gerekli faz payını göstermektedir. A deęeri  $w \geq w_t \text{ rad/s}$  frekansları arasındaki gürültü azaltma miktarı, B ise  $w \leq w_s \text{ rad/s}$  frekansları arasındaki hassasiyet fonksiyonudur.

### 7.1.2 Geliştirilmiş Tabu Arama Algoritmasının Çoklu Amaç Fonksiyonuyla Çoklu Parametre Optimizasyonu için Kullanılması

Klasik Tabu arama algoritmalarında tekli hareket ve tekli amaç fonksiyonu kullanılmaktadır. Bu çalışmada ise Tezin 5. bölümünde sunulmuş olan Tabu Arama Algoritması tabanlı optimizasyon yöntemiyle çoklu amaç fonksiyonu kullanılarak KDPID'nin parametreleri elde edilmiştir. Algoritma Denklem 7.6'da verilmiş olan 5 doğrusal olmayan kriterin ilgili denklemini sağlayıp sağlamadığı durumuna göre hareket etmektedir. Bu çalışmada durum sayıcısı kavramı tanımlanarak kullanılmıştır. Bu sayıcıya göre optimizasyon esnasında ilgili denetçi KDPID konfigürasyonu ilgili kriteri sağlıyorsa sayıcı bir artmaktadır.

Maksimum sayıya ulaşan çözüme iyi değer olarak kabul edilerek işleme devam edilmektedir. Kısacası diğer yöntemlerde sadece birim basamak cevabına göre hatanın düşüşü baz alınarak optimizasyon yapılırken bu yapıda ise hataya bakmanın yanı sıra ilk şart olarak her aday çözümün bu şartlardan kaç adet sağladığına bakılmaktadır. En yüksek sağlama sayısına sahip aday o iterasyonun en iyi çözümü olarak alınmaktadır. Tabi burada şöyle bir olasılığın gelme ihtimali söz konusudur. Durum sayıcısının değeri 2 veya daha fazla aday çözüm için eşit olması durumunda, eşit aday çözümler arasında birim basamak cevabına göre en küçük hata fonksiyonu değerine sahip aday çözüm yeni başlangıç değeri olarak alınmaktadır [126].

$$\begin{aligned}
 & \text{Saglama } f_1(x) : x_i \in R \\
 & \text{Saglama } f_2(x) : x_i \in R \\
 & \vdots \\
 & \text{Saglama } f_n(x) : x_n \in R
 \end{aligned} \tag{7.6}$$

Denklemden  $(i = 1, 2, \dots, k)$ . Denklem 7.6'da verilen tüm amaç fonksiyonları sağlandığında optimizasyon algoritması sonlandırılır.

Optimizasyon algoritmasının başlangıcında verilen maksimum iterasyon sayısınca algoritma arama yapmaktadır. Bu aramalar süresince eğer Denklem 7.6'da bu denklemlerin hepsini sağlayan konfigürasyon bulunmaması halinde ise en fazla denklemleri sağlayan ve en küçük hata fonksiyonuna yani en iyi birim basamak cevabını veren parametre konfigürasyonu ilgili optimizasyon döngüsünün en iyi değeri olarak belirlenmektedir.

Tezin 5. bölümünde anlatıldığı gibi optimizasyonu yapılacak parametreler  $x = [x_1, x_2, \dots, x_n]$  parametre vektörünü oluşturur. Her parametreye özgü hareketlerin miktarının bulunduğu vektör  $S = [s_1, s_2, s_3, s_4, \dots, s_n]$ 'dir. Her parametre sistemi farklı etkilediğinden parametrelerin hareket miktarları da farklı olmaktadır. Denklem 7.7'de ilgili problem için oluşturulan komşuluk matrisi gösterilmektedir.

$$N^n = \begin{bmatrix} x_1 + s_1 & x_2 & x_3 & \cdots & \cdots & \cdots & \cdots & x_k \\ x_1 - s_1 & x_2 & x_3 & \vdots & \vdots & \vdots & \vdots & x_k \\ \vdots & x_2 + s_2 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & x_2 - s_2 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & x_k + s_k \\ x_1 & x_2 & x_3 & \cdots & \cdots & \cdots & \cdots & x_k - s_k \end{bmatrix} \quad (7.7)$$

Verilen eşitlikler k adet parametrenin k adet amaç fonksiyonuna göre optimizasyonunu yapabilecek yetenektedir. Amaç fonksiyonunun değeri komşuluk matrisinin her satırı için hesaplanmaktadır. Aşağıda verilen kod ile bahsedilen durum gerçekleştirilmiştir [126].

```

for n = 1 : 2k
    for i = 1 : k
        if Equ(i) satisfies givencondition
            EC(n) = EC(n) + 1
        end
    end
end
end

```

$E^n$  sayıcı matrisidir. Komşuluk matrisindeki her bir satırın kaç adet amaç fonksiyonunu sağladığı değerlerin toplamını tutmaktadır.

$$E^n = \begin{bmatrix} EC(1) \\ EC(2) \\ \vdots \\ EC(2k) \end{bmatrix} \quad (7.8)$$

Denklemden  $EC(n)$ ,  $n = 1, 2, \dots, 2k$  dır. Yapı şöyle çalışır: komşuluk matrisinin her bir satırı sisteme uygulanarak kaç adet amaç fonksiyonunu sağladığı her satır için tespit edilerek  $E^n$  matrisine kaydedilir. Daha sonra  $E^n$  matrisindeki en büyük elemana sahip konfigürasyon en fazla amaç fonksiyonuna sahip parametre

konfigürasyonu anlamına gelmektedir. Bu değer bir sonraki iterasyonun başlangıç değeri olmakla beraber Tabu listesinde kaydedilir.

Tabu arama yöntemlerinin genel kuralı olarak her komşuluk listesi Tabu listesindeki elemanlarla karşılaştırılarak ortak olan elemanlar komşuluk matrisinden silinir. Böylece farklı parametre vektör uzayları kolaylıkla taranır. [126].

Tezin 5. bölümünde sunulduğu gibi tabu listesi optimize edilen parametre sayısının iki katı olacak şekilde sınırlandırılmıştır. Tabu listesi başlangıçta boşdur iterasyonlar geçtikçe dolmakta fakat 10 elemanı geçmemektedir. İlgili eleman 10 iterasyon sonunda serbest bırakılmaktadır. Denklem 7.9'da tabu listesi gösterilmektedir.

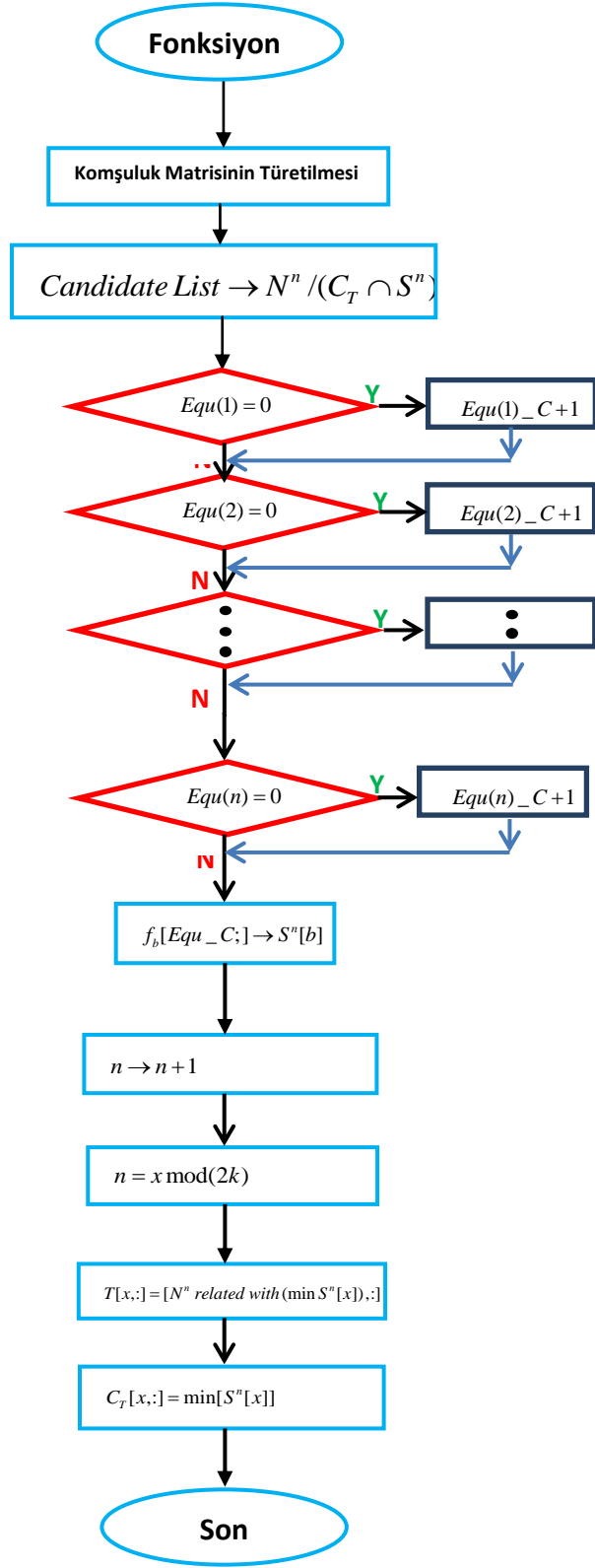
$$T = \begin{bmatrix} x_{11n} & x_{21n} & \cdots & x_{k1n} \\ x_{12n} & x_{22n} & \cdots & x_{k2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{12kn} & x_{22kn} & \cdots & x_{k2kn} \end{bmatrix} \quad (7.9)$$

$C_T$  aday listesidir. Bu listede komşuluk matrisi ile tabu listesinin optimizasyon süresince karşılaştırılmasıyla oluşturulur. Sayıcı matrisinin en yüksek değerine sahip eleman en iyi cevap üreten değer olmaktadır. Bu en iyi değer aynı zamanda her iterasyonda aspirasyon kriteriyle karşılaştırılmakta daha iyi bir değer bulunması halinde güncellenmektedir. Algoritma daha önceki örnekte de bahsedildiği gibi uzun süre sağlatabildiği amaç fonksiyonu sayısının sabit olması halinde, sabitleştiği değer aralığı alınarak algoritma başka hareket miktarlarıyla yeniden optimizasyon işlemine tabi tutulur. Eğer değerlerde iyileşme söz konusu ise belirlenen maksimum iterasyon sayısı kadar arama yapılmaktadır. Eğer bariz bir iyileşme yok ise en son bulunan değer algoritmanın sonucudur. Şekil 7.1'de önerilen optimizasyon algoritmasının akış grafiği gösterilmektedir.

### 7.1.3. Kesir Dereceli PID Denetçinin Çoklu Amaç Fonksiyonlu Geliştirilmiş Tabu Arama Algoritması ile Optimizasyonu

Bu bölümde çoklu amaç fonksiyonu olarak Monje–Vinagre'nin KDPID'ler için önermiş oldukları Denklem 7.1-7.5 aralığında verilen 5 adet doğrusal olmayan denklem kullanılmaktadır [106, 166-168]. Verilmiş olan bu 5 adet doğrusal olmayan denklemin optimizasyon süresince denenen denetçi konfigürasyonu için sağlayıp sağlamadığına bakılır. Eğer ilgili denklemini sağlaması halinde durum sayıcısı artırılır. En yüksek sayıda sağlayan konfigürasyon en iyi değer olarak kabul edilerek algoritma sürdürülür.

İlk olarak  $k_p, k_i, k_d, \lambda, \mu$  KDPID parametrelerine  $s_1, s_2, s_3, s_4, s_5$  şeklinde hareket miktarları tanımlanarak Denklem 5.2'de verilen komşuluk matrisi oluşturulur. Bir sonraki adımda ise komşuluk matrisindeki her satır için kaç adet doğrusal olmayan denklemini sağladığında bakılarak ilgili değerler  $E^n$  sayıcı matrisine kaydedilir. En fazla denklemini sağlayan konfigürasyon en iyi değer olarak alınıp bir sonraki iterasyonda başlangıç değeri olarak kullanılırken Tabu listesinde kaydedilir. Çünkü 1. iterasyondan sonra komşuluk matrisi öncelikle tabu listesiyle karşılaştırılıp aynı olan konfigürasyonlar silinmektedir. Algoritma bu kurallar çerçevesinde parametre taraması yapmaktadır. Algoritmayla alakalı tüm kurallar Tezin 5. Bölümünde verildiği için bu kısımda sadece farklılıklar üzerinde durulmuştur [126].



Şekil 7.1 Önerilen Çoklu Amaç Fonksiyonlu TAA'nın Akış Şeması

#### 7.1.4 Simülasyon Sonuçları

Bu bölümde literatürden alınmış iki örnek üzerinde optimizasyon algoritması belirtilen kurallar çerçevesinde uygulanmıştır [126].

##### 7.1.4.1 Simülasyon Örnek 1:

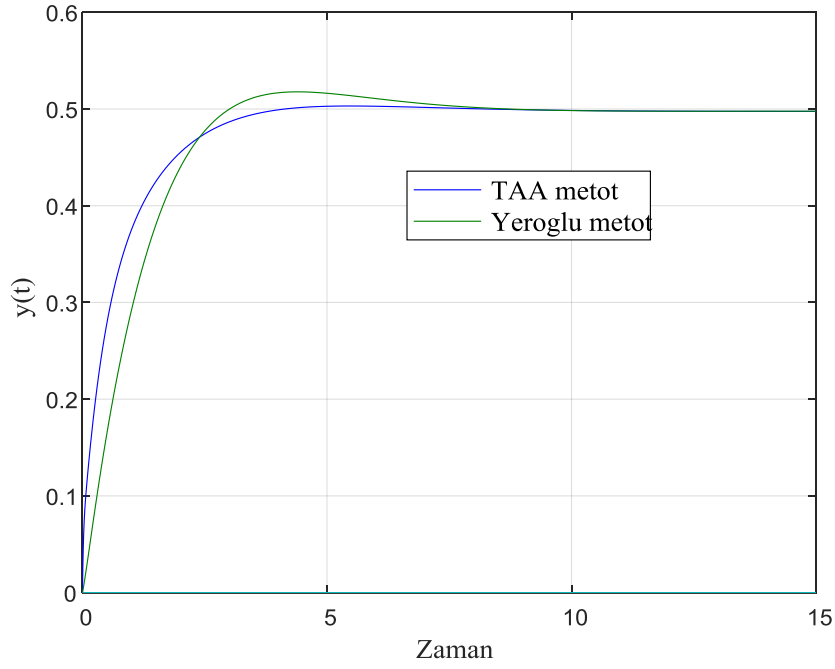
İlk örnek olarak 7.10 da verilen matematiksel model alınmaktadır.

$$G_1(s) = \frac{2}{80s + 1} \quad (7.10)$$

Sistemin özellikleri geçiş frekansı  $w_{cg} = 2.1$  rad/s, faz payı  $\Phi_{pm} = 80^\circ$ ,  $A = -20dB$ ,  $\omega \geq \omega_t = 10$  rad/s.  $B = -20dB$   $\omega \leq \omega_s = 0.01$  rad/s. şeklinde verilmiştir. 5 doğrusal olmayan denklemin değerleri verilmiş olan değerlere göre [106]'da verildiği gibi elde edilir. En fazla denklemi sağlatmaya dönük çalışan optimizasyon algoritması belirlenen iterasyon sayısı sonucunda denklem 7.11'de verilen denetçi konfigürasyonunu elde etmiştir.

$$C_1(s) = 4 + \frac{1.21}{s^{0.12}} + 24s^{0.55} \quad (7.11)$$

Şekil 7.2'de literatürde olan bir yöntem ile optimizasyon algoritmasının sonucu karşılaştırılmıştır. Görüldüğü üzere optimizasyon algoritması daha iyi sonuç elde edebilmektedir.



**Şekil 7.2**  $C1(s)G1(s)$  için birim basamak cevabı [106]

#### 7.1.4.2 Simülasyon Örnek 2:

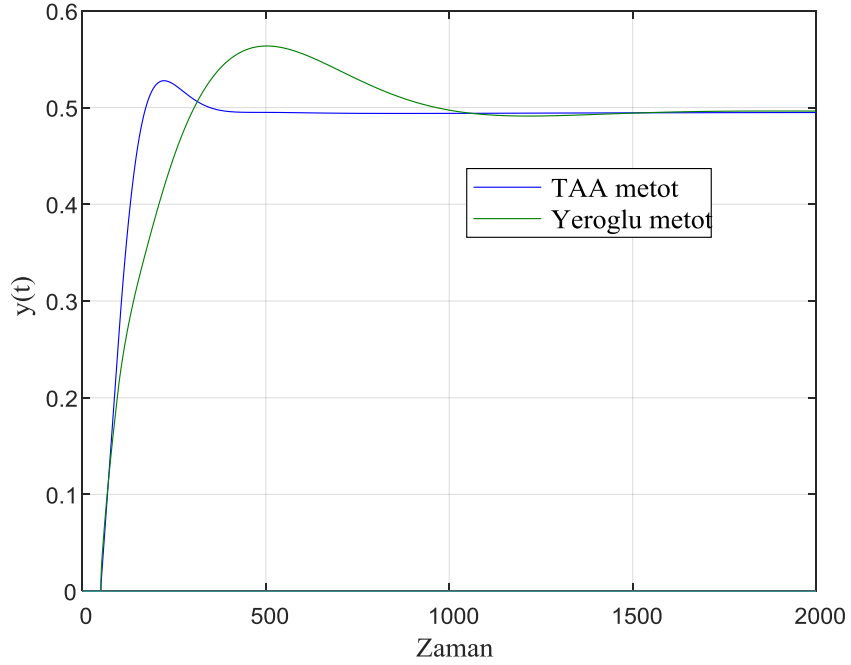
Örnek 2’de ise [171]’de alınmış olan zaman gecikmeli seviye kontrolü yapılan bir sistem verilmiştir.

$$G_2(s) = \frac{k}{\tau s + 1} e^{-Ls} = \frac{3.13}{433.33s + 1} e^{-50s} \quad (7.12)$$

Sistemin özellikleri; geçiş frekansı  $\omega_{cg} = 0.008$  rad/s, faz marjı  $\Phi_{pm} = 60^\circ$ ,  $A = -20dB$ ,  $\omega \geq \omega_t = 10$  rad/s.  $B = -20dB$   $\omega \leq \omega_s = 0.01$  rad/s. şeklindedir. İlk örnekteki aynı işlemlerin uygulanmasıyla Denklem 7.13’de verilen kesir dereceli PID denetçi elde edilir.

$$C_2(s) = 1.45 + \frac{1.12}{s^{0.97}} + 0.004s^{0.92} \quad (7.13)$$

Şekil 7.3’de elde edilen denetçi ile mevcut sonuç karşılaştırılarak elde edilen sonucun daha iyi birim basamak cevabı ürettiği gözlemlenmiştir.



Şekil 7.3 C2(s)G2(s) için birim basamak cevabı

## 7.2. Çoklu Amaç Fonksiyonlu Büyük Patlama Büyük Çöküş (BP-BÇ) Optimizasyon Algoritmasıyla Kesir Dereceli PID Denetçi Tasarımı

Tezin bu bölümünde ise Büyük Patlama Büyük Çöküş (BP-BÇ) optimizasyon algoritmasıyla tezin daha önceki bölümlerinde anlatılmış olan çok girişli çok çıkışlı helikopter prototipi olan TRMS sistemi için kesir dereceli denetçi tasarımı yapılmaktadır. Denetçi tasarlanırken genelde kullanılan hata fonksiyonu değerinin yanında sisteme bozucu uygulandığında bu bozucuları da engelleyebilen Referans Bozucu Oranı (RBO) (Referans Disturbance Ratio) kullanılmıştır [169].

Tanımlanan amaç fonksiyonunun şöyle bir farkı bulunmaktadır. Bilindiği üzere hata fonksiyonu amaç fonksiyonlarında minimizasyon şeklinde kullanılır.

Fakat RBO detayları takip eden bölümlerde verilen bir maksimizasyon problemidir. Bu problem için tanımlanan amaç fonksiyonu ile hem minimizasyon

hemde maksimizasyon problemi aynı anda çözülebilir. Bu yaklaşımda literatür açısından son derece önemli bir yaklaşımdır [169].

Günümüzde KDPID denetçiler klasik PID denetçilere nazaran daha iyi kontrol performansı ürettiklerinden dolayı birçok pratik uygulamada da kullanılır [170-176]. Tabi KD denetçinin bu kullanımı karşımıza ciddi bir optimizasyon probleminde beraberinde getirir.

Sezgisel optimizasyon algoritmalarının kontrol problemlerinde kullanılmasının önemli bir avantajı da çoklu parametre optimizasyonuna izin vermesidir.

Aslında denetçinin ayarlanaması problemi birçok düşük aşım, hızlı yerleşme, bozucuyu engelleme, dayanıklı karalılık gibi amaç fonksiyonlarını eş zamanlı olarak sağlatmaktır. Optimizasyon işleminin performansı birçok amaç fonksiyonunun birleştirilmesiyle geliştirilmektedir. Bu amaç fonksiyonları hata fonksiyonları yükselme zamanı, yerleşme zamanı, kalıcı hal hatası, kazanç payı, faz payıdır [169].

Literatür de çoklu amaç fonksiyonuyla optimizasyonun yapıldığı birçok çalışma bulunur. Örneğin, voltaj regülatör sistemi için KDPID denetçinin IAE, kalıcı hal hatası ve yerleşme zamanı amaç fonksiyonları kullanılarak optimize edilmektedir [177]. Aşım, kalıcı hal hatası, yükselme ve yerleşme zamanı gibi çoklu amaç fonksiyonları kullanılarak PI ve PID denetçinin parametrelerinin optimize edildiği çalışma [178]'de sunulmaktadır. Aşım, yükselme zamanı, oturma zamanı, kalıcı hal hatası, IAE hata yapısı girişin karesel integrali, kazanç payı ve faz payı çoklu amaç fonksiyonları kullanılarak parçacık sürü optimizasyon yöntemiyle KDPID parametreleri [27]'de tasarlanmaktadır. Yeni bir hibrid yöntem ile çoklu amaç fonksiyonları kullanılarak dağıtılmış KD amartisörün tasarımı [179]'da sunulmaktadır.

Tezin bu bölümünde ise BP-BÇ optimizasyon algoritmasıyla bozucuları engelleyen KDPID tasarlanmıştır. Tasarım esnasında MSE hatanın minimizasyonu ile referans bozucu oranının (RBO) maksimize edildiği çoklu amaç fonksiyonu kullanılmıştır.

Referans bozucu oranı (RBO) Alagöz ve arkadaşları tarafında [180,181]'de açıklanmaktadır. RBO oranı bu çalışmalarda PID ve KDPID parametreleri için

türetilmektedir. Bu oran KDPID ve klasik PID denetçi parametrelerini kullanarak hesaplanabilen matematiksel formülasyonları mevcuttur [180, 181].

Bu çalışmada da hata fonksiyonuyla birlikte bu oranda amaç fonksiyonunda kullanılarak birim basamak cevapları iyi olan ve bozucuları engelleyen KD denetçi tasarlanmıştır. RBO oranı maksimizasyon problemi olmasına karşın MSE ise minimizasyon problemidir. Tanımlanan amaç fonksiyonunda iki yapı birleştirilmiştir.

Özellikle uçuş kontrol sistemlerinde, hızlı yerleşme ve bozucuların engellenmesi son derece önemli kavramlardır. Bundan dolayı TRMS sisteminin ana motorunun matematiksel modeli kullanılarak KDPID denetçi elde edilmektedir. RBO oranı kontrol performansı için oluşturulduğundan hesaplamalar düşük frekans bölgesin için yapılmıştır. Bunun yanı sıra diğer çalışmalarda olduğu gibi bu çalışmada da kesir dereceli denetçi Valerio'nun araç kutusu kullanılarak gerçekleştirilmiştir [56, 162, 182, 183]. Birim basamak cevapları bu araç kutusu yardımıyla MATLAB simulink üzerinden elde edilmiştir.

### 7.2.1 Çoklu Amaç Fonksiyonu ile Denetçi Parametrelerinin Ayarlanması

Pratik kontrol uygulamalarında sistemin zaman cevabı için iki önemli kriter vardır. Bunlar ayar noktası bozucuları engelleme performansdır. Ayar noktası performansı Denklem 7.14'de verilen MSE değerine göre elde edilebilir.

$$E_{MSE} = \frac{1}{n} \sum_{i=1}^n (e(t))^2 \quad (7.14)$$

Bozucuları engelleme performansı için ise [180, 181]'de verilmiş olan RBO'nın kullanılmasıyla elde edilir. Sistemin bozucuları engelleme performansını arttırmak için RBO oranının maksimize edilmesi gerekir [180, 181]. Kapalı döngü PID ve KDPID için RBO değeri denklem 7.15 de gösterilmektedir [180, 181].

$$RBO(\omega) = \left( (k_p + k_i \cos(\frac{\pi}{2})\lambda)\omega^{-\lambda} + k_d \cos(\frac{\pi}{2})\omega^\mu \right)^2 + \left( \left( k_d \sin(\frac{\pi}{2})\omega^\mu - k_i \sin(\frac{\pi}{2})\omega^{-\lambda} \right)^2 \right)^{\frac{1}{2}} \quad (7.15)$$

Kontrol sistemleri düşük frekans bölgesinde çalıştığından RBO oranı düşük frekans bölgesi için Denklem 7.16'deki gibi hesaplanır.

$$RBO_L = \sum_{\omega_i \in (0, \omega_{\max})} RBO(\omega_i) \quad (7.16)$$

Burada, MSE ve RBO kullanılarak çoklu amaç fonksiyonunun tekrar hesaplanması gerekir. Bu çalışmada RBO maksimizasyon problemidir. RBO oranının çarpma işlemine göre tersinin alınmasıyla minimizasyon problemine dönüştürülerek Denklem 7.17'de verilen yeni çoklu amaç fonksiyonu elde edilir.

$$J = \min \{ c_1 E_{MSE} + c_2 RBO_L^{-1} \} \quad (7.17)$$

Denkleimde  $c_1$  ve  $c_2$  değerleri ağırlık katsayılarıdır. Kontrol performansına ve sistemin durumuna göre optimizasyon başlangıcında ayarlanması gerekir.

### 7.2.2 Büyük Patlama Büyük Çöküş Optimizasyon Algoritması

Büyük Patlama Büyük Çöküş algoritması Erol ve Eksin tarafından önerilmiş sezgisel bir optimizasyon algoritmasıdır [20].

Bu algoritma büyük patlama ve büyük çöküş şeklinde iki adet seviyeden oluşmaktadır. Algoritmanın rassal olarak aday çözümleri oluşturmuş olduğu safha büyük patlamadır [184]. Algoritmanın bu aday çözümleri kullanarak en küçük maliyet fonksiyonuna sahip olan aday çözümü bulduğu ve aday çözümlerin kütle merkezini bulduğu safha ise büyük çöküştür. Algoritmanın optimizasyon adımları aşağıda verilmektedir [20]:

**Büyük patlama büyük çöküş optimizasyon fonksiyonunun adımları:**

**Adım 1:** Belirlenen parametre arama uzayında aday çözümlerin rassal olarak oluşturulması.

**Adım 2:** Tüm aday çözümlerin maliyet fonksiyon değerinin hesaplanması

**Adım 3:** Denklem 7.18'e göre aday çözümlerin ağırlık merkezinin bulunması.

$$x_c = \frac{\sum_{i=1}^N \frac{1}{f_i} x_i}{\sum_{i=1}^N \frac{1}{f_i}} \quad (7.18)$$

Denklemde  $x_i$  n yönlü uzayda oluşturulan aday çözüm,  $f_i$  ilgili aday çözümün maliyet fonksiyonu değeri, N Optimizasyon algoritmasının popülasyon sayısını gösterir.

**Adım 4:** Yeni aday çözümün  $x_{new} = x_c + lr/k$  formülü yardımıyla hesaplanması.  $x_c$  Denklem 7.18'de hesaplanan kütle merkezi değeri. L ilgili parametrenin üst limiti, r normal rassal sayı ve k ilgili iterasyonun sayısıdır.

**Adım 5:** Durdurma kriteri sağlanıncaya kadar Adım 2'ye dön. (Bu tarz algoritmalarda durdurma kriteri başlangıçta belirlenen maksimum iterasyon sayısıdır.)

### **7.2.3 Büyük Patlama Büyük Çöküş Optimizasyon Algoritması ile Çoklu Amaç Fonksiyonu Kullanılarak Kesir Dereceli PID Denetçinin Tasarımı**

Bu bölümde Denklem 7.17 tanımlanmış çoklu amaç fonksiyonu ve BP-BÇ optimizasyon algoritması kullanılarak kesir dereceli denetçi tasarlanmıştır.

RBO referans sinyalin enerjisini hesaplayan bir orandır. Bu oran sistem çıkışındaki bozucu sinyalin enerjisinin engeller. Bu da kapalı döngü kontrol sistemleri için iyi bir kontrol performansı üretmesi anlamına gelir [181]. MSE ise sistemin ayar noktası performansını iyileştirmektedir. Sistemin yükselme ve yerleşme zamanları, kontrol performansı açısından iyi değerler almasını sağlar.

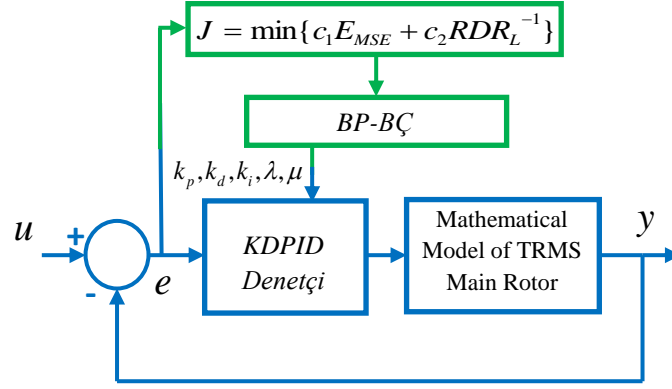
Bu çoklu amaç fonksiyonunda minimizasyon problemi olan MSE ile maksimizasyon problemi olan RBO eş zamanlı çözümlenir. Yani minimizasyon

ve maksimizasyon problemlerini aynı anda çözümlenmesine imkan sağlayan bir amaç fonksiyonudur.

Bilindiği gibi uçuş kontrol sistemleri uçuş esnasında çevresel birçok bozucuya maruz kalmaktadır. RBO performansının iyileştirilmesi uçuş kontrol sistemleri için son derece önem arz etmektedir. Bu çalışmada daha önceki bölümlerde detayı verilmiş olan TRMS uçuş kontrol sisteminin ana motorunun matematiksel modeli kullanılmıştır. Bu modele tanımlanan çoklu amaç fonksiyonuyla kesir dereceli PID denetçi tasarlanmıştır. Bu sayede iyi bir birim basamak cevabı üretilirken bozucuları da engelleyen bir kontrol döngüsü elde edilmiştir.

Önerilen sistemin blok şeması Şekil 7.4'de gösterilmektedir. Bu şekle göre BP-BÇ optimizasyon algoritması tanımlanan ve Denklem 7.17'de verilmiş olan çoklu amaç fonksiyonuna göre optimizasyonla KDPID denetçi parametrelerini elde etmektedir. MSE, TRMS ana motorunun birim basamak cevabına göre hesaplanırken RBO oranı ise Denklem 7.16'da verilen formül ile düşük frekans bölgesinde ( $\omega$ ) frekans aralığı [0-10] alınarak hesaplanır.

BP-BÇ optimizasyon algoritmasının kesir dereceli PID denetçi katsayıları olan  $k_p, k_d, k_i, \lambda$  ve  $\mu$  için aday çözümleri normal rassal bir adım ile temel matematiksel operatörler olan toplama çıkarma çarpma ve bölme işlemlerini kullanarak elde etmektedir. Örneğin  $k_p$  değişkeni için 4 adet yeni aday çözüm söz konusudur. Bunlar  $k_p + rand$ ,  $k_p - rand$ ,  $k_p / rand$  ve  $k_p * rand$  şeklindedir. Bu şekilde aday listesi oluşturma bu Tezin 6. bölümünde detaylarıyla sunulmuştur. Bu çalışmada da BP-BÇ optimizasyon algoritması [1]'de verilen temel optimizasyon algoritmasıyla modifiye edilmektedir. Bu sayede çok farklı parametre vektör uzayları taranabilir.



**Şekil 7.4** Optimizasyonla KDPID denetçi parametrelerinin ayarlandığı sistem modeli

***Çoklu amaç fonksiyonu ile BP-BÇ optimizasyon algoritmasının adımları:***

**Adım 1:** Başlangıç  $k_p, k_d, k_i, \lambda$  ve  $\mu$  parametre değerlerinin belirlenmesi ve rassal bir hareket ile temel matematiksel operatörler olan toplama(+) çıkarma(-) çarpma(\*) bölmeye(/) göre aday listesinin belirlenmesi.

**Adım 2:** Denklem 7.17' ya göre amaç foksiyonun hesaplanması.

**Adım 3:** Denklem 7.18'e göre aday çözümlerin kütle merkezinin bulunması. Denklem 7.19'da  $k_p$  değeri için kütle merkezinin bulunduğu formül verilmektedir.

$$x^{k_p}_c = \frac{\sum_{i=1}^N \frac{1}{f_i} k_{p_i}}{\sum_{i=1}^N \frac{1}{f_i}} \quad (7.19)$$

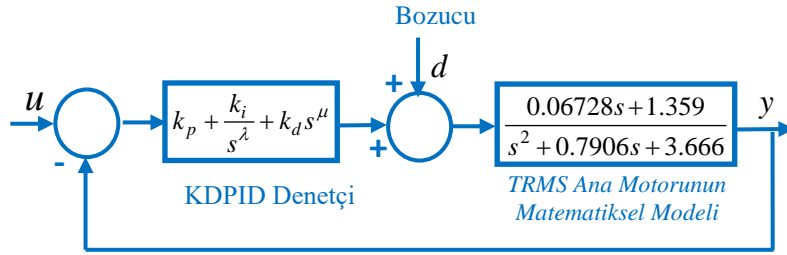
**Adım 4:**  $x_{new} = x_c \pm lr/k$  formülüyle yapılan hesaplamalar yardımıyla ilgili parametrenin yeni değeri hesaplanmaktadır. Denklemdaki  $l$  ilgili parametrenin maksimum değerini,  $r$  normal rassal değişkeni  $k$  ise bulunan iterasyon sayısını göstermektedir. Örneğin  $k_p$  değişkeni için yeni değer  $x_{k_p} = x_c^{k_p} \pm lr/k$  formülüyle hesaplanmaktadır.

**Adım5:** Maksimum iterasyon sayısına ulaşıncaya kadar 2. Adıma dön.

## 7.2.4 Simülasyon Sonuçları

Bu bölümde simülasyon sonuçları klasik PID ve KDPID'nin birim basamak cevapları karşılaştırılarak verilmektedir. TRMS'nin matematiksel modeli [82]'de elde edilmektedir. Birim basamak cevaplarının alındığı TRMS ana motorunun modelinde içerisinde bulunduğu simülasyon modeli Şekil 7.5'de verilmektedir.

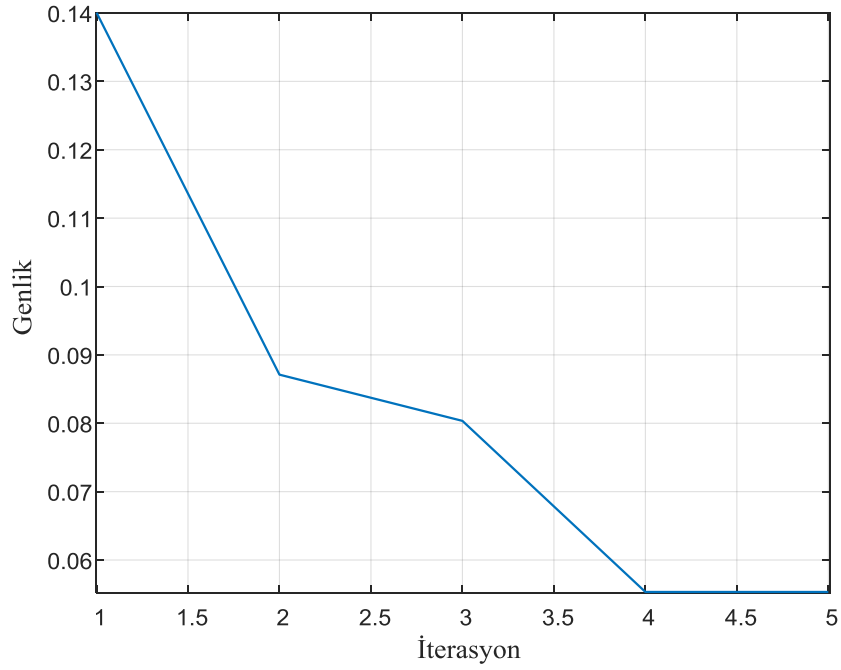
Test esnasında başlangıçta birim basamak sinyali giriş olarak uygulanmaktadır. Simülasyonun zamanının ortasında ise sisteme birim basamak bozucu uygulanarak sonuçlar irdelenmiştir. Böylece amaç fonksiyonunda kullanılan tüm ifadelerin kontrol performansına olan etkileri tespit edilebilir. Bu çalışmada da CFE yaklaşımı KDPID gerçekleştirilmede kullanılmıştır. Bu yaklaşımın MATLAB simulinkteki verisyonu olan Valerio araç kutusu kullanılmaktadır [64].



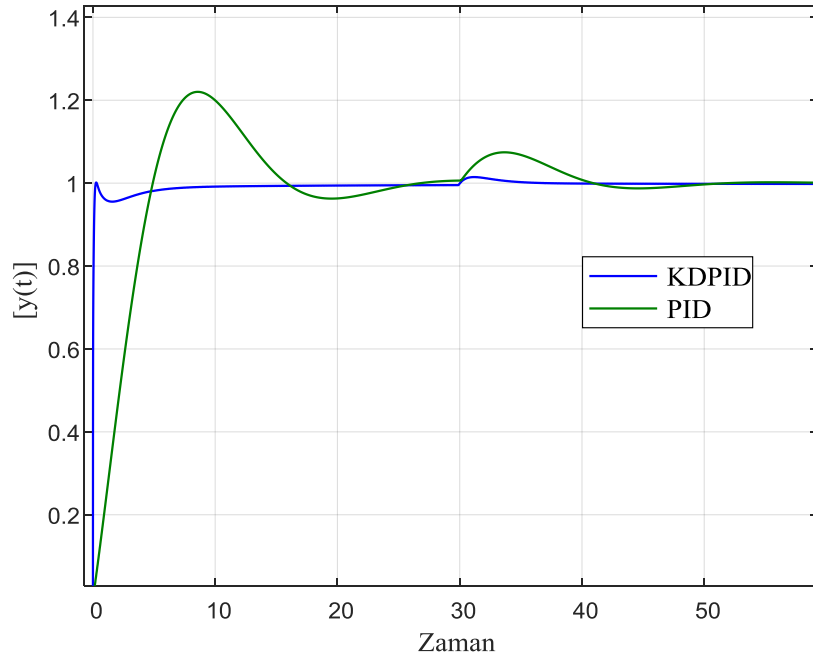
**Şekil 7.5** TRMS sisteminin ana motorunun bozucu sinyalinin KDPID ile kontrol edildiği sistem şeması

Şekil 7.6'da optimizasyon süresince Denklem 7.17'ye göre hesaplanan çoklu amaç fonksiyonunun değişimi gösterilmiştir. Bu amaç fonksiyonunun sürekli azalan eğilimde olması algoritmanın çalıştığına en önemli göstergesidir. Şekil 7.7'de ise çoklu amaç fonksiyonuna göre BP-BÇ optimizasyon algoritmasıyla türetilmiş klasik PID ve KDPID denetçilerin birim basamak cevaplarının karşılaştırılması verilmektedir. Şekilden de anlaşılacağı üzere KDPID denetçi daha iyi birim basamak cevabı üretmektedir. KDPID ve PID denetçilerin değerleri Çizelge 7.1'de sunulmaktadır. KDPID denetçi daha iyi ayar noktasına sahiptir. Bunun yanı sıra bozucuları da etkili bir şekilde engellemektedir. KDPID denetçi bozucunun negatif etkisini önemli ölçüde

azaltır. Sonuç olarak KDPID denetçi klasik PID göre daha dayanıklı bir kontrol performansı üretmektedir.

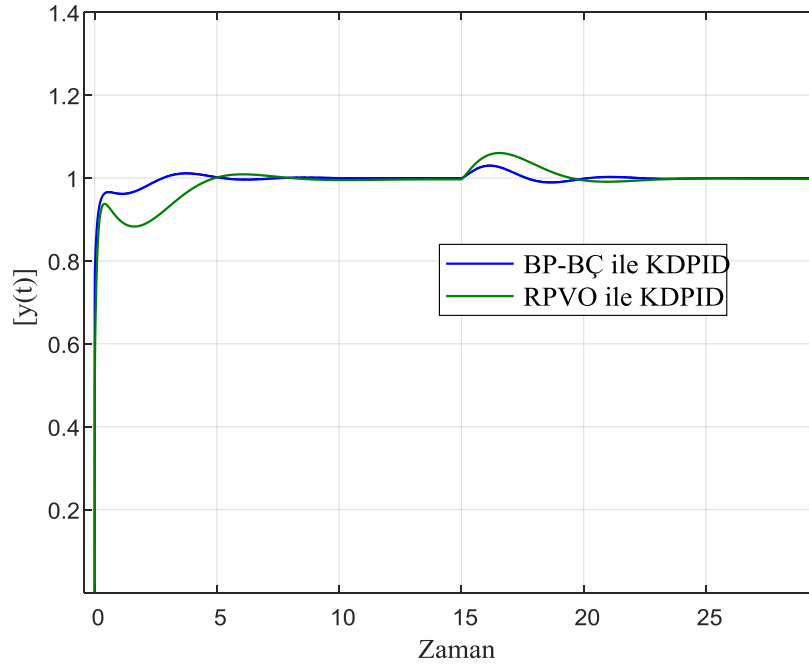


Şekil 7.6 KDPID denetçi optimizasyonu süresince maliyet fonkiyonunun değişimi



Şekil 7.7 KDPID ile Klasik PID denetçilerin birim basamak cevaplarının karşılaştırılması

Şekil 7.8’de ise aynı matematiksel model için başka bir optimizasyon algoritması olan RPVO yöntemi ile elde edilmiş KDPID denetçi yapısıyla BP-BÇ optimizasyon algoritmasıyla elde edilmiş sonuçlar karşılaştırılmıştır. RPVO ile elde edilen kesir dereceli denetçi katsayıları Çizelge 7.1 de verilmiştir. Aslında birim basamak cevapları iki yöntemde de yakınsamaktadır. Fakat BP-BÇ optimizasyon algoritmasında çoklu amaç fonksiyonu kullanıldığından dolayı bozucuları RPVO algoritmasına göre daha iyi engellemektedir. Çünkü RPVO algoritmasında sadece hata fonksiyonu amaç fonksiyonu olarak kullanılmıştır. Buradan da anlaşılacağı üzere amaç fonksiyonunda kullanılan denklemlerin cinsi kontrol performansını önemli ölçüde etkilemektedir.



Şekil 7.8 BP-BÇ ve RPVO algoritmalarıyla türetilmiş olan KDPID denetçilerin birim basamak cevapları

**Çizelge 7.1:** BP-BÇ ve RPVO algoritmalarıyla elde edilen denetçi katsayıları

	$k_p$	$k_i$	$k_d$	$\lambda$	$\mu$
<b>Çoklu amaç Fonksiyonu ve BP-BÇ ile KDPID</b>	27.46	18.04	26.82	0.79	0.89
<b>RPVO ile Kesir dereceli PID [82]</b>	9.90	7.78	11.89	1.0053	1.07
<b>Çoklu amaç Fonksiyonu ve BP-BÇ ile Klasik PID</b>	14.23	11.97	6.93	0	0

## 8. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında öncelikle yarı analitik ve yarı nümerik bir yöntem olan HPM kullanılarak kesir dereceli integro diferansiyel ifadelerin çözümü sunulmuştur. Bu yöntem ile kesir dereceli denetçilerle oluşturulmuş olan kontrol sistemlerinde denetçinin belirlenmesi probleminin çözümüne ilişkin bir yöntem gösterilmiştir. Elde edilen sonuçlar karşılaştırılmıştır. Sezgisel optimizasyon algoritmalarıyla uçuş kontrol sistemine ve başka sistem modellerine kesir dereceli ve tam sayı dereceli denetçiler tasarlanmıştır.

Öncelikle literatürde olmayan fakat literatürdeki yöntemlerden faydalanılarak geliştirilen RPVO yöntemi önerilmiştir. Bu yöntem TRMS sisteminin simülasyon ve gerçek zamanlı modelleri üzerinden denenerek sonuçlar karşılaştırmalı olarak irdelenmiştir.

TAA algoritması KD denetçi parametrelerini optimize edebilecek kabiliyete ulaştırılarak literatürde olan sistem modelleri için kesir dereceli denetçiler tasarlanmıştır.

YFO algoritması tam sayı dereceli ve kesir dereceli denetçileri optimize edebilecek şekilde yeniden kodlanarak GYFO elde edilmiştir. Elde edilen bu yapı ile sistemlere kesir dereceli ve tam sayı dereceli denetçiler tasarlanmıştır. Algoritmanın çalışma performansına etki eden yapılar optimizasyon esnasında incelenerek ilgili model için en uygun konfigürasyonlar belirlenmiştir.

Tezin önceki bölümlerinde optimizasyon esnasında tekli amaç fonksiyonu olan hata fonksiyonu kullanılmıştır. Tezin son bölümünde ise kontrol performansını artırmak için hata fonksiyonuna başka amaç fonksiyonları da eklenerek optimizasyonla çoklu amaç fonksiyonuna göre optimizasyon işlemi gerçekleştirilmiştir.

Bu tez çalışmasında sezgisel optimizasyon algoritmalarıyla karmaşık denetçi problemlerinin de çözülebileceği gösterilmiştir.

Bu tez çalışmasının çıktılarıyla hazırlanan çalışmalar birçok indexe giren dergide makale, alan indexi dergilerinde makale, ulusal dergilerde makale ve uluslararası konferanslarda sözlü sunumlu basılan bildiri olarak yayımlanmıştır.

**Bu tez çalışmasının sonuçlarında yararlanılarak şu çalışmalar yapılabilir.**

- Uygulama yapılan sezgisel algoritmalar başka programlama dillerinde kodlanarak ilgili sistem donanımının üstüne bilgisayardan bağımsız olarak yüklenerek çalıştırılabilir. Böylece gerçek zamanlı sistem üzerinde bilgisayardan ayrı bir şekilde çalışabilen algoritmalar elde edilebilir.
- Sezgisel algoritmaların başka zor optimizasyon problemlerinde kullanılabilir. Örneğin sistem modelleme, güzergah planlama, proses kontrol v.b.
- Sezgisel algoritmaların esneklik avantajı ile analitik yöntemlerin kesinlik avantajından faydalanılarak hibrid etkin optimizasyon yöntemleri önerilebilir.

## 9.KAYNAKLAR

- [1] S. A. Salem, BOA: A novel optimization algorithm, International Conference on Engineering and Technology, Cairo, Egypt, Oct. 10-11, (2012), pp. 1-5.
- [2] W.M. Spears, D.F. Spears, J.C. Hamann and R. Heil, *Distributed, physics-based control of swarms of vehicles*, **Autonomous Robots**, 17:2-3 (2004), pp.137-162.
- [3] P.Y. Chevalier, J.M. Hendrickx and R.M. Jungers, *Efficient Algorithms for the Consensus Decision Problem*, **SIAM J. Control Optim.**, 53:5 (2014) pp. 3104–3119.
- [4] S. Panda, B.K. Sahu and P.K. Mohanty, *Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization*, **Journal of the Franklin Institute**, 349:8 (2012) pp. 2609-2625.
- [5] D. Wachsmuth and J.E. Wurst, *An interior point method designed for solving linear quadratic optimal control problems with hp finite elements*, **Optimization Methods and Software**, 30:6 (2015), pp. 1276-1302.
- [6] T. Weise, *Global Optimization Algorithms Theory and Application*. Self Published, Boston, USA, 2008, p. 1-572.
- [7] N. Kang, M. Kokkolaras and P.Y. Papalambros, *Solving multiobjective optimization problems using quasi-separable MDO formulations and analytical target cascading*, **Structural and Multidisciplinary Optimization**, 50:5 (2014) pp. 849-859.
- [8] S.D. Mohammadi and M. Rais-Rohani, *Exponential penalty function formulation for multilevel optimization using the analytical target cascading framework*, **Structural and Multidisciplinary Optimization**, 47:4 (2013) pp. 599-612.
- [9] A. Kaveh and S. Talatahari, *An improved ant colony optimization for constrained engineering design problems*, **Engineering Computations**, 27:1 (2010) pp. 155-182.
- [10] M.S. Bazaraa, H.D. Sherali and C.M. Shetty, *Nonlinear programming: theory and algorithms*. John Wiley & Sons, New Jersey, USA, 2013, p. 1-843.
- [11] R. Misener, J.B. Smadbeck and C.A. Floudas, *Dynamically generated cutting planes for mixed-integer quadratically constrained quadratic programs and their incorporation into GloMIQO 2*, **Optimization Methods and Software**, 30:1 (2015) pp. 215-249.
- [12] M.H. Lin and J. Tsai, *A deterministic global approach for mixed-discrete structural optimization*. **Engineering Optimization**, 46:7 (2014) pp. 863-879.

- [13] W.M. Spears and D.F. Gordon, Using artificial physics to control agents, International Conference on Information Intelligence and Systems, Bethesda, MD, USA, 31 Oct. 31- Nov. 3, (1999), pp. 281-288.
- [14] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publisher, USA, 1997, p. 2093-2229.
- [15] L.J. Fogel, A.J. Owens and M.J. Walsh, *Artificial Intelligence Through Simulated Evolution*, Wiley, Chichester, 1966, p. 1-170.
- [16] M. Dorigo, V. Maniezzo, and A. Coloni, *The ant system: optimization by a colony of cooperating agents*, **IEEE Trans. Syst. Man Cybern.**, 26:1 (1996) pp. 1-13.
- [17] J.H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975, p. 1-194.
- [18] R.C. Eberhart and J. Kennedy, A new optimizer using particle swarm theory *In Proceedings of the sixth international symposium on micro machine and human science*, Oct. 4-6, (1995), Nagoya, Japan, pp. 39-43.
- [19] S. Kirkpatrick, C. Gelatt and M. Vecchi, *Optimization by simulated annealing*, **Science**, **220:4598** (1983) pp. 671–680.
- [20] O.K.Erol and I. Eksin, *New optimization method: Big Bang-Big Crunch*, **Adv. Eng. Softw.**, 37:2 (2006) pp. 106–111.
- [21] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, *GSA: a gravitational search algorithm*, **Inf. Sci.**, **179:13** (2009) pp. 2232–2248.
- [22] D. Karaboga, B. Basturk, *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm*, **Journal of global optimization**. **39:3** (2007) pp. 459-471.
- [23] D. DasGupta, *An overview of artificial immune systems and their applications*, Springer, Berlin Heidelberg, Germany, 1999, p. 3-21.
- [24] M. Abdechiri, M.R. Meybodi and H. Bahrami, *Gases Brownian motion optimization: an algorithm for optimization (GBMO)*. **Applied Soft Computing**, **13:5** (2013) pp. 2932-2946.
- [25] C.F. Juang, C.M. Lu, C. Lo and C.Y. Wang, *Ant colony optimization algorithm for fuzzy controller design and its FPGA implementation*, **Transactions on Industrial Electronics**, **55:3** (2008) pp. 1453-1462.
- [26] E.A. Gargari, F. Hashemzadeh, R. Rajabioun and C. Lucas, *Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process*, **International Journal of Intelligent Computing and Cybernetics**, **1:3** (2008) pp. 337-355.

- [27] M. Zamani, M.K. Ghartemani, N. Sadati and M. Parniani, *Design of a fractional order PID controller for an AVR using particle swarm optimization*, **Control Engineering Practice**, **17:12** (2009) pp. 1380-1387.
- [28] T.H. Huynh, A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers, *International Conference on Industrial Technology*, Apr. 21-24, (2008), Chengdu, China, p.1-6.
- [29] M.W. Iruthayarajan and S. Baskar, *Evolutionary algorithms based design of multivariable PID controller*, **Expert systems with Applications**, **36:5** (2009) pp. 9159-9167.
- [30] A. Biswas, S. Das and A.D. Abraham, *Design of fractional-order PI $\lambda$ D $\mu$  controllers with an improved differential evolution*, **Engineering applications of artificial intelligence**, **22:2** (2009) pp. 343-350.
- [31] N. Thomas and P. Poongodi, *Position control of DC motor using genetic algorithm based PID controller*, **Proceedings of the World Congress on Engineering**, 2 (2009) pp. 1-3.
- [32] C.H. Lee and F.K. Chang, *Fractional-order PID controller optimization via improved electromagnetism-like algorithm*, **Expert Systems with Applications**, **37:12**, (2010) pp. 8871-8878.
- [33] W. Sheng and Y. Bao, *Fruit fly optimization algorithm based fractional order fuzzy-PID controller for electronic throttle*, **Nonlinear Dynamics**, **73:1-2** (2013) pp. 611-619.
- [34] E.S. Ali and S.M. Abd-Elazim, *Bacteria foraging optimization algorithm based load frequency controller for interconnected power system*, **International Journal of Electrical Power & Energy Systems**, **33:3** (2011) pp. 633-638.
- [35] D.S.C. Leandro and V.C. Mariani, *Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning*, **Computers & Mathematics with Applications**, **64:8** (2012) pp. 2371-2382.
- [36] G. Yan and C. Li, *An effective refinement artificial bee colony optimization algorithm based on chaotic search and application for pid control tuning*, **J. Comput Inf Syst.**, **7:9** (2011) pp: 3309-3316.
- [37] E.S. Ali, *Optimization of power system stabilizers using BAT search algorithm*, **International Journal of Electrical Power & Energy Systems.**, **61** (2014) pp. 683-690.
- [38] S.A. Taher, M.H. Fini and S.F. Aliabadi, *Fractional order PID controller design for LFC in electric power systems using imperialist competitive algorithm*, **Ain Shams Engineering Journal**, **5:1** (2014) pp. 121-135.

- [39] S. Berrazouane and K. Mohammedi, *Parameter optimization via cuckoo optimization algorithm of fuzzy controller for energy management of a hybrid power system*, **Energy Conversion and Management**, 78 (2014) pp. 652-660.
- [40] R. Caponetto, G. Dongola, L. Fortuna, and I. Petras, *Fractional Order Systems Modeling and Control Applications*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2010, p.1-177.
- [41] I. Podlubny, *Fractional Differential Equations*, *Mathematics in Science and Engineering*, Academic Press, New York, USA, 1999, p.1-337.
- [42] K.B. Oldham and J. Spanier, *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*, Academic Press, New York, USA, 1974, p. 1-225.
- [43] B. Ross, *Fractional Calculus and its Applications*, Springer-Verlag, Berlin, Germany, 1975, pp. 1-36.
- [44] J. Sabatier, O.P. Agrawal and J.A. Machado, *Advances in Fractional Calculus: Theoretical Developments and Applications in Physics and Engineering* Springer, Dordrecht, Netherlands, 2007, p. 551.
- [45] A.A. Kilbas, H.M. Srivastava and J.J. Trujillo, *Theory and Applications of Fractional Differential Equations*, Elsevier, Amsterdam, Netherland, 2006, p. 1-540.
- [46] S. Das, *Functional Fractional Calculus for System Identification and Controls*, Springer, New York, USA, 2008, p.1-239.
- [47] I. Podlubny, *Fractional-Order Systems and PID Controllers*, **IEEE Transactions on Automatic Control**, 44:1 (1999) pp. 208–214.
- [48] C. Yeroglu and N. Tan, *Classical controller design techniques for fractional order case*, **ISA Trans.**, 50:3 (2011) pp. 461-472.
- [49] B. Şenol, C. Yeroglu, N. Tan, Kesir Dereceli Kontrol Sistemlerinin Analizi İçin Kolay Kullanımlı Program, Otomatik Kontrol Ulusal toplantısı TOK11, Eylül 2011, İzmir, Türkiye, pp.1-6.
- [50] B. Şenol, *Kesir Dereceli Sistemlerin Frekans Cevaplarının Analizi İçin Matlab Ortamında Toolbox*, Yüksek Lisans Tezi, İnönü Üniversitesi F.B.E., Malatya. Türkiye, 2011.
- [51] M. Kano, K. Tasaka, M. Ogawa, S. Ootakara, A. Takinami, S. Takahashi and S. Yoshii, Practical Direct PID/I-PD Controller Tuning and Its Application to Chemical Processes, IEEE International Conference on Control Applications, Sep. 8-10, 2010, Yokohama, Japan, pp. 2426-2431.

- [52] Á. Cuenca, J. Salt, A. Sala and R. Pizá, *A Delay-Dependent Dual-Rate PID Controller Overan Ethernet Network*. **IEEE Transactions on Industrial Informatics**, **7:1** (2011) pp. 18-29.
- [53] F.J.V. Parada, J.A.O. Tapiaand and J.A. Ramirez, *Effective Medium Equations for Fractional Fick Slaw in Porousmedia*, **Physica A**, 373 (2007) 339-353.
- [54] P. Arena, R. Caponetto, L. Fortunaand and D. Porto, *Nonlinear Non-integer Order Circuits and Systems- An Introduction*, World Scientific, Singapore, 2000- p.1-203.
- [55] M.F. Silva, J.A.T. Machadoand and A.M. Lopes, *Fractional order control of a hexapod robot*. **Nonlinear Dynamics**, **38:1-4** (2004) pp. 417-433.
- [56] B.M. Vinagre, Y.Q. Chen and I. Petras, *Two direct Tustin discretization methods for fractional-order diferentiator/integrator*, **J. Franklin Institute**, 340:5 (2003) pp. 349-362.
- [57] M.B. Milovanović, D.S. Antić, M.T. Milojković, S.S. Nikolić, S.L. Perić and M.D. Spasić, *Adaptive PID control based on orthogonal endocrine neural Networks*, **Neural Networks**, 84 (2016) pp. 80-90.
- [58] W.K. Ho, T.H. Lee, H.P. Han and Y. Hong, *Self-tuning IMC-PID control with interval gain and phase margins assignment*, **IEEE Transactions on Control Systems Technology**, **9:3** (2001) pp. 535-541.
- [59] T.K. Liuand and J.G. Juang, *A SingleNeuron PID Control for Twin Rotor MIMO System*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics Suntec Convention and Exhibition Center, July 14-17, (2009), Singapore, pp.186-191.
- [60] T. Machado, M. Da Graca Marcos, F. Duarte, *Fractional Dynamics in the Trajectory Control of Redundant Manipulators*, **Communications in Nonlinear Science and Numerical Simulation**, **13:9** (2008) pp. 1836–1844.
- [61] B.M. Vinagre, C.A. Monje, V. Feliu and Y.Q. Chen, *On Auto-Tuning of Fractional Order Controllers*, IFAC Workshop on Fractional Differentiation and its Application (FDA'06), July 19-21, (2006), Porto, Portugal, pp.1-6.
- [62] C. Yeroglu and N. Tan, *Note on fractional order proportional-integral-differential controller design*, **IET Control Theory and Applications**, **5:17** (2011) pp. 1978-1989.
- [63] A. Oustaloup, *La commande CRONE: commande robuste d'ordre non entier*. Hermès, Paris, 1991, p.1-495.
- [64] D. Valerio, (2005). Ninteger v.2.3 Fractional Control Toolbox for MATLAB, <http://web.ist.utl.pt/~duarte.vale>. (on-line Access on 13 Nov. 2018)

- [65] K. Matsuda and H. Fujii, *H $\infty$ -optimized waveabsorbing control: analytical and experimental results*, **Journal of Guidance Control and Dynamics**, 16:6 (1993) pp. 1146–1153.
- [66] Feedback Instruments MATLAB GUIDE for Feedback Control Instrumentation 33-949S (For use with MATLAB R2006bversion 7.3, 2006)
- [67] A. Ates, B.B. Alagöz and C. Yeroğlu, Applicability of homotopy perturbation method for analytical solution of fractional order nonlinear systems, International Conference on Fractional Differentiation and its Applications July 18-20, (2016), Novi Sad, Serbia, pp.841-850.
- [68] W. Dongqing, F. Ding and L. Ximei, *Least squares algorithm for an input nonlinear system with a dynamic subspace state space model*, **Nonlinear Dynamics**, 75:1-2 (2004) pp. 49-61.
- [69] L. Yan, Y.Q. Chen and H.S. Ahn, *Fractional-order iterative learning control for fractional-order linear systems*, **Asian Journal of Control**, 13:1 (2011) pp. 54-63.
- [70] Y.Q. Chen, I. Petras, D. Xue, Fractional order control-a tutorial, American Control Conference, ACC'09, June 10-12, (2009) St. Louis, MO, USA, pp.1397-1411.
- [71] L. Dorčák, J. Valsa, E. Gonzalez, J. Terpák, I. Petráš and L. Pivka, *Analogue Realization of Fractional-Order Dynamical Systems*, **Entropy**, 15:10 (2013) pp. 4199-4214.
- [72] C. Chen and X. Zunxiang, *Robust Stability Analysis of Nonlinear Fractional-Order Time-Variant Systems*, **Mathematical Problems in Engineering**, 2014 (2014) pp. 1-6.
- [73] S.J. Liao, *An approximate solution technique not depending on small parameters: A special example*, **International Journal of Non-Linear Mechanics**. **30:3** (1995) pp. 371-380.
- [74] J.H. He, *Homotopy perturbation technique*, **Comp. Methods Appl. Mech. Eng.**, **178:3-4** (1999) pp. 257–262.
- [75] S.H. Hosseinniaa, A. Ranjbara and S. Momanib, *Using an enhanced homotopy perturbation method in fractional differential equations via deforming the linear part*, **Computers & Mathematics with Applications**, 56:12 (2008) pp. 3138-3149.
- [76] J.H. He, *A coupling method of a homotopy technique and a perturbation technique for non-linear problems*, **International Journal of Non-Linear Mechanics**, **35:1** (2000) pp. 37-43.

- [77] H. Jafari and S. Momani, *Solving fractional diffusion and wave equations by modified homotopy perturbation method*, **Phys. Lett. A**, 370:5-6 (2007) pp. 388-396.
- [78] D. Mehdi, J. Manafian and A. Saadatmandi, *Solving nonlinear fractional partial differential equations using the homotopy analysis method*, **Numerical Methods for Partial Differential Equations**, 26:2 (2010) pp. 448-479.
- [79] G.M. Abd El-Latif, *A homotopy technique and a perturbation technique for non-linear problems*, **Applied Mathematics and Computation**, 169:1 (2005) pp. 576-588.
- [80] D. Xue, C. Zhao and Y.Q. Chen, *A Modified Approximation Method of Fractional Order System*, IEEE International Conference on Mechatronics and Automation, June 25-28, (2006), Luoyang, Henan, China, pp. 1043-1048.
- [81] D. Sierociuk, I. Podlubny and I. Petras, *Experimental evidence of variable-order behavior of ladders and nested ladders*, **Control Systems Technology**, 21:2 (2013) pp. 459-466.
- [82] C. Yeroglu and A. Ates, *A stochastic multi-parameters divergence method for online auto-tuning of fractional order PID controller*, **Journal of the Frankline Institute**, 351:5 (2014) pp. 2411–2429.
- [83] B.B. Alagoz, A. Ates and C. Yeroglu, *Auto-tuning of PID controller according to fractional order reference model approximation for DC rotor control*, **Mechatronics**, 23:7 (2013) pp.789-797.
- [84] R. Caponetto, G. Dongola, L. Fortuna and I. Petras, *Fractional Order Systems Modeling and Control Applications*. **World Scientific Series on Nonlinear Science A series**, 72. (2010) pp. 200.
- [85] A. Ranjbar, S.H. Hosseinnia, H. Soltani and J. Ghasemi, *A Solution of Riccati Nonlinear Differential Equation Using Enhanced Homotopy Perturbation Method*, **Int. J. Eng.**, 21 (2008) 27–38.
- [86] Y. Khan and Q. Wu, *Homotopy perturbation transform method for nonlinear equations using He's polynomials*, **Computers & Mathematics with Applications**, 61:8 (2011) pp. 1963-1967.
- [87] J.H. He, *Some asymptotic method for strongly nonlinear equations*, **Internat. J. Modern Phys. B.**, 20:10 (2006) pp. 1141-1199.
- [88] Z. Odibat and S. Momani, *Modified homotopy perturbation method: application to quadratic Riccati differential equation of fractional order*, **Chaos, Solitons & Fractals**, 36:1, (2008) pp. 167-174.
- [89] Riccati Equations (2018), [www.sosmath.com/diffeq/first/riccati/riccati.html](http://www.sosmath.com/diffeq/first/riccati/riccati.html) (on-line access on: 18.11.2017).

- [90] S. Mukherjee and R. Barnamali, *Solution of Riccati Equation with Variable Co-efficient by Differential Transform Method*, **International Journal of Nonlinear Science**, 14:2, (2012) pp. 251-256.
- [91] R.E. Gutiérrez, J.M. Rosário and J.A.T. Machado, *Fractional Order Calculus: Basic Concepts and Engineering Applications*. **Mathematical Problems in Engineering**, 2010 (2010) pp. 1-19.
- [92] G.W. Leibnitz, *Letter from Hanover, 1695 to GA L'hospital*, *Leibnizen Mathematische Schriften Olms Verlag*, First published in 1849, Germany: Hildesheim, 1962.
- [93] Y. Luo, Y.Q. Chen, H.S. Ahn and Y.G. Pi, *Fractional Order Periodic Adaptive Learning Compensation for State-Dependent Periodic Disturbance*, **IEEE Transactions on Control System Technology**, 20:2 (2012) pp. 465-472.
- [94] H.S. Li, Y. Luo and Y.Q. Chen. *A Fractional Order Proportional and Derivative (FOPD) Motion Controller: Tuning Rule and Experiments*, **IEEE Transactions on Control System Technology**, 18:2 (2010) pp. 516-520.
- [95] L. Liu, S. Zhang, Y. Chen and D. Xue, *General Robustness Analysis and Robust Fractional-Order PD Controller Design for Fractional-Order Plants*. **IET Control Theory & Applications**, 12:2 (2018) pp. 1730-1736.
- [96] X. Li, Y. Wang, N. Li, M. Han, Y. Tang and F. Liu, *Optimal fractional order PID controller design for automatic voltage regulator system based on reference model using particle swarm optimization*, **International Journal of Machine Learning and Cybernetics**, 8:5 (2017) pp. 1595-1605.
- [97] B. Xu, R.S. Pandian, N. Sakagami and F. Petry, *Neuro-fuzzy control of underwater vehicle-manipulator systems*. **Journal of the Franklin Institute**, 349:3, (2012) pp.1125-1138.
- [98] M. Radek, R. Prokop, *Single-parameter tuning of PI controllers: Theory and application*. **Journal of the Franklin Institute**, 348:8 (2011) pp. 2059-2071.
- [99] Ö. Aydoğdu, R. Akkaya, *An effective real coded GA based fuzzy controller for speed control of a BLDC motor without speed sensor*, **Turkish Journal of Electrical Engineering & Computer Sciences**, 19:3 (2011) pp. 413-430.
- [100] A. Bagis, *Tabu search algorithm based PID controller tuning for desired system specifications*, **Journal of the Franklin Institute**, 348:10, (2011), pp. 2795-2812.
- [101] C.A. Monje, B.M. Vinagre, Y.Q. Chen, V. Feliu, P. Lanusse and J. Sabatier, *Proposals for fractional tuning*, **1st IFAC Workshop on Fractional Differentiation and its Applications FDA'04**; 19-21 July, (2004), Bordeaux, France.

- [102] D.Song, J. Han and G. Liu, *Active Model-Based Predictive Control and Experimental Investigation on Unmanned Helicopters in Full Flight Envelope*, **IEEE Transactions on Control System Technology**, 21:4 (2013) pp. 1502-1509.
- [103] I. Pan, S. Das and A Gupta, *Handling packet dropouts and random delays for unstable delayed processes in NCS by optimal tuning of controllers with evolutionary algorithms*, **ISA transactions**, 50:4 (2011) pp. 557-572.
- [104] B.T. Krishna and K.V. Reddy, *Active and passive realization of fractance device of order  $\frac{1}{2}$* , **Active and Passive Electronic Components**, 2008 (2008) pp.1- 5.
- [105] M.M. Özyetkin, C. Yeroglu, N. Tan and M. E. Tagluk, *Design of PI and PID controllers for fractional order time delay systems*, **IFAC Proceedings Volumes**, **43** (2010) pp. 355-360.
- [106] C. Yeroglu and N. Tan, *Note on fractional-order proportional-integral-differential controller design*, **IET Control Theory Appl.**, 5:17 (2011) pp. 1978-1989.
- [107] B. B. Alagoz, A. Ates, C. Yeroglu, *Auto-tuning of PID controller according to fractional-order reference model approximation for DC rotor control*, **Mechatronics**, **23:7** (2013) pp. 789-797.
- [108] T.K. Liu and J.G. Juang, *A Single Neuron PID Control for Twin Rotor MIMO System*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, July, 14-17 July, (2009), Suntec Convention and Exhibition Center, Singapore, pp.186-191.
- [109] P. Wen and T.W. Lu, *Decoupling control of a twin rotor MIMO system using robust deadbeat control technique*, **IET Control Theory Applications**, 2:11 (2008) pp.999-1007.
- [110] J.G. Juang, M.T. Huang and W.K. Liu, *PID Control Using Pre-searched Genetic Algorithms for a MIMO System*, **IEEE Transactions on Man and Cybernetics- Part C: Applications and Reviews**, 38:5 (2008) pp.716-727.
- [111] H. Chao, Y. Luo, L. Di and Y.Q. Chen, *Roll-channel fractional order controller design for a small fixed-wing unmanned aerial vehicle*, **Control Engineering Practice**, 18:7 (2010) pp. 761-772.
- [112] A. Ateş and C. Yeroğlu, *Online Tuning of Two Degrees of Freedom Fractional Order Control Loops*, **Balkan Journal Of Electrical & Computer Engineering**, 1:4 (2016) pp. 5-11.
- [113] AAteş and C. Yeroğlu, *SMDO algoritması ile iki serbestlik dereceli FOPID kontrol çevrimi tasarımı*, **Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi**, 22:8 (2016) pp. 671-676.

- [114] I.M. Horowitz, *Synthesis of Feedback Systems*. Academic Press, New York, The University of Michigan, (1963), p. 1-726.
- [115] M.Araki, *PID control system with reference feedforward (PID-FF control systems)*, Proc 23rd SICE (Society of Instrument and Control Engineers) Annual Conference, (1984), pp.31-32.
- [116] M. Araki and H. Taguchi, *Two-degree-of-freedom PID controllers*, **International Journal of Control Automation and Systems**, 1:4 (2003) pp. 401-411.
- [117] T. Nagashio, T. Kida, Y. Hamada and T. Ohtani, *Robust Two-Degrees-of-Freedom Attitude Controller Design and Flight Test Result for Engineering Test Satellite-VIII Spacecraft*, **Control Systems Technology**, **22:1** (2014) pp.157-168.
- [118] M. Ajmeri and A. Ali, *Two degree of freedom control scheme for unstable processes with small time delay*, **ISA Transactions**, 56 (2015) pp. 308-326.
- [119] B.B. Ghosh, B.K. Sarkar and R. Saha, *Realtime performance analysis of different combinations of fuzzy-PID and bias controllers for a two degree of freedom electro hydraulic parallel manipulator*, **Robotics and Computer-Integrated Manufacturing**, 34 (2014) pp.62-69.
- [120] G.H. Choi, K. Park and J.H. Jung, *An optimal H<sub>2</sub> decoupling design for non-square plant systems based on the two-degree-of-freedom standard model*. **International Journal of Control, Automation and Systems**, 7:2 (2009) pp. 193-202.
- [121] F. B. Merrikh, N. Mirebrahimi and M.R. Khalili, *Discrete-time fractional-order PID controller: Definition, tuning, digital realization and some applications*, **International Journal of Control, Automation and Systems**, 13:1 (2015) pp. 81-90.
- [122] G. Feng and Z. Xiao-ping, *Research on fractional order two-degrees-of-freedom flight control technology of unmanned air vehicle*, **Computer Science and Information Processing**, Aug.24-26, (2012), Xi'an, Shaanxi, China, pp. 807-812.
- [123] S. Debbarma, L. C. Saikia and N. Sinha, *Automatic generation control using two degree of freedom fractional order PID controller*, **International Journal of Electrical Power & Energy Systems**, **58** (2014) pp. 120-129.
- [124] G. Saravanakumar, A. Babu, R. John, M. Vishnuprasad, S. Sivaguru, *Design of Fractional Order PID Controller for Integrating Time Delay Systems*, **Digital Signal Processing**, **10:3** (2018) pp. 40-44.
- [125] R. El-Khazali, *Fractional-order controller design*, **Computers & Mathematics with Applications**, **66:5** (2013) pp. 639-646.

- [126] A. Ates and C. Yeroglu, Tabu Search Algorithm for Fractional Order PID via Non-linear Multi Objective Function, International Conference on Fractional Differentiation and Its Applications, 23-25 June, (2014), Catania, Italy, pp.1-6.
- [127] A. Ates, C. Yeroglu, B. B. Alagoz and B. Senol, Tuning of Fractional Order PID with Master Slave Stochastic Multi-Parameter Divergence Optimization Method, International Conference on Fractional Differentiation and Its Applications, 23-25 June, (2014), Catania, Italy, pp.1-6.
- [128] A. Ates and C., Yeroğlu, *Optimal fractional order PID design via Tabu Search based algorithm*, **ISA Transactions**, **60** (2016) pp. 109-118.
- [129] A. Hajiloo, N. Nariman-zadeh and A. Moeini, *Pareto optimal robust design of fractional-order PID controllers for systems with probabilistic uncertainties*, **Mechatronics**, **22:6** (2012) pp. 788-801.
- [130] M.T. Lin, C.L. Yen, M.S. Tsai and H.T. Yau, *Application of robust iterative learning algorithm in motion control system*, **Mechatronics**, **23:5** (2013) pp. 530-540.
- [131] H. Malek, Y. Luo and Y. Q. Chen, *Identification and tuning fractional order proportional integral controllers for time delayed systems with a fractional pole*, **Mechatronics**, **23:7** (2013) pp. 746-754.
- [132] C. Yeroglu and G. Kavuran, *Sliding Mode Controller Design with Fractional Order Differentiation: Applications on Unstable Time Delay Systems*, **Turkish Journal of Electrical Engineering & Computer Sciences**, **22:5** (2014) pp.1270-1286.
- [133] F. Glover, *Tabu search-part I*, **ORSA Journal on computing**, **1:3** (1989) pp. 190-206.
- [134] F. Glover, *Tabu search-part II*, **ORSA Journal on computing**, **2:1** (1990). pp. 4-32.
- [135] R.L. Haupt and S.E. Haupt, *Practical Genetic Algorithms*. A John Wiley & Sons, INC. Publication, Canada, 2004, p. 251.
- [136] R. Martí, M. Laguna and F. Glover, *Principles of scatter search*, **European Journal of Operational Research**, **169:2** (2006) pp. 359-372.
- [137] D. Henderson, S. H. Jacobson and A. W. Johnson, *The theory and practice of simulated annealing*. In *Handbook of metaheuristics*, Springer, US, 2003, p. 287-319.
- [138] H. Katagiri, T. Hayashida, I. Nishizaki and Q. Guo, A hybrid algorithm based on tabu search and ant colony optimization for k-minimum spanning tree problems, **Expert Systems with Applications**, **39:5** (2012) pp. 5681-5686.

- [139] R. Ramezani, D. Rahmani and F. Barzinpour, *An aggregate production planning model for two phase production systems: Solving with genetic algorithm and tabu search*, **Expert Systems with Applications**, **39:1** (2012) pp. 1256-1263.
- [140] B. Ramkumar, M. P. Schoen and F. Lin, *Hybrid enhanced continuous tabu search and genetic algorithm for parameter estimation in colored noise environments*, **Expert Systems with Applications**, **38:4** (2011) pp. 3909-3917.
- [141] D. Karaboga and A. Kalinli, Tuning PID controller parameters using tabu search algorithm Proceedings of the IEEE International Conference on Systems Man and Cybernetics, Oct. 14-17, (1996), Beijing, China, pp. 134-136.
- [142] N. Talbi and K. Belarbi, Fuzzy rule base optimization of fuzzy controller using hybrid tabu search and particle swarm optimization learning algorithm World Congress on Information and Communication Technologies, Dec. 11-14 2011, Mumbai, India, pp. 1139-1143.
- [143] M. H. Mashinchi, M. A. Orgun and W. Pedrycz, *Hybrid optimization with improved tabu search*, **Applied Soft Computing**, **11:2** (2011) pp. 1993-2006.
- [144] J. Brandão, *A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem*, **Computers & Operations Research**, **38:1** (2011) pp. 140-151.
- [145] Y. Xia and Z. Fu, *A tabu search algorithm for distribution network optimization with discrete split deliveries and soft time Windows*, *Cluster Computing*, (2018), pp.1-11.
- [146] A. Ates and C. Yeroglu, *Modified Artificial Physics Optimization for Multi-parameters Functions*, **Transaction on Iranian journal electrical and electronic engineering**, **42.4** (2018) pp. 465-478.
- [147] L. Xie, J. Zeng and X. Cai, A hybrid vector artificial physics optimization with multi-dimensional search method Second International Conference on Innovations in Bio-inspired Computing and Applications (IBICA), Dec. 16-18, (2011) Shenzhen, China, pp. 116-119.
- [148] W. M. Spears, D.F. Spears, R. Heil, W. Kerr and S. Hettiarachchi, *An Overview of Physicomimetics*, **Lecture Notes in Computer Science-State of the Art Series**, (2005) pp. 84-97.
- [149] W. M. Spears, D.F. Spears, R. Heil, W. Kerr and S. Hettiarachchi, *An overview of physicomimetics*. Springer, Berlin, (2004), p. 84-97.
- [150] W. M. Spears and D. F. Spears, *Physicomimetics: Physics-based swarm intelligence*. Springer Science & Business Media, Springer, Berlin, (2012) p. 629.
- [151] G. Beni and S. Hackwood, Stationary waves in cyclic swarms, *IEEE International Symposium on Intelligent Control*, Aug.11-13, (1992), Glasgow, UK, pp.234-24.

- [152] G. Beni and J. Wang, Swarm intelligence, Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan, Tokyo, Japan, (1989), pp. 425-428.
- [153] D. Marco and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Intelligence*, Oxford University Press, New York, USA, 1999, p. 1-309.
- [154] A.T. Hayes, A. Martinoli and R.M. Goodman, Swarm robotic odor localization, In Intelligent Robots and Systems IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct.29-Nov.3, (2001), Maui, HI, USA, pp. 1073-1078.
- [155] L. Xie, J. Zeng and R. A. Formato, *Convergence analysis and performance of the extended artificial physics optimization algorithm*, **Applied Mathematics and Computation**, **218:8** (2011) pp. 4000-4011.
- [156] L.Xie, J. Zeng and Z. Cui, General framework of artificial physics optimization algorithm, World Congress on Nature & Biologically Inspired Computing, Dec.9-11, (2009), Coimbatore, India, pp.1321-1326.
- [157] L.Xie and J. Zeng, An extended artificial physics optimization algorithm for global optimization problems, Fourth International Conference on Innovative Computing, Information and Control, Dec.7-9, (2009), Kaohsiung, Taiwan, pp.881-884.
- [158] L. Xie, Y. Tan and J. Zeng, The convergence analysis and parameter selection of artificial physics optimization algorithm, The International Conference on Modelling, Identification and Control, July 17-19, (2010), Okayama, Japan, pp. 562-567.
- [159] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*, International Society of Automation. (1995).
- [160] D.Xue, Y. Q. Chen, D. P. Atherton, *Linear feedback control: analysis and design with MATLAB*. Society for Industrial and Applied Mathematics, (2007), p.347.
- [161] I. Zelinka, S. Celikovský, H. Richter and G. Chen, *Evolutionary Algorithms and Chaotic Systems*, *Studies Computational Intelligence 267*, Springer, Verlag Berlin Heidelberg, (2010), p. 1-413.
- [162] Y. Q. Chen, B. M. Vinagre and I. Podlubny, *Continued fraction expansion approaches to discretizing fractional order derivatives-an expository review*. **Nonlinear Dynamics**, 38:1-4 (2004) pp.155-170.
- [163] I. Podlubny, I. Petraš, B. M. Vinagre, P. O'leary and L. Dorčák, *Analogue realizations of fractional-order controllers*, **Nonlinear dynamics**, **29:1-4** (2002) pp.281-296.

- [164] D. Davendra, I. Zelinka, R. Senkerik, *Chaos driven evolutionary algorithms for the task of PID control*, **Computers & Mathematics with Applications**, 60:4 (2010) pp.1088-1104.
- [165] A. Tepljakov, E. Petlenkov, J. Belikov and M. Halas, Design and implementation of fractional-order PID controllers for a fluid tank system American Control Conference (ACC), Jun. 17-19, (2013), Washington, USA.
- [166] 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**, 16:7 (2008) pp.798-812.
- [167] 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) pp.25-39.
- [168] B. M. Vinagre, C. A. Monje, A. J. Calderón and J. I. Suárez, *Fractional PID controllers for industry application. A brief introduction*, **Journal of Vibration and Control**, 13:9-10 (2007) pp. 1419-1429.
- [169] A. Ateş, B. B. Alagöz, C. Yeroğlu, J. Yuan, Y.Q. Chen, Disturbance Rejection FOPID Control of Rotor by Multi-Objective Bb-Bc Optimization Algorithm Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE, Aug. 6-9, (2017), Cleveland, USA, pp. 1-6.
- [170] 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, July 29- Aug. 1, (2005), Niagara Falls, Ont., Canada, pp. 216-221).
- [171] A. Zamani, S. M. Barakati and S. Yousofi-Darmian, *Design of a fractional order PID controller using GBMO algorithm for load-frequency control with governor saturation consideration*, **ISA transactions**, 64 (2016) pp.56-66.
- [172] Z. Wu, D. Li and L. Wang, Control of the superheated steam temperature: A comparison study between PID and fractional order PID controller, 35th Chinese Control Conference, July 27-29, (2016), Chengdu, China, pp.10521-10526.
- [173] Z. Chen, X. Yuan, B. Ji, P. Wang and T. Tian, *Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II*, **Energy Conversion and Management**, 84 (2014) pp. 390-404.
- [174] D. Valério and J. S. da Costa, *Tuning of fractional PID controllers with Ziegler–Nichols-type rules*, **Signal Processing**, 86:10 (2006) pp.2771-2784.

- [175] A.A. Zamani, S. Tavakoli and S. Etedali, *Fractional order PID control design for semi-active control of smart base-isolated structures: A multi-objective cuckoo search approach*, **ISA Transactions**, 67 (2017) pp. 222-232.
- [176] X. Liu, *Optimization design on fractional order PID controller based on adaptive particle swarm optimization algorithm*, **Nonlinear Dynamics**, 84:1 (2016), 379-386.
- [177] G. Q. Zeng, J. Chen, Y. X. Dai, L. M. Li, C. W. Zheng and M. R. Chen, *Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization*. **Neurocomputing**, 160 (2015) pp. 173-184.
- [178] G. Q. Zeng, J. Chen, M. R. Chen, Y. X. Dai, L. M. Li, K. D. Lu and C. W. Zheng, *Design of multivariable PID controllers using real-coded population-based extremal optimization*, **Neurocomputing**, 151 (2015) pp. 1343-1353.
- [179] Y. Naranjani, Y. Sardahi, Y. Chen and J. Q. Sun, *Multi-objective optimization of distributed-order fractional damping*, **Communications in Nonlinear Science and Numerical Simulation**, 24:1-3 (2015) pp.159-168.
- [180] B. B. Alagoz, F. N. Deniz, C. Keleş and N. Tan, *Disturbance rejection performance analyses of closed loop control systems by reference to disturbance ratio*. **ISA Transactions**, 55 (2015) pp. 63-71.
- [181] B. B. Alagoz, N. Tan, F. N. Deniz and C. Keles, *Implicit disturbance rejection performance analysis of closed loop control systems according to communication channel limitations*, **IET Control Theory & Applications**, 9:17 (2015) pp. 2522-2531.
- [182] D. Xue and Y. Q. Chen, *Modeling, Analysis and Design of Control Systems in MATLAB and Simulink*. World Scientific Publishing Company, 2014, p.580.
- [183] R. Sharma, P. Gaur and A. P. Mittal, *Performance analysis of two-degree of freedom fractional order PID controllers for robotic manipulator with payload*. **ISA Transactions**, 58 (2015) pp. 279-291.
- [184] H. M. Genç, I. Eksin and O. K. Erol, *Big Bang-Big Crunch optimization algorithm hybridized with local directional moves and application to target motion analysis problem*, In Proceedings of the International Conference on Systems Man and Cybernetics, Oct. 10-13, (2010) Istanbul, Turkey, pp:881-887.

## ÖZGEÇMİŞ

**Ad- Soyadı:** Abdullah ATEŞ

**Doğum Yeri- Tarihi:** Elâzığ- 01.01.1988

**E-posta:** abduallah.ates@inonu.edu.tr

**Lisans:** Erciyes Üniversitesi Elektrik Elektronik Mühendisliği Bölümü (2011)

**Yüksek Lisans:** İnönü Üniversitesi Bilgisayar Mühendisliği Bölümü (2014)

**Doktora:** İnönü Üniversitesi, Bilgisayar Mühendisliği Bölümü (2018)

**Mesleki Deneyim ve Ödüller:** Araştırma görevlisi-İnönü Üniversitesi Bilgisayar Mühendisliği Bölümü. (2011- )

Araştırmacı- University of California Merced- MESA Lab. (2017 Mayıs-2018 Mayıs)

## Yayın Listesi:

### 1. Uluslararası hakemli dergilerde yayımlanan makaleler (SCI & SSCI)

**A1. A.Ates, C. Yeroglu, Modified Artificial Physics Optimization for Multi Parameter Functions, Iranian Journal of Science and Technology, Transactions of Electrical Engineering, pp 1-14, 2018.**

**A2. B. Senol, A. Ates, B. B. Alagoz, C. Yeroglu, Stability Analysis of Fractional Order Multiple Linear Uncertain Systems: Exposed Edge Sampling Approach, Journal of Control Engineering and Applied Informatics, 20 (2), pp. 14-21, 2018.**

**A3. Kavuran, G., Ates, A., Alagoz, B. B., Yeroglu, C. "An Experimental Study on Model Reference Adaptive Control with Fractional Order Adjustment Rules for DC Rotor Control." Journal of Control Engineering and Applied Informatics, 19(4), 101-111, 2017.**

**A4. Alagoz, B. B., Kavuran, G., Ates, A., Yeroglu, C., "Reference-shaping adaptive control by using gradient descent optimizers", PloS one, 12(11), 2017.**

**A5. Ateş A., Alagöz B.B., Kavuran G., Yeroğlu C., "Implementation of fractional order filters discretized by modified Fractional Order Darwinian Particle Swarm Optimization", Measurement, vol.107, pp.153-164, 2017**

**A6. Ateş A., Alagöz B.B., Yeroğlu C., "Master-Slave Stochastic Optimization for Model-Free Controller Tuning", Iranian Journal of science and technology- transactions of electrical engineering, vol.1, pp.1-11, 2017.**

**A7. A. Ates and C. Yeroglu, Optimal Fractional Order PID Design via Tabu Search Based Algorithm, Isa Transactions, vol. 60, pp. 109–118, 2016.**

**A8. B. B. Alagoz, C. Yeroglu, B. Senol, A. Ates, Probabilistic robust stabilization of fractional order systems with interval uncertainty, Isa Transactions, vol. 57, pp. 101-110, 2015.**

**A9.** B. B. Alagoz, **A. Ates**, C. Yeroglu, B. Senol, An experimental investigation for error-cube PID control, Transactions Of The Institute Of Measurement And Control, vol. 37, no. 5, pp. 657-660, 2015

**A10.** C. Yeroglu, **A. Ates**, A stochastic multi-parameters divergence method for online auto-tuning order PID controllers of fractional, Journal of the Franklin Institute-Engineering and Applied Mathematics, vol. 351, no. 5, pp.2411-2429, 2014.

**A11.** B. Senol, **A. Ates**, B. B. Alagoz, C. Yeroglu, A numerical investigation for robust stability of fractional-order uncertain systems, Isa Transactions, vol. 53, no.2, pp. 189-198, 2014.

**A12.** B. B. Alagoz, **A. Ates**, C. Yeroglu, Auto-tuning of PID controller according to fractional-order reference model approximation for DC rotor control, Mechatronics, vol. 62, no. 7, pp. 845-855, 2013

## **2. Uluslararası diğer hakemli dergilerde yayımlanan makaleler (Other Index)**

**B1.** **A. Ates**, C. Yeroglu, Online Tuning of Two Degrees of Freedom Fractional Order Control Loops, Balkan Journal of Electrical & Computer Engineering, vol. 4, no. 1, pp. 5-11, 2016.

**B2.** G. Kavuran, C. Yeroglu, **A. Ates**, B. B. Alagoz, Effects of fractional order integration on ASDM signals, International Journal of Dynamics and Control, pp.1-8, 2016.

**B3.** G. Kavuran, B. B. Alagoz, **A. Ates**, C. Yeroglu, Implementation of Model Reference Adaptive Controller with Fractional Order Adjustment Rules for Coaxial Rotor Control Test System, Balkan Journal of Electrical & Computer Engineering, vol. 4, no. 2, pp. 84-88, 2016.

**B4.** **A. Ates**, B. B. Alagoz, G. T. Alisoy, C. Yeroglu, H. Z. Alisoy, Fuzzy Velocity and Fuzzy Acceleration in Fractional Order Motion, Balkan Journal of Electrical & Computer Engineering, vol. 3, no. 2, pp. 98-102, 2015.

## **3. Ulusal ulakbimde ve dergi parkta taranan dergilerde yapılan çalışmalar**

**C.1** **A. Ateş**, C. Yeroğlu, SMDO Algoritması ile İki Serbestlik Dereceli FOPID Kontrol Çevrimi Tasarımı, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, Pamukkale Univ Muh Bilim Derg, 22(8), 671-676, 2016.

**C.2** G. Kavuran, **A. Ates**, B. B. Alagoz, C. Yeroglu, Kesir dereceli model referans denetleyici ile görüntü işleme destekli nesne takip uygulaması, Pamukkale Univ Muh Bilim Derg, 22(8), 659-665, 2016.

## **4. Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler**

**D1.** **A. Ates**, J. Yuan, S. Deghan, Y. Zhao, C. Yeroglu, PID2018 Benchmark Challenge: Multi Objective Stochastic Optimization Algorithm, PID 2018 Conference, Ghent, Belgium.

**D2.** **A. Ates**, C. Yeroglu, J. Yuan, Y. Q. Chen, S.E. Hamamci, Optimization of the FO[PI] Controller for MTDS using MAPO with Multi Objective Function, ,

International Conference on Fractional Differentiation and its Applications, Amman, July 16-18, 2018.

**D3.** J. Yuan, **A. Ates**, S. Deghan, Y. Zhao, S. Fei, Y. Q. Chen, PID2018 Benchmark Challenge: Model-based Feedforward Compensator with A conditional integrator, PID 2018 Conference, Ghent, Belgium.

**D4.** S. Deghan, T. Zhao, Y. Zhao, J. Yuan, **A. Ates**, Y.Q. Chen, PID2018 Benchmark Challenge: Model Predictive Control with conditional integral control using a general purpose Optimal control problem solver- RIOTS, PID 2018 Conference, Ghent, Belgium.

**D5.** Y. Zhao, S. Deghan, **A. Ates**, J. Yuan, F. zhaou, Y. Li, Y. Q. Chen, PID2018 Benchmark Challenge: learning feedforward control, PID 2018 Conference, Ghent, Belgium.

**D6.** **Ateş A.**, Alagöz B.B., Yeroğlu C., Yuan J., Chen Y., "Disturbance Rejection FOPID Control of Rotor by Multi-Objective Bb-Bc Optimization Algorithm", Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2017, Cleveland, A.B.D., 6-9 Ağustos 2017, pp.1-6

**D7.** Kavuran G., Alagöz B.B., **Ateş A.**, Yeroğlu C., "A Model Reference Control Scheme by MIT Rule with Fractional Order Sliding Surface", International Conference of Mathematical Modelling and Engineering (MME), Ankara, Türkiye, 27-29 Nisan 2017, pp.1-1

**D8.** Alisoy H., Kavuran G., **Ateş A.**, Alagöz B.B., Yeroğlu C., "Numerical Modelling of Inaccessible Subsystem Dynamics: An Application for Control Engineering Practices", International Conference of Mathematical Modelling and Engineering (MME), Ankara, Türkiye, 27-29 Nisan 2017, pp.1-1

**D9.** B. B. Alagoz, C. Yeroglu, G. Kavuran, **A. Ates**, Detection of RR Interval Alterations in ECG Signals by Using First Order Fractional Filter, 39<sup>th</sup> International Conference on Telecommunications and Signals Processing, Vienna. Austria, 2016.

**D10.** **A. Ates**, G. Kavuran, B. B. Alagoz, C. Yeroglu, Improvement of IIR Filter Discretization for Fractional Order Filter by Discrete Stochastic Optimization, 39<sup>th</sup> International Conference on Telecommunications and Signals Processing, Vienna. Austria, 2016.

**D11.** **A. Ates**, B. B. Alagoz, C. Yeroglu, Applicability of homotopy perturbation method for analytical solution of fractional order nonlinear systems, International Conference on Fractional Differentiation and its Applications, Novi Sad Serbia, July 18-20, 2016.

**D12.** G. Kavuran, B. B. Alagoz, **A. Ates**, C. Yeroglu, Implementation of fractional order filters by embedded programming techniques, International Conference on Fractional Differentiation and its Applications, Novi Sad Serbia, July 18-20, 2016.

**D13.** C. Yeroglu, **A. Ates**, Optimized fractional order controller for multiple time delay systems, International Conference on Fractional Differentiation and its Applications, Novi Sad Serbia, July 18-20, 2016.

**D14.** **A. Ates**, B. B. Alagoz, C. Yeroglu, H. Alisoy, Sigmoid Based PID Controller Implementation for Rotor Control, European Control Conference (ECC'15), Linz, Austria. 2015.

**D15.** G. Kavuran, C. Yeroglu, **A. Ates**, B. B. Alagoz, A Note on Effects of Fractional Order Integration on ASDM Signals, The International Symposium on Fractional Signals and Systems (FSS2015), Cluj-Napoca, Romania, 2015

**D16. A. Ates**, C. Yeroglu, Tabu Search Algorithm for Fractional Order PID via Non-linear Multi Objective Function, International Conference on Fractional Differentiation and Its Applications (ICFDA14), Catania, Italy, 2014.

**D17. A. Ates**, C. Yeroglu, B. B. Alagoz, B. Senol, Tuning of Fractional Order PID with Master Slave Stochastic Multi-Parameter Divergence Optimization Method, International Conference on Fractional Differentiation and Its Applications (ICFDA14), Catania, Italy, 2014.

## **5. Ulusal arası Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler**

**E1. A. Ates** and C. Yeroglu, SMDO Algoritması ile İki Serbestlik Dereceli FOPID Kontrol Çevrimi Tasarımı, Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, Denizli, Turkey, 2015.

**E2. G. Kavuran**, **A. Ates**, B. B. Alagoz, C. Yeroglu, Kesir Dereceli Model Referans Denetleyici ile Görüntü İşleme Destekli Kapalı Çevrim Nesne Takip Uygulaması, Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, Denizli, Turkey, 2015.

**E3. B. B. Alagoz**, **A. Ates**, G. Kavuran, C. Yeroglu, Darbe Genişliği Modülasyonlu PID Kontrolör İle Rotor Kontrolü, Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, Denizli, Turkey, 2015.

**E4. A. Ates**, B. B. Alagoz, C. Yeroglu, B. Şenol, Kesir Dereceli PID Kontrolörler İçin Referans Model Tabanlı Optimizasyon Yöntemi ,Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, Malatya, Turkey, 2013.

**E5. B. B. Alagoz**, **A. Ates**, C. Yeroglu, Hata-Küpü Kontrol Yapısının Teorik İncelenmesi, Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, Malatya, Turkey, 2013.

**E6. A. Ates** and C. Yeroglu, TRMS İçin Referans Modele Dayalı Optimal Kesir Dereceli PID Tasarımı, Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, Niğde, Turkey, 2012.

**E7. A. Ates** and C. Yeroglu, Gerçek Zamanlı TRMS için Geliştirilen YSA Algoritması, Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı, Niğde, Turkey, 2012.