

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRONİK ve HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI**

DOKTORA TEZİ

**SERİ ve PARALEL ROBOTLARDA PARÇACIK SÜRÜ
OPTİMİZASYONU İLE YÖRÜNGE KONTROLÖRÜ TASARIMI**

OĞUZHAN KARAHAN

KOCAELİ 2012

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK ve HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

SERİ ve PARALEL ROBOTLARDA PARÇACIK SÜRÜ
OPTİMİZASYONU İLE YÖRÜNGE KONTROLÖRÜ TASARIMI

Oğuzhan KARAHAN

Prof.Dr. Zafer BİNGÜL
Danışman, Kocaeli Üniv.

Doç.Dr. Serdar KÜÇÜK
Jüri Üyesi, Kocaeli Üniv.

Doç.Dr. Oğuzhan URHAN
Jüri Üyesi, Kocaeli Üniv.

Doç.Dr. Erkan ZERGEROĞLU
Jüri Üyesi, Gebze Yük. Tek. Ens.

Yrd.Doç.Dr. Metin AYDIN
Jüri Üyesi, Kocaeli Üniv.



Tezin Savunulduğu Tarih: 28.06.2012

ÖNSÖZ ve TEŞEKKÜR

Son yıllarda endüstriyel, savunma sanayi, tıbbi ve yüksek teknoloji gibi birçok uygulamalarda paralel manipülatörler hassas konumlama, yüksek hızda çalışma, iyi bir dinamik performans ve ağır yük taşıma kapasiteleri gibi avantajlarından dolayı seri manipülatörlerin yerine geçmektedir. En yaygın paralel manipülatörlerden birisi de günümüzde kullanılan Stewart Platform manipülatörüdür. Bu paralel manipülatör, boyları prizmatik eklemler vasıtasıyla değişebilen 6 bacak ile birbirine bağlı altıgen biçimli, biri sabit ve diğeri hareketli, iki platformdan oluşmaktadır.

Stewart Platform gibi paralel manipülatörlerin seri manipülatörlere göre daha üstün yeteneklere sahip olmaları, endüstrinin birçok alanında geniş çalışma alanı bulmuştur. Dolayısıyla, lazerle hassas kesme, helikopter inme pisti, ameliyatlarda kullanılan hassas cerrahi operasyonlar, hava taşıtı simülatörleri, mikro konumlandırma, hafif metal işleme, kesme, şekil verme ve montaj gibi işlerde kullanılması, bu tip mekanizmaların üretimdeki özel önemini göstermektedir.

Bu tez kapsamında, seri ve paralel robotlar için yörünge kontrolü sağlayan parçacık sürü optimizasyon (PSO) tabanlı farklı kontrolörlerin tasarımı gerçekleştirilmiştir. Bu kontrolörleri test etmek için iki serbestlik dereceli düzlemsel robot ve 6 DOF Stewart Platform'u, seri ve paralel manipülatörler olarak seçilmiştir. Bu manipülatörlerin kinematik ve dinamik analizleri, simülasyon modeli ve yörünge kontrolü ayrıntılı bir şekilde incelenmiştir.

Bu tezi bitirme sürecinde başından sonuna yardımlarını esirgemeyen ve yol gösteren sayın hocam Prof. Dr. Zafer BİNGÜL'e en içten teşekkürlerimi sunarım. Ayrıca, tez çalışması süresince ve tüm hayatımın her anında sürekli bana destek olan eşim Nurdan KARAHAN'a sonsuz teşekkür ederim.

Haziran - 2012

Oğuzhan KARAHAN

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ	v
TABLolar DİZİNİ	ix
SİMGELER DİZİNİ ve KISALTMALAR	xi
ÖZET	xiv
ABSTRACT	xv
GİRİŞ	1
1. GENEL BİLGİLER	4
1.1. Giriş	4
1.2. Robotların Sınıflandırılması	5
1.3. Paralel Robotlar	6
1.4. Paralel ve Seri Robotların Karşılaştırılması	10
1.5. Paralel Manipülatörlerin Analizi	11
1.5.1. Kinematik analiz	11
1.5.2. Dinamik analiz	12
1.6. Kontrol	12
2. ROBOT KONTROL SİSTEMLERİ	14
2.1. Giriş	14
2.2. PID Kontrolör	16
2.3. Kesir Dereceli PID Kontrolör	18
2.3.1. Kesirli matematiğin geliřimi	21
2.3.2. Kesirli matematiğin temel kavramları	21
2.3.3. Kesirli matematiğin PID kontrol sistemine uygulanması	24
2.4. Bulanık Sistemler	25
2.4.1. Bulanık kümeler	26
2.4.2. Bulanık küme işlemleri	28
2.4.3. Kavramsal deęişkenler ve ifadeler	30
2.5. Bulanık Kontrol Sistemi	30
2.6. Bulanık Kontrol Çeřitleri	33
2.6.1. Bulanık PD kontrolör	33
2.6.2. Bulanık PI kontrolör	34
2.6.3. Bulanık PID kontrolör	35
2.6.4. Bulanık kazanç ayarlamalı PID kontrolör	37
2.6.5. Karma tip Bulanık PID kontrolör	38
2.7. Bulanık Mantık Kontrolörünün Tasarım Ařamaları	39
2.8. Sonuç	39
3. PARÇACIK SÜRÜ OPTİMİZASYON ALGORİTMASI	41
3.1. Giriş	41
3.2. Algoritmanın İçerięi	42
3.3. Pseudo Kodu ve Notasyon	43
3.4. Sonuç	45
4. SERİ ROBOTLAR	46

4.1. Giriş	46
4.2. İki Serbestlik Dereceli Robotun Bulanık Mantık Yörünge Kontrolü.....	60
4.2.1. Dinamik modeli	61
4.2.2. Bulanık mantık kontrolörü	62
4.2.3. PSO ile bulanık mantık kontrolörün ayarlanması.....	64
4.2.4. PSO ile PID kontrolörün ayarlanması.....	71
4.2.5. Bulanık ve PID kontrol için deneysel sonuçlar	72
4.3. İki Serbestlik Dereceli Robotun Kesir Dereceli PID Yörünge Kontrolü	83
4.3.1. Kesir dereceli PID kontrolör	83
4.3.2. PSO ile kesir dereceli PID kontrolörün ayarlanması.....	84
4.3.3. GA ile kesir dereceli PID kontrolörün ayarlanması.....	86
4.3.4. GA ve PSO tekniklerin karşılaştırılması.....	88
4.4. Sonuç	94
5. STEWART PLATFORM'UN DİNAMİK ANALİZİ	96
5.1. Giriş	96
5.2. Ters Kinematik Analizi.....	104
5.3. Jacobian Matrisinin Elde Edilmesi.....	109
5.3.1. Yöntem A	109
5.3.2. Yöntem B	113
5.4. Dinamik Denklemlerin Çıkarımı.....	115
5.4.1. Üst platformun kinetik enerjisi.....	116
5.4.2. Üst platformun potansiyel enerjisi.....	118
5.4.3. Bacakların kinetik enerjisi.....	119
5.4.4. Bacakların potansiyel enerjisi.....	123
5.4.5. Dinamik denklemlerin yazılımı	124
5.4.6. Eyleyiciler dinamiği.....	128
5.4.7. Eyleyici dinamiği eklenmiş sistem dinamiği.....	132
5.5. Deneysel Kurulum ve Simülasyon Blokları.....	134
5.5.1. Simülasyon blokları	136
5.5.2. Eyleyici ve platform parametreleri	138
5.6. Deneysel ve Simülasyon Sonuçları	139
5.6.1. Deney 1	140
5.6.2. Deney 2	142
5.6.3. Deney 3	144
5.6.4. Deney 4	145
5.6.5. Deney 5	147
5.6.6. Deney 6	149
5.6.7. Deney 7	150
5.7. Sonuç	154
6. STEWART PLATFORM'UN KONTROL ANALİZİ	155
6.1. Giriş	155
6.2. PSO Tabanlı PID Kontrol Tasarımı	169
6.3. PSO Tabanlı FOPID Kontrol Tasarımı.....	172
6.4. PSO Tabanlı Bulanık PID Kontrol Tasarımı	174
6.5. PSO Tabanlı Bulanık FOPID Kontrol Tasarımı	179
6.6. Deneysel Kurulum.....	184
6.7. Deneysel Sonuçlar	190
6.7.1. Yörünge takip deneyi 1	191
6.7.2. Yörünge takip deneyi 2	195

6.7.3. Yörünge takip deneyi 3	198
6.7.4. Yörünge takip deneyi 4	201
6.7.5. Yörünge takip deneyi 5	204
6.7.6. Yörünge takip deneyi 6	207
6.7.7. Yörünge takip deneyi 7	210
6.8. Sonuç	213
7. SONUÇLAR VE ÖNERİLER.....	215
7.1. Giriş	215
7.2. Sonuçlar	216
7.3. Öneriler	219
KAYNAKLAR.....	221
EKLER.....	235
KİŞİSEL YAYINLAR VE ESERLER.....	263
ÖZGEÇMİŞ.....	264

ŞEKİLLER DİZİNİ

Şekil 1.1.	Genel bir Stewart Platform'u	3
Şekil 1.2.	1955'te yapılan ilk Gough Platform lastik test makinesi	7
Şekil 1.3.	Geliştirilmiş Gough Platform lastik test makinesi	7
Şekil 1.4.	Uçuş simülatörü (a) ve Stewart'ın önerdiği paralel mekanizma mimarisi	8
Şekil 1.5.	Moog Base (a), Detapod (b), Stewart Platform (c) ve Motion Chair (d) ...	9
Şekil 2.1.	PID kontrollü geri beslemeli sistem	17
Şekil 2.2.	NINTEGER programın simülink kütüphanesi	20
Şekil 2.3.	Klasik PID (a), Kesir dereceli PID'nin P-I-D düzlemi (b)	24
Şekil 2.4.	$\mu_A(x)$ üyelik fonksiyonu	27
Şekil 2.5.	"Genç" bulanık kümesi (a), "Genç" bulanık kümesinin tümleyeni (b)	29
Şekil 2.6.	"Orta Yaşlı" bulanık kümesi (a), "Genç" ve "Orta Yaşlı" bulanık kümelerinin kesişimi (b)	29
Şekil 2.7.	"Yaşlı" bulanık kümesi (a), "Genç" ve "Orta Yaşlı" bulanık kümelerinin kesişimi ile "Orta Yaşlı" ve "Yaşlı" bulanık kümelerin kesişiminin birleşimi (b)	29
Şekil 2.8.	Kavramsal değişken örneği	30
Şekil 2.9.	Bulanık mantık kontrolörün yapısı	31
Şekil 2.10.	Bulanık PD kontrolör	34
Şekil 2.11.	Bulanık PI kontrolör	34
Şekil 2.12.	İki farklı kural tablosu kullanılarak oluşturulan iki girişli Bulanık PID kontrolör yapısı	36
Şekil 2.13.	İki girişli tek kural tabanlı Bulanık PID kontrolör yapısı	36
Şekil 2.14.	PID kontrolör parametrelerini ayarlayan bulanık kontrolör yapısı	37
Şekil 2.15.	Karma PID kontrolör yapısı	38
Şekil 3.1.	PSO algoritmasının genel işleyiş yapısı	44
Şekil 4.1.	Endüstriyel bir manipülatörün eksenleri ve hareket yönleri	47
Şekil 4.2.	İleri ve ters kinematığın şematik gösterimi	48
Şekil 4.3.	Bağlar üzerinde oluşan kuvvet ve torklar	51
Şekil 4.4.	Kuvvet ve tork yayılımı	51
Şekil 4.5.	Bağ (i) üzerindeki dinamik kuvvetler	52
Şekil 4.6.	İki serbestlik dereceli robot kolu modeli	61
Şekil 4.7.	Bulanık kümelerin başlangıç üyelik fonksiyonları ve kontrol yüzeyleri	63
Şekil 4.8.	Robot kolu için bulanık mantık kontrolörün simülink modeli	64
Şekil 4.9.	PSO için pseudo kodu	64
Şekil 4.10.	PSO ile bulanık kontrol parametrelerin ayarlanma süreci	66
Şekil 4.11.	MRSE (a), MAE (b) ve MMFAE (c) ile optimizasyon sonrası üyelik fonksiyonları	67
Şekil 4.12.	MRSE (a), MAE (b) ve MMFAE (c) ile optimizasyon sonrası bulanık kontrol yüzeyleri	69
Şekil 4.13.	Optimizasyon süreci boyunca 1 nolu Parçacığın (P_1) hareketi	71
Şekil 4.14.	Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için kütle değişiklikleri altında MRSE maliyet fonksiyonlu Bulanık-PSO ve PID-PSO kontrolörlerin performansı	73

Şekil 4.15. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için kütle değişiklikleri altında MAE maliyet fonksiyonlu Bulanık-PSO ve PID-PSO kontrolörlerin performansı	74
Şekil 4.16. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için kütle değişiklikleri altında MMFAE maliyet fonksiyonlu Bulanık-PSO ve PID-PSO kontrolörlerin performansı	75
Şekil 4.17. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için hem genlik hem de faz değişiklikleri altında MRSE maliyet fonksiyonlu Bulanık-PSO ve PID-PSO kontrolörlerin performansı.....	78
Şekil 4.18. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için hem genlik hem de faz değişiklikleri altında MAE maliyet fonksiyonlu Bulanık-PSO ve PID-PSO kontrolörlerin performansı.....	79
Şekil 4.19. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için hem genlik hem de faz değişiklikleri altında MMFAE maliyet fonksiyonlu Bulanık-PSO ve PID-PSO kontrolörlerin performansı.....	80
Şekil 4.20. Düzlemsel robotun ve FOPID kontrolörlerin simülink modeli	84
Şekil 4.21. PSO ile FOPID kontrolör parametrelerin ayarlanma süreci.....	84
Şekil 4.22. Ayarlama işlemi sırasında birinci adımdan (a) son adıma (b) kadar bir parçacığın hareketi	86
Şekil 4.23. Optimizasyon süreci sırasında MRSE maliyet fonksiyonlu FOPID-PSO (a) ve FOPID-GA (b) kontrolörlerin yakınsama oranları ...	88
Şekil 4.24. Kütle değişikliği altında MRSE maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin kontrol sinyali (b,d) ve konumları (a,c)	90
Şekil 4.25. Hem genlik hem de fazın değiştiği durumda MRSE maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin kontrol sinyali (b,d) ve konumları (a,c).....	92
Şekil 4.26. 2π Hz frekansında MRSE (a,b) maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin frekans cevapları.....	93
Şekil 4.27. 3π Hz frekansında MRSE (a,b) maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin frekans cevapları.....	93
Şekil 4.28. MRSE maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin birim basamak cevapları (a,b).....	94
Şekil 5.1. Stewart Platform'u (a) ve kinematik düzenleşimi (b)	105
Şekil 5.2. Tüm platformun (a) ve üst platformun eklem noktaları (b).....	106
Şekil 5.3. Sabit açı setiyle dönme işlemi.....	107
Şekil 5.4. Stewart Platform'unun bir bacağına ait kinematik düzenleşim.....	110
Şekil 5.5. Stewart Platform'unda bir bacağın kinematik ve dinamik düzenleşimi.	119
Şekil 5.6. DC Motor ve motorun dişli mekaniksel kısmı.....	130
Şekil 5.7. Tüm Stewart Platform dinamiğinin simülasyon blok diyagramı.....	134
Şekil 5.8. Dinamik simülasyon algoritması.....	135
Şekil 5.9. Stewart Platform düzeneği.....	135
Şekil 5.10. Referans bacak uzunluğunu hesaplayan "Leg_Trajectory" bloğu	136
Şekil 5.11. Gerçekleşen bacak uzunluğu üreten "Stewart_Platform_Dynamics" bloğu.....	136
Şekil 5.12. Bacak uzunluklarının hesaplandığı "Lengths of Legs" bloğu	137
Şekil 5.13. Stewart Platform'unun simülink modeli.....	137
Şekil 5.14. Eyleyicilerde üretilen torkların simülink modeli	138
Şekil 5.15. Vidalı Bilye (Ball-Screw).....	138

Şekil 5.16. Deneş 1 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması	141
Şekil 5.17. Deneş 2 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması	143
Şekil 5.18. Deneş 3 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması	144
Şekil 5.19. Deneş 4 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması	146
Şekil 5.20. Deneş 5 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması	148
Şekil 5.21. Deneş 6 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması	150
Şekil 5.22. Deneş 7 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması	151
Şekil 6.1. Stewart Platform'un yörünge takip kontrolü için kontrol blok diyagramı	168
Şekil 6.2. PSO ile PID kontrolörün optimizasyonu	170
Şekil 6.3. Optimizasyon süreci boyunca PSO tabanlı PID kontrolörün yakınsama eğilimi	171
Şekil 6.4. PSO ile FOPID kontrolörün optimizasyonu	172
Şekil 6.5. Optimizasyon süreci boyunca PSO tabanlı FOPID kontrolörün yakınsama eğilimi	173
Şekil 6.6. Optimizasyon adımı sırasında μ (*) ve λ (o) için elde edilen optimize değerler	174
Şekil 6.7. Bulanık PID ile kontrol edilen sistemin blok diyagramı	175
Şekil 6.8. Bulanık kontrolör girişlerin ve çıkışın, başlangıç Gauss üyelik fonksiyonları ile bulanıklaştırılması	176
Şekil 6.9. Bulanık kontrolörün optimizasyon öncesi başlangıç kontrol yüzeyi	176
Şekil 6.10. PSO ile bulanık PID kontrolörün optimizasyonu	178
Şekil 6.11. Bulanık PID kontrolörün parametre optimizasyonu boyunca maliyet fonksiyonun yakınsama karakteristiđi	179
Şekil 6.12. Kesir dereceli bulanık PID kontrol yapısı	180
Şekil 6.13. Bulanık kontrolörün hata, kesir dereceli hata deđişimi ve çıkış deđişkenleri için başlangıç Gauss üyelik fonksiyonları	180
Şekil 6.14. Bulanık FOPID kontrolörü için giriş ve çıkışa ait optimizasyon öncesi başlangıç kontrol yüzeyi	182
Şekil 6.15. Önerilen PSO ayarlı bulanık FOPID kontrol sistemi	183
Şekil 6.16. Bulanık FOPID kontrolörün parametre optimizasyonu boyunca maliyet fonksiyonun yakınsama karakteristiđi	184
Şekil 6.17. Uygulama deneyleri için sistemin ana kontrol modeli	185
Şekil 6.18. PID ile kesir dereceli PID'nin Bode diyagramlarının karşılaştırılması	186
Şekil 6.19. Bulanık PID kontrolörün giriş (error ve differror) ve çıkışına (voltage) ait optimize edilmiş nihai üyelik fonksiyonları	187
Şekil 6.20. PSO ile optimize edilen bulanık PID kontrolörün kontrol yüzeyi	188
Şekil 6.21. Bulanık FOPID kontrolörün giriş (error ve differror) ve çıkışına (voltage) ait optimize edilmiş nihai üyelik fonksiyonları	189
Şekil 6.22. PSO ile optimize edilen bulanık FOPID kontrolörün kontrol yüzeyi	189
Şekil 6.23. Deneş 1'e göre kontrol altındaki bacakların yörünge takibi	193
Şekil 6.24. Deneş 1'e göre kontrolörlerin yörünge takip hataları	194

Şekil 6.25. Deney 2'ye göre kontrol altındaki bacakların yörünge takibi	196
Şekil 6.26. Deney 2'ye göre kontrolörlerin yörünge takip hataları	197
Şekil 6.27. Deney 3'e göre kontrol altındaki bacakların yörünge takibi	199
Şekil 6.28. Deney 3'e göre kontrolörlerin yörünge takip hataları	200
Şekil 6.29. Deney 4'e göre kontrol altındaki bacakların yörünge takibi	202
Şekil 6.30. Deney 4'e göre kontrolörlerin yörünge takip hataları	203
Şekil 6.31. Deney 5'e göre kontrol altındaki bacakların yörünge takibi	205
Şekil 6.32. Deney 5'e göre kontrolörlerin yörünge takip hataları	206
Şekil 6.33. Deney 6'ya göre kontrol altındaki bacakların yörünge takibi	208
Şekil 6.34. Deney 6'ya göre kontrolörlerin yörünge takip hataları	209
Şekil 6.35. Deney 7'ye göre kontrol altındaki bacakların yörünge takibi	211
Şekil 6.36. Deney 7'ye göre kontrolörlerin yörünge takip hataları	212

TABLolar DİZİNİ

Tablo 1.1. Seri ve paralel robotların karşılaştırılması.....	11
Tablo 4.1. Bulanık mantık kural atama tablosu.....	63
Tablo 4.2. Üyelik fonksiyon parametrelerin sınır değerleri.....	67
Tablo 4.3. Farklı maliyet fonksiyonları için PID katsayıları.....	72
Tablo 4.4. Herhangi bir bozucu etki olmaksızın kontrolörlerin yapmış olduğu maliyet ve ortalama RMS değerleri	72
Tablo 4.5. Kütle değişikliği altında kontrolörlerin maliyet ve ortalama RMS değerleri	76
Tablo 4.6. Farklı varyanslı gürültü eklendiğinde MRSE fonksiyonuna göre maliyet değerleri	77
Tablo 4.7. Farklı varyanslı gürültü eklendiğinde MAE fonksiyonuna göre maliyet değerleri	77
Tablo 4.8. Farklı varyanslı gürültü eklendiğinde MMFAE fonksiyonuna göre maliyet değerleri	77
Tablo 4.9. Verilen yörünge değişikliği durumunda MRSE fonksiyonuna göre maliyet değerleri	81
Tablo 4.10. Verilen yörünge değişikliği durumunda MAE fonksiyonuna göre maliyet değerleri	82
Tablo 4.11. Verilen yörünge değişikliği durumunda MMFAE fonksiyonuna göre maliyet değerleri	82
Tablo 4.12. FOPID kontrolör parametrelerin sınır değerleri	85
Tablo 4.13. Tasarlanan $FOPI^{\lambda}D^{\mu}$ kontrolörün PSO ile ayarlı parametreleri.....	88
Tablo 4.14. Tasarlanan $FOPI^{\lambda}D^{\mu}$ kontrolörün GA ile ayarlı parametreleri.....	89
Tablo 4.15. Herhangi bir bozucu etki olmaksızın kontrolörlerin göstermiş olduğu maliyet ve ortalama RMS değerleri	89
Tablo 4.16. Kütle değişikliği altında kontrolörlerin maliyet ve ortalama RMS değerleri	91
Tablo 4.17. Farklı varyanslı gürültü eklendiğinde MRSE fonksiyonuna göre maliyet değerleri	91
Tablo 5.1. Eyleyici parametreleri	138
Tablo 5.2. Stewart Platform'un yapısal parametreleri	139
Tablo 5.3. Deney 1'e göre dinamik modellerin maliyet fonksiyon değerleri	142
Tablo 5.4. Deney 2'ye göre dinamik modellerin maliyet fonksiyon değerleri	143
Tablo 5.5. Deney 3'e göre dinamik modellerin maliyet fonksiyon değerleri	145
Tablo 5.6. Deney 4'e göre dinamik modellerin maliyet fonksiyon değerleri	147
Tablo 5.7. Deney 5'e göre dinamik modellerin maliyet fonksiyon değerleri	148
Tablo 5.8. Deney 6'ya göre dinamik modellerin maliyet fonksiyon değerleri	149
Tablo 5.9. Deney 7'ye göre dinamik modellerin maliyet fonksiyon değerleri	152
Tablo 5.10. Yöntemlere göre tüm sistemin maliyet fonksiyon değerleri.....	153
Tablo 5.11. Deneysel robot ve dinamik modellerin açık çevirim hatalarına göre maliyet fonksiyon değerleri	153
Tablo 6.1. Bulanık mantık kural atama tablosu.....	177
Tablo 6.2. Hata, hatanın kesir dereceli türevi ve bulanık çıkış için kural tabanı ...	181

Tablo 6.3. Optimum PID ve FOPID kontrolörlerin kazanç katsayıları	186
Tablo 6.4. Bulanık PID ve bulanık FOPID kontrolörlerin optimum giriş ve çıkış kazanç katsayıları	190
Tablo 6.5. Deney 1'e göre kontrolörlerin performans karşılaştırması.....	194
Tablo 6.6. Deney 2'ye göre kontrolörlerin performans karşılaştırması.....	197
Tablo 6.7. Deney 3'e göre kontrolörlerin performans karşılaştırması.....	200
Tablo 6.8. Deney 4'e göre kontrolörlerin performans karşılaştırması.....	203
Tablo 6.9. Deney 5'e göre kontrolörlerin performans karşılaştırması.....	206
Tablo 6.10. Deney 6'ya göre kontrolörlerin performans karşılaştırması.....	209
Tablo 6.11. Deney 7'ye göre kontrolörlerin performans karşılaştırması.....	212
Tablo 6.12. Tüm deneyler için kontrolörlerin toplam yörünge takip hataları ve kontrol sinyallerin ortalama RMS değerleri	213

SİMGELER DİZİNİ VE KISALTMALAR

B_m	: Motor milinin viskoz sürtünme katsayısı
B_s	: Vidalı bilye viskoz sürtünme katsayısı
$B\text{-}XYZ$: Sabit platformun merkezinin koordinat sistemi
c	: Guassian üyelik fonksiyonunun merkezi
$C(q, \dot{q})$: Coriolis ve merkezkaç (centrifugal) kuvvet matrisi
$D(q)$: Kütle matrisi
${}_a D_t^\alpha$: integro-diferansiyel operatörü
e	: Hata
\dot{e}	: Hatanın türevi
f_i	: Eklem kuvveti
Ge	: Hata değerinin kazanç katsayısı
G_{ce}	: Hata değişim değerinin kazanç katsayısı
G_i	: Çıkış integratörün katsayısı
$G(q)$: Yerçekimi ivme vektörü
GT_i	: Hareketli platform üzerindeki bağlantı noktasının konumu
Gu	: Çıkış katsayısı
$G(X_{p-o})$: Dinamik denklemin yer çekimi kuvvet vektörü
I_{C_i}	: Atalet tensörü
$I_{(mf)}$: Hareketli platformun atalet tensörü
J	: Jacobian matrisi
J_m	: Motor mili atalet momenti
J_s	: Vidalı bilye atalet momenti
K_b	: Elektromotor kuvvet sabiti
K_d	: Türev kazancı
K_i	: İntegral kazancı
K_{Legs}	: Bacakların toplam kinetik enerjisi
K_{Li}	: Her bir bacağın kinetik enerjisi
$K_{Li(rot)}$: Bacağın yöneliminden kaynaklanan kinetik enerji
$K_{Li(trans)}$: Bacağın ötelemesinden kaynaklanan kinetik enerji
K_p	: Orantı kazancı
$K(q, \dot{q})$: Toplam kinetik enerjisi
K_t	: Motor tork sabiti
K_{up}	: Hareketli platformun toplam kinetik enerjisi
$K_{up(rot)}$: Hareketli platformun yöneliminden kaynaklanan kinetik enerji
$K_{up(trans)}$: Hareketli platformun ötelemeden kaynaklanan kinetik enerji

l_1	: Bacağın sabit parçasının uzunluğu
l_2	: Bacağın hareketli parçasının uzunluğu
L	: Lagrange fonksiyonu
L_i	: Bacağın konum vektörü
$\ L_i\ $: Bacak uzunluğu
l_i	: Bağın uzunluğu
m_1	: Bacağın sabit parçasının kütlesi
m_2	: Bacağın hareketli parçasının kütlesi
m_i	: Bağın kütlesi
M_{Legs}	: Bacakların kütle matrisi
M_{up}	: Hareketli platformun kütle matrisi
$M(X_{p-o})$: Dinamik denklemin kütle matrisi
n_i	: Eklem torku
P	: Konum vektörü
P_i	: Parçacığın konumu
P_{Legs}	: Bacakların toplam potansiyel enerjisi
$P(q)$: Toplam potansiyel enerjisi
P_{up}	: Hareketli platformun toplam potansiyel enerjisi
q	: Robotun genelleştirilmiş koordinatı
\dot{q}	: Genelleştirilmiş koordinatın zaman göre birinci türevi
R	: Dönme matrisi
r_{base}	: Sabit platformun yarıçapı
r_{up}	: Hareketli platformun yarıçapı
r_1, r_2	: 0 ile 1 arasındaki sayılar
t_f	: Simülasyon süresi
T_j	: Hareketli platform üzerindeki bağlantı noktasının konumu
\dot{T}_j	: Hareketli platform üzerindeki bağlantı noktasının hızını
$T-xyz$: Hareketli platformun merkezinin koordinat sistemi
u_i	: Bacağın doğrultusunu gösteren birim vektör
\dot{V}	: Doğrusal ivme
V_i	: Parçacığın hızı
\vec{V}_{T_j}	: Bacakların uç hızlarını gösteren vektör
$V(X_{p-o}, \dot{X}_{p-o})$: Dinamik denklemin Coriolis ve merkezkaç matrisi
X_{p-o}	: Kartezyen uzayda hareketli platformun konumu ve yönelimi
\dot{X}_{p-o}	: Kartezyen uzayda hareketli platformun lineer ve açısal hızları
\ddot{X}_{p-o}	: Kartezyen uzayda hareketli platformun lineer ve açısal ivmesi
λ	: İntegratör derecesi
μ	: Türev derecesi
$\mu_A(x)$: Bulanık A kümesi için üyelik fonksiyonu

$\Gamma(\cdot)$: Euler Gama fonksiyonu
χ	: Sınırlayıcı faktör
τ	: Genelleştirilmiş kuvvet ya da tork
τ_{mi}	: Motor tarafında üretilen tork
$\vec{\Omega}_{up(mf)}$: Hareketli platformun açısız hızı
$\vec{\Omega}_{up(ff)}$: Hareketli platformun sabit alt platform cinsinden açısız hızı
φ_1, φ_2	: Pozitif öğrenme katsayıları
σ	: Gaussian üyelik fonksiyonun genişliği
θ_b	: Her bir alt eklem noktasının diğer alt eklem noktasıyla yaptığı açı
θ_i	: Eklemin konumu
$\dot{\theta}_i$: Eklemin hızı
$\ddot{\theta}_i$: Eklemin ivmesi
θ_{mi}	: Motor milinin açısız konumu
$\dot{\theta}_{mi}$: Motor milinin açısız hızı
$\ddot{\theta}_{mi}$: Motor milinin açısız ivmesi
θ_p	: Her bir üst eklem noktasının diğer üst eklem noktasıyla yaptığı açı
ρ	: Vidalı bilye dönüş adımı sayısı
$\dot{\omega}$: Açısız ivme
$\omega_i; \omega_h$: Alt ve üst frekans aralığı

Kısaltmalar

FOPID	: Fractional Order PID (Kesir Dereceli PID)
GA	: Genetic Algorithm (Genetik Algoritma)
gbest	: Global Best (Küresel En iyi Parçacık)
MAE	: Mean of Absolute Magnitude of the Error (Ortalama Mutlak Hata)
MMFAE	: Mean Minimum Fuel and Absolute Error (Ortalama En küçük Enerji ve Mutlak Hata)
MRMS	: Mean of RMS (Ortalama RMS)
MRSE	: Mean Root of Squared Error (Karesel Hataların Karekök Ortalaması)
pbest	: Personal Best (Yerel En iyi Parçacık)
PID	: Proportional (Oransal) + Integral (Tümlev) + Differential (Türev)
PSO	: Particle Swarm Optimisation (Parçacık Sürü Optimizasyonu)
RMS	: Root Mean Square (Karekök Ortalama)
SIM-A	: Yöntem A Dinamik Modeli
SIM-B	: Yöntem B Dinamik Modeli

SERİ ve PARALEL ROBOTLARDA PARÇACIK SÜRÜ OPTİMİZASYONU İLE YÖRÜNGE KONTROLÖRÜ TASARIMI

ÖZET

Bu tezde, seri ve paralel robotlar için yörünge kontrolü sağlayan parçacık sürü optimizasyon (PSO) tabanlı farklı kontrolörlerin tasarımı gerçekleştirilmiştir. Bu kontrolörleri test etmek için seri robot olarak iki serbestlik dereceli (DOF) düzlemsel robot ve paralel robot için 6 DOF Stewart Platform'u seçilmiştir. Bu manipülörlerin kinematik ve dinamik analizleri, simülasyon modeli ve yörünge kontrolü ayrıntılı bir şekilde incelenmiştir.

2 DOF robotun yörünge kontrolü için sırasıyla PID kontrol, bulanık kontrol ve kesir dereceli PID (FOPID) kontrol yapıları geliştirilmiştir. Bu kontrolörlere ait parametreler PSO algoritması kullanılarak optimize edilmiştir. Birinci simülasyon çalışmasında, eklem uzayı yörünge takip kontrolü için robota PID-PSO ve Bulanık-PSO kontrolörleri uygulanarak simülasyon deneyleri gerçekleştirilmiştir. İkinci çalışmada ise yörünge kontrolü için PSO ve genetik algoritma (GA) teknikleri ile FOPID kontrolörüne ait parametreler optimize edilmiştir. Ayrıca kontrolörlerin optimizasyonunda, maliyet fonksiyonların etkilerini göstermek amacıyla her iki çalışma için 3 farklı maliyet fonksiyonu kullanılmıştır. Optimize edilen kontrolörlerin performanslarını karşılaştırmak ve bu kontrolörlerin sağlamlığını test etmek için sistemin model parametreleri değiştirilmiş, sisteme değişik varyanslı gürültü eklenmiş ve farklı yörüngeler verilmiştir. Sonuç olarak, hem farklı yörünge hem de değişik tipteki ve büyüklükteki bozuculara karşı robotun yörünge kontrolü için PSO ile parametreleri ayarlanan Bulanık ve FOPID kontrol yapıları ile sağlam kontrolörler tasarlanabilmiştir. 6 DOF Stewart Platform'un dinamik modelini doğru bir şekilde elde etmekte en önemli adım Jacobian matrisinin elde edilmesidir. Burada, Jacobian matrisi iki farklı şekilde elde edilmiştir. Ayrıca bacaklarda yer alan eyleyicilerin mekanik ve elektriksel dinamiği çıkarılarak manipülörün katı gövde dinamiğine eklenmiştir. İki farklı Jacobian matrisine göre çıkarılan tüm sistemin dinamiği MATLAB-SIMULINK ortamında modellenip, fiziksel robot üzerinde model doğruluğu için yörünge deneyleri yapılarak sonuçlar değerlendirilmiştir. Bu sonuçlar doğrultusunda, elde edilen dinamik matematiksel model, yüksek doğrulukla fiziksel sistemin davranışını karşılamaktadır. Manipülörün eklem esaslı yörünge kontrolü için geleneksel PID, FOPID, bulanık PID ve bulanık FOPID kontrol yapıları geliştirilmiştir. Dinamik model kullanılarak, PSO ile bu kontrolörlere ait parametrelerin en iyi değerleri bulunmuştur. Optimize edilen kontrolörler ile fiziksel Stewart Platform manipülatörü üzerinde yörünge takip deneyleri yapılarak kontrol performansları karşılaştırılmıştır. Bu karşılaştırma sonucunda en iyi sonuçları sırayla bulanık FOPID, bulanık PID, FOPID ve PID sağlamıştır. Özellikle bulanık FOPID, yörünge kontrolünde oldukça yeni bir yaklaşımdır.

Anahtar Kelimeler: Bulanık Kontrol, Dinamik ve Kinematik Model, Kesir Dereceli PID kontrol, Parçacık Sürü Algoritması, PID Kontrol, Robotik Manipülatör.

TRAJECTORY BASED CONTROLLER DESIGN USING PARTICLE SWARM OPTIMISATION FOR SERIAL AND PARALLEL ROBOTS

ABSTRACT

In this thesis, the design of the different controllers based on particle swarm optimisation (PSO) algorithm was realized for trajectory control of serial as well as parallel robots. In order to test these controllers, a 2 DOF planar robot and 6 DOF Stewart Platform were chosen as serial and parallel robot respectively. The kinematic and dynamic analyses, simulation and trajectory control of these manipulators were explored in detail.

The control structures based on PID controller, fuzzy controller and fractional order PID (FOPID) controller were respectively developed for trajectory control of the planar robot. The parameters of these controllers were optimized by using PSO. In the first simulation study, for the joint trajectory tracking control of the robot using the PID-PSO controller and the Fuzzy-PSO controller, the simulation experiments were carried out. In the second study, all of the parameters related with the FOPID controller were tuned using PSO and genetic algorithm (GA) for trajectory control. Moreover, in order to see effects of the cost functions on the controller parameter optimisation, the three different cost functions were used in both simulation studies. In order to test the robustness and compare the performance of the tuned controllers, the model parameters and the given trajectory were changed and the white noise was added to the system. As a result, the robust controllers were designed for trajectory control of the planar robot using fuzzy and FOPID control schemes tuned with PSO to cope with not only the different types and magnitudes of disturbances, but also the different trajectories. Most important is the derivation of Jacobian matrix for obtaining accurate dynamic model of 6 DOF Stewart Platform. Jacobian matrix was derived here in two different ways. The mechanical and the electrical dynamics of the actuators were also added to the rigid body dynamics of the manipulator. The dynamic equations including rigid body and actuator dynamics were simulated in MATLAB and experimentally verified on physical system, and the results were evaluated. Based on these results, the resulting dynamic model has met the behavior of the physical system with high accuracy. Traditional PID, FOPID, fuzzy PID and fractional order fuzzy PID (fuzzy FOPID) techniques were developed for the trajectory control based joint of the platform. Using the dynamic model, the optimum parameters of the proposed controllers were obtained using PSO. The trajectory tracking experiments of the controllers optimized by PSO were implemented on the experimental Stewart Platform, and the control performances of the controllers were compared. In consequence of this comparison, the best results were provided by the fuzzy FOPID, fuzzy PID, FOPID and PID controllers, respectively. Especially, the fuzzy FOPID controller is a highly new approach for trajectory control.

Keywords: Fuzzy Control, Dynamic and Kinematic Model, Fractional Order PID Control, Particle Swarm Optimisation, PID Control, Robotic Manipulator.

GİRİŞ

Bu çalışmada, seri ve paralel robotların kinematik ve dinamik analizlerinin ayrıntılı olarak yapılması, yörünge takip kontrolörlerin geliştirilmesi ve deneysel robot üzerinde etkili biçimde uygulanması amaçlanmıştır. Bu tezin kapsamında sunulan başlıca katkılar aşağıda liste halinde verilmektedir.

- i. Eklem uzayı hızları ile kartezyen uzayı hızları arasındaki ilişkiyi veren Stewart Platform'un Jacobian matrisi, iki farklı şekilde çıkarılmıştır. Jacobian matrisinin doğru bir şekilde elde edilmesi, hem eklem hem de kartezyen uzayında ters dinamik modelin doğru şekilde elde edilmesi açısından önemlidir.
- ii. 6-6 Deneysel Stewart Platform yapısına ait tam dinamik model (katı gövde dinamiği ve eyleyici dinamiği), Lagrange formülasyonu kullanılarak iki farklı Jacobian matrisine göre türetilmiştir. Elde edilen iki farklı dinamik modelin doğruluğu, yapılan deneylerle gösterilmiştir. İleri seviye kontrolü için doğru bir dinamik model gerekir.
- iii. Stewart Platform manipülatörün kontrol performansını, özellikle eklem esaslı yörünge takip performansını geliştirmek için geleneksel PID, kesir dereceli PID, bulanık PID ve kesir dereceli bulanık PID kontrol yapıları geliştirilmiştir. Ayrıca, elde edilen tam dinamik model kullanılarak etkili bir arama tekniği olan parçacık sürü optimizasyon algoritması ile bu kontrol yapılarına ait parametrelerin en iyi değerleri bulunmuştur.
- iv. Optimize edilen kontrolörlerin performansı, bacaklara ait yörünge takip hatalarına ve kontrol sinyallerine göre deneysel olarak karşılaştırılmıştır. Bu karşılaştırma sonucu en iyi cevabı, kesir dereceli bulanık PID kontrol yapısı üretmiştir.

Bu tezin içerik çerçevesi 7 bölüm şeklinde organize edilerek aşağıdaki gibi oluşturulmuştur.

Bölüm 1’de, Stewart Platform manipülatörün tarihçesi, paralel manipülatörler ve seri manipülatörler ile karşılaştırılması, paralel manipülatörlerde uygulanan kontrol çeşitleri özet olarak verilmektedir. Ayrıca, tezin amaç ve kapsamı da yer almaktadır.

Bölüm 2’de, Stewart Platform manipülatörün eklem esaslı yörünge takip kontrolü için tasarlanacak olan PID, Kesir Dereceli PID (FOPID) ve Bulanık Mantık kontrolörlerin literatür taraması ve bunlar hakkında ayrıntılı bilgiler verilmektedir. Ayrıca, kontrolörlere ait parametrelerin özelliklerinden, kontrole etkisinden ve doğrusal veya doğrusal olmayan sistemlerdeki başarımından söz edilmektedir.

Bölüm 3’te, Stewart Platform manipülatörün kontrolü için tasarlanan kontrolörlere ait parametrelerin ayarlanmasında kullanılacak parçacık sürü optimizasyon (PSO) algoritması sunulmaktadır.

Bölüm 4’te seri manipülatörlerin kinematiği ve dinamiği hakkında literatür çalışmasına ve seri robot kolunun dinamik modelini çıkarmada kullanılan yöntemlere ayrıntılı olarak değinilmektedir. Ayrıca, iki serbestlik dereceli düzlemsel bir robot kolunun dinamik denklemleri verilerek iki farklı yörünge kontrolü simülasyon çalışması sunulmaktadır. Birinci simülasyon çalışmasında, PSO tabanlı bulanık mantık ve PSO tabanlı PID kontrol yöntemleri ile eklem uzayında yörünge takip kontrolü yapılmaktadır. İkinci simülasyon çalışmasında, iki farklı optimizasyon tekniği olan PSO ve gerçek kodlu genetik algoritma (GA) kullanılarak bu düzlemsel robotun yörünge takip kontrolü için kesir dereceli PID kontrolörlerin parametrelerinin optimizasyonu yapılmaktadır. Bunun yanında, tasarlanan kontrolörlerin yörünge performanslarının ayrıntılı olarak karşılaştırılmasına yer verilmektedir.

Bölüm 5’te, 6 serbestlik dereceli Stewart Platform manipülatörün ters kinematik analizi ve Lagrange formülasyonu kullanılarak manipülatöre ait kapalı formdaki katı gövde dinamik denklemlerin elde edilmesi gösterilmektedir. Ayrıca, Jacobian matrisinin iki farklı şekilde elde edilmesine yer verilmektedir. Bunun yanında,

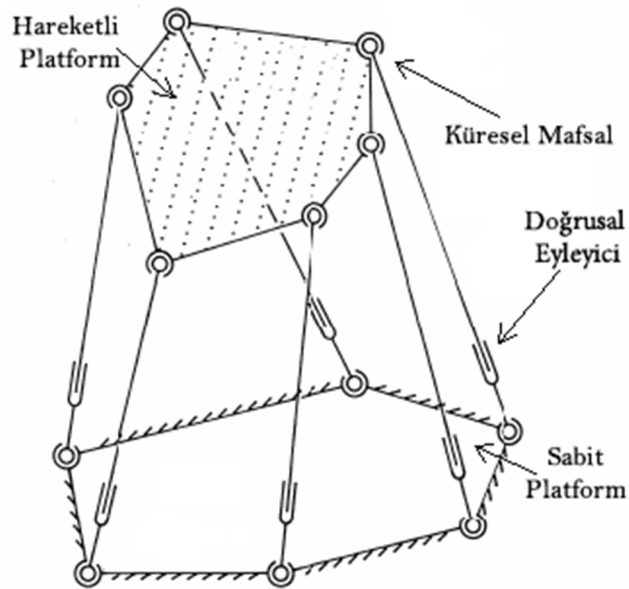
bacaklarda yer alan eyleyicilerin mekanik ve elektriksel dinamiklerin modelleme içine katılmasından bahsedilmektedir. Son olarak, iki farklı Jacobian matrisine göre çıkarılan tüm sistemin dinamiği MATLAB-SIMULINK ortamında modellenerek fiziksel robot üzerinde yapılan model doğrulama deneylerine ve sonuçlarına yer verilmektedir.

Bölüm 6'da, literatürde yapılan çalışmalara değinilerek Stewart Platform manipülatörün kontrol performansını, özellikle eklem esaslı yörünge takip performansını geliştirmek için geleneksel PID, kesir dereceli PID (FOPID), bulanık PID (Fuzzy PID) ve bulanık kesir dereceli PID (Fuzzy FOPID) kontrol yapılarının geliştirilmesine yer verilmektedir. Ayrıca, Stewart Platform manipülatörün tam dinamik modeli kullanılarak PSO algoritması ile bu kontrolörlere ait parametrelerin en iyi değerlerinin bulunması gösterilmektedir. Bunun yanında, PSO tabanlı bu kontrolörlerin eklem esaslı yörünge kontrolüne dayanan yörünge takip doğruluğunu, sağlamlığını ve kararlılığını değerlendirmek amacıyla gerçek robot üzerinde yapılan deneylere yer verilmektedir. Son olarak, yapılan her bir deneyde, bu kontrol yapıları altındaki deneysel robotun gerçekleştirdiği konumlar, bacaklarda meydana gelen konum hataları ve kontrol sinyalleri temel alınarak performans karşılaştırması yapılmaktadır.

1. GENEL BİLGİLER

1.1. Giriş

Robotik alanında, geleneksel seri manipülatörlere göre paralel manipülatörler, yüksek doğruluk, yüksek sağlamlık, yüksek hızda çalışma ve yüksek yük kapasitesine sahip olduğundan bunlara olan ilgi son zamanlarda artmaktadır. Bu üstünlüklerinden dolayı, paralel manipülatörler; hava taşıtı simülatörleri, mikro konumlandırma, hafif metal işleme, tıbbi operasyonlar ve petrol platformları gibi birçok uygulamada yer almaktadır. En yaygın paralel manipülatörlerden birisi de Stewart Platform'udur [1]. Literatürde bu manipülatöre Stewart Platform'u denildiği gibi, Gough'nun önerisiyle tam paralel hal alan manipülatöre Gough-Stewart platformu da denmektedir. Paralel bir manipülatörün uzayda 3 yönde öteleme 3 yönde de dönme hareketi yaparak uçuş simülatörü olarak kullanma fikrini ortaya ilk atan kişi Stewart olduğundan Stewart Platform Manipülatörü (SPM) ismi daha yaygındır [2].



Şekil 1.1. Genel bir Stewart Platform'u [2]

Şekil 1.1’de gösterildiği gibi Stewart Platform’u, her iki ucunda küresel mafsal veya bir ucunda küresel diğer ucunda üniversal mafsal bulunan altı adet bacak ile birbirine bağlanmış, biri sabit diğeri hareketli iki adet platformdan oluşur.

Her bacak, prizmatik bir eklemlerle birbirine bağlantı yapılan bir silindir ve bir pistondan oluşur. Bacağın üst kısmı küresel eklemlerle hareket eden platforma (üst tabaka), alt kısmı ise evrensel eklemlerle sabit platforma (alt tabaka) bağlanır. Her prizmatik eklem, doğrusal vidalı bilye (jaklar), dişli ve kasnak gibi güç iletim elemanları vasıtasıyla DC motor, doğrusal motor ya da hidrolik eyleyici tarafından sürülür.

1.2. Robotların Sınıflandırılması

Çeşitli kriterler kullanılarak robotlar geniş bir şekilde sınıflandırılabilir. Tsai [2], robotların serbestlik derecesine (degree of freedom, DOF), kinematik yapısına, sürme teknolojisine, çalışma uzayı geometrisine ve hareket kontrol karakteristiğine dayalı sınıflandırma yapmıştır.

Serbestlik derecesi açısından, uzamsal robotun çalışma uzayı genel olarak 6 boyutludur ($x, y, z, \phi, \psi, \theta$). Bu nedenle, 6 serbestlik dereceli görevleri yerine getirmek için en az 6 boyutlu eklem uzayına ihtiyaç duyulur. Böyle bir robot, genel amaçlı bir robot olarak sınıflandırılır. Eğer eklem uzayının boyutu 6’dan az ise robot eksik eklemlidir (deficient), 6’dan fazla ise robotun eklemleri gereğinden fazladır (redundant).

Kinematik yapı açısından robotlar, yapısal topolojilerine dayalı olarak sınıflandırılır. Eğer robotun kinematik yapısı açık zincir biçiminde ise seri robot olarak adlandırılır. Seri robotlara endüstride en çok kullanılan PUMA örnek olarak gösterilebilir. Yapısal olarak mekanik kapalı döngü zincirlere sahip olan robotlar, paralel robotlar olarak adlandırılır. Bu manipülatörlere örnek olarak Stewart Platform’u gösterilebilir.

Sürme teknolojisi açısından robotlar, eyleyici elemanlarına (actuator) dayalı sınıflandırılabilir. Bunlar, elektrik, hidrolik ve pnömatik gibi elemanlardır. İşlem

hızına ve yük kapasitesine dayalı olarak bu elemanlar belirlenir. Örneğin 6 serbestlik dereceli bir paralel robotta, daha ağır yük taşıyabilmek için hidrolik silindirik eyleyiciler kullanılır.

Temel tanım olarak robotların çalışma uzayı, uç işlevcinin (end-effector) erişebildiği uzayın hacmidir. Aynı zamanda robotun “görev uzayı” olarak da ifade edilir. Örnek olarak, kartezyen robotlar (ilk üç bağın tamamı prizmatik eklem), üç ötelemeli çalışma uzayı oluşturur. Silindirik robotlar (ilk bağ döner, ikinci ve üçüncü bağ prizmatik eklem) ve küresel robotlar (ilk iki bağ döner, üçüncü bağ prizmatik eklem), silindirik ve küresel koordinatlar kullanarak ifade edilebilen çalışma uzayına sahip robotlardır. Paralel manipülatörde çalışma uzayı ise hareketli platformun ağırlık merkezine yerleştirilen koordinat sisteminin merkezinin ulaşabileceği bölge diye tanımlanır.

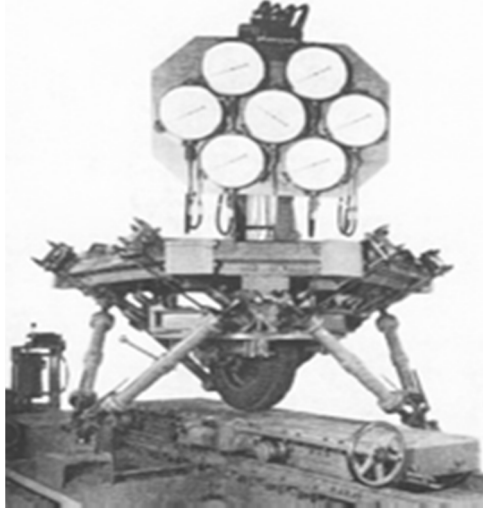
Son olarak, hareket kontrol karakteristiği açısından robotlar, noktadan noktaya hareket ya da sürekli yörünge hareketine dayalı olarak sınıflandırılır. Noktadan noktaya harekete, nokta kaynağı, tutma ve yerleştirme, yükleme ve boşaltma şeklindeki uygulamalar örnek olarak verilebilir. Sürekli yörünge hareketi için spreyci boyama, ark kaynağı ve yapıştırma gibi uygulamalar gösterilebilir.

1.3. Paralel Robotlar

Paralel robotlar, hareketli platformu sabit platforma en az iki noktada birbirinden bağımsız kinematik bağlantı elemanlarıyla bağlayan kapalı çevrimli yapılardır. Bu tür kapalı çevrimli kinematik yapılardan en çok kullanım alanı bulan Stewart'ın geliştirdiği mekanizmadır.

Paralel robotlar ilk kez ortaya çıkışından itibaren birçok kullanım alanı bulmuştur. Gough, 1947'de Dunlop lastik endüstrisi için lastiklerin aşınması ve yıpranmasını test edecek altı serbestlik derecesine sahip bir paralel mekanizmanın tasarımını ve temel ilkelerini sunmuştur. Dunlop firması, 1955'te Şekil 1.2'de görüldüğü gibi Gough'un tasarladığı işlevsel bir prototip yapmıştır. Bu paralel mekanizma, birbirine 6 doğrusal prizmatik eyleyiciler ile bağlanmış biri sabit diğeri hareketli olmak üzere 2

platformdan oluşmaktadır. Bu mekanizmanın hareketli platformu, bir top ve soket eklem ile prizmatik bağlara bağlanır. Sabit platform ise evrensel bir eklem ile her bir bağa bağlanır. Yeni konum ve yönelimleri elde etmek için hareketli platform, bağlantı vidaları el ile döndürülmesiyle ayarlanır. Lastik, hareketli platforma yerleştirilir ve bir taşıma bandı üzerinde sürülür [3].



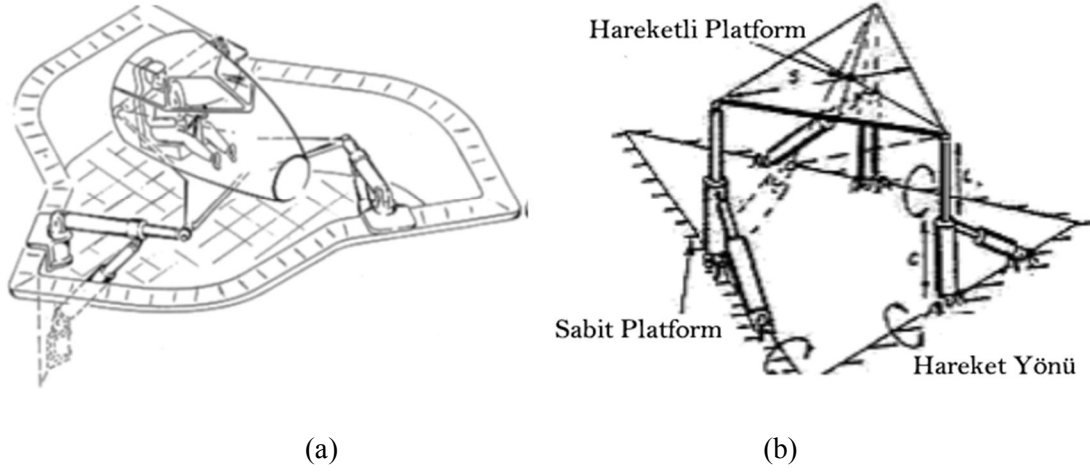
Şekil 1.2. 1955'te yapılan ilk Gough Platform lastik test makinesi [3]

2000 yılında, hareketli platformun konum ve yönelmesini otomatik olarak ayarlayabilen bilgisayar destekli bir arabirim kullanılarak, Dunlop lastik test makinesinin yeni bir versiyonu geliştirildi. Şekil 1.3'te, hâlâ işlevsel olan ve günümüzde de kullanılan Dunlop lastik test makinesi gösterilmektedir.



Şekil 1.3. Geliştirilmiş Gough Platform lastik test makinesi

1965 yılında Stewart, altı serbestlik derecesine sahip bir paralel mekanizma önermiştir. Makalesinde uçuş simülatörleri, okyanusta petrol çıkarma platformları ve çoklu kablo vinçlerini içeren birkaç paralel robotik tasarımını geliştirmiştir. Şekil 1.4.a'da Stewart'ın önerdiği uçuş simülatör tasarımı görülmektedir. Stewart tarafından önerilen mekanizma günümüzde Stewart Platform'u olarak anılan mekanizmadan biraz farklıdır. Ortaya atılan mekanizma 6 serbestlik dereceli, her biri zeminle bağlantılı 6 motora sahip bir mekanizmaydı. Üçgen biçimli hareketli üst platform küresel mafsallarla bacaklara bağlıydı. Stewart'ın tasarladığı paralel mekanizma Şekil 1.4.b'de gösterilmektedir.



Şekil 1.4. Uçuş simülatörü (a) ve Stewart'ın önerdiği paralel mekanizma mimarisi [1]

Stewart, birkaç yıl önce yapılan Gough platformunun farklı tasarımını önerdi. Robotik alanında Stewart Platform tasarımı fazla ilgi görmezken Stewart'ın önerdiği tasarımları da kapsayan Gough platformun çeşitli tasarımları çok büyük ilgi görmüştür ve günümüzde kullanılmaktadır. Gough-Stewart Platform mekanizması, biri hareketli diğeri sabit olmak üzere iki platformdan oluşmakta ve 6 adet prizmatik bağ ile paralel olarak hareket ettirilmektedir.

Stewart Platform'u, kullanılan eklem türlerine göre ve bağ sayılarına göre isimlendirilir. Eklem tiplerine baktığımızda, U üniversal, P prizmatik, R dönel, C silindirik, S ise küresel şeklindeki mafsallardır. Örneğin, robotun prizmatik eklemlere sahip bacakları, alt platforma üniversal, üst platforma küresel mafsallarla bağlı ise bu mekanizma UPS eklemlerli Stewart Platform Manipülatörü (SPM) olarak adlandırılır.

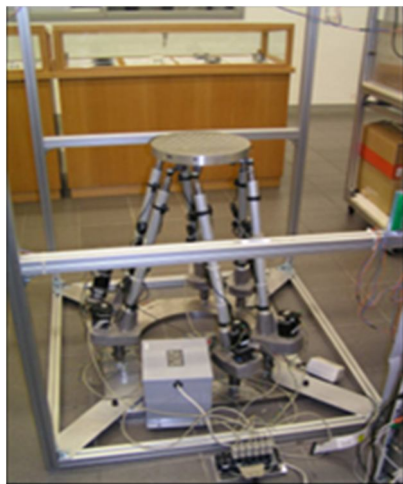
Robot manipulatörleri alanında literatüre bakıldığında, paralel manipulatörlerin araştırılması, geliştirilmesi ve Gough-Stewart Platform yapısının en önemli çalışma konuları olduğu görülmektedir. Çünkü araştırmacılar, Stewart tarafından önerilen paralel mekanizmaları birçok değişik uygulama alanlarında kullanılabileceğinin farkına varmışlardır. Endüstriyel tip Gough- Stewart Platform'u ve bazı prototip paralel mekanizmalar son yıllarda geliştirilmiştir. Gough-Stewart Platform manipulatörlerinin seri manipulatörlere göre daha üstün yeteneklere sahip olmaları ve hava taşıtı simulatörleri, mikro konumlandırma, hafif metal işleme, tıbbi operasyonlar, kesme, şekil verme ve montaj gibi işlerde kullanılması, bu tip mekanizmaların üretimdeki özel önemini göstermektedir. Şekil 1.5'te birçok uygulamalarda yararlanılan paralel manipulatör örnekleri verilmektedir.



(a)



(b)



(c)



(d)

Şekil 1.5. Moog Base (a), Detapod (b), Stewart Platform (c) ve Motion Chair (d) [4]

Şekil 1.6.a'da, otomobil hareketini simüle etmek için kullanılabilen ayrıca eğlence sektöründe gerçek hareketi simüle etmek için de kullanılabilen bir hareket simülatörü gösterilmektedir. Şekil 1.6.b'de, sayısal olarak kontrol edilen yer bulucu noktalar ile kaynak yapmak ve çerçeveleme sırasında araba iskelet kısımlarının geometrisini tanımlamak için İsveç'te ABB Corporate Research'de geliştirilmiş bir Deltapod gösterilmektedir. Şekil 1.6.c'de, bir NC makinesinde metal kaldırma işlemleri için bir destek aracı olarak ya da montaj işlemi için parçaları konumlandırmada kullanılabilen National University of Singapore'de geliştirilmiş bir Stewart Platform'u gösterilmektedir. Şekil 1.6.d'de, Almanya'da Traffic Sciences GmbH için Wuerzburg enstitüsünde geliştirilmiş bir hareketli sandalye gösterilmektedir.

1.4. Paralel ve Seri Robotların Karşılaştırılması

Paralel manipülatörler, geleneksel seri manipülatörlere göre birçok üstünlüğe sahip olmalarına karşın birçok sakıncaları da vardır. Sonuçta her iki sistemin de avantajları ve dezavantajları vardır. Seri manipülatörlerin uzanabildikleri mesafe ve çalışma uzayları, paralel manipülatörlere göre daha yüksektir. Fakat dinamik özellikleri karşılaştırıldığında paralel manipülatörlerin daha üstün oldukları görülür. Tekillikler, çalışma uzayı içinde mekanizmanın kontrol edilemediği konumlar şeklinde ifade edilir. Seri manipülatörlerde, serbestlik derecesi azaldığında ve kısmi kilitlenmeler meydana geldiğinde teklik oluşur. Seri manipülatörlerin açık döngülü yapılarından dolayı eyleyici hataları uca doğru toplanarak büyür. Paralel manipülatörlerde bağlardan yalnızca bir kaç eyleyiciler tarafından hareket ettirilir. Eyleyiciler tabana monte edilmiş olduğundan hareketli parçaların ağırlığı da düşüktür. Böylece, kapalı döngü yapılarından dolayı eyleyici hataları uca doğru eklenerek büyüzmez. Öte yandan, seri manipülatörler fazla katı (rigid) değildir. Seri manipülatörlerde, zemine bağlı bağlardan uca doğru tüm bağlar, hem kendinden önceki bağ ve motorların ağırlığını taşıyabilecek kadar büyük olmalı, hem de istenilen hassasiyeti sağlamalıdır. Bu sebeple dinamik performansta düşüş meydana gelir. Seri manipülatörler ağır yüklerde ve hızlı titreşimlerde burkulma eğilimindedirler.

Tablo 1.1. Seri ve paralel robotların karşılaştırılması

Özellik	Seri	Paralel
Yük Kapasitesi	Düşük	Yüksek
Sağlamlık	Düşük	Yüksek
Çalışma Uzayı	Geniş	Küçük
Tekillik	Az	Karmaşık
Kinematik ve Dinamik Analizi	Kolay	Zor
Ataleti	Düşük	Yüksek
Yeni düzenleşimi	Kolay	Zor

Tablo 1.1’de, seri ve paralel robotlar arasındaki genel karşılaştırma özetlenmektedir. Sonuç olarak, kapalı döngü yapıdan dolayı paralel manipülatörler, aynı boyut ve ağırlıktaki seri manipülatörlere göre daha sağlamdır ve 6 eyleyicinin de kuvvet bileşenleri eklendiği için çıkış kuvveti/manipülatör ağırlığı oranı seri manipülatörlerden bir derece yüksektir. Ancak çalışma uzayının darlığı, mekanik tasarımının zorluğu, ileri kinematik ve dinamik hesaplamalar ile kontrol algoritmalarının karmaşıklığı, seri manipülatörlere göre zayıf olan özellikleridir.

1.5. Paralel Manipülatörlerin Analizi

Paralel manipülatörler ile ilgili kinematik analiz, dinamik analiz ve kontrol gibi birçok araştırma konusu vardır.

1.5.1. Kinematik analiz

Paralel manipülatörlerin kinematik analizi incelenirken temel iki kinematik problem ile ilgilenilir. Bunlar, ters ve ileri kinematik analizlerdir. Bu kinematik analizler, manipülatörlerin çalışma uzayını bulmak ve benzetimini yapmak için kullanılır. Paralel manipülatörlerde ters kinematik problem, ileri kinematik probleme göre daha kolaydır. Seri manipülatörlerde ise ileri kinematik problem daha kolayken ters kinematik problem daha zordur. Paralel manipülatörlerin ters kinematik probleminde konum vektörü ve dönme matrisi verilir, bağ (bacak) uzunlukları bulunur. Bağlantı noktalarının konumundan dolayı hareketli platformun yönelimi ve konumunun bilinmesi, platformun bağ uzunluklarının kolay bir şekilde bulunmasını sağlar. Diğer taraftan, bir paralel manipülatörün ileri kinematiği, bacak uzunlukları bilindiğinde hareketli platformun konumu ve yönelimini belirler. Stewart Platform ileri kinematik probleminde, çözülmesi son derece zor ve lineer olmayan birçok denklem vardır.

Birçok arařtırmacı bu denklemlerin çözümleri üzerinde çalıřmıřlardır. Analitik olarak ya da sayısal olarak çözümleri bulmaya çalıřmıřlar ve bir takım metotlar geliřtirmeyi denemiřlerdir. Raghavan [5], “sürekli” metot ile ileri kinematik problemini çözebilmiř ve de Stewart Platform’un 40 ileri kinematik çözümleri olduđunu göstermiřtir.

1.5.2. Dinamik analiz

Bir manipülatörün istenilen görevi yerine getirirken gerekli kuvvet ve momentleri tanımlamak için hareketin dinamik denklemleri kullanılır. Bu denklemler, ileri ya da ters dinamik řeklinde temsil edilir. İleri dinamik; eklem konumu, hızı ve kuvvet/torkları verildiđinde uç iřlevcinin ivmesinin hesaplanmasına denir. İleri dinamik, doğrusal kontrol ve robot manipülatörlerin model tabanlı kontrolü gerçekleştirilirken kullanılır. Ters dinamik ise eklem konumu, hızı ve ivmeleri verildiđinde eklem kuvvet/torklarının hesaplanmasına denir. Ters dinamik özellikle hesaplanan tork kontrolünde (Computed Torque Control) kullanılır. Manipülatörün gerçek zamanlı kontrolünde manipülatör dinamiđinin verimli olarak hesaplanması çok önemlidir.

Sonuç olarak, ileri dinamikte eyleyici kuvvetler verilerek manipülatör hareketi aranır. Bu analiz benzetim için kullanılır. Ters dinamik analizinde manipülatör hareket ettirildiđinde eyleyiciler üzerindeki kuvvet bulunur. Bu analiz de kontrol için kullanılır.

1.6. Kontrol

Bir manipülatörün kontrolü, robotik alanında önemli bir yer teřkil eder. Yük deđiřimleri, çalıřma düzeni deđiřimleri ve de sistem belirsizlikleri, kontrol problemini önemli kılmaktadır.

Paralel manipülatörlerin kontrolü, istenilen bir görev veya tanımlanan bir yörünge için hareketli platformun ađırlık merkezinin konum ve yöneliminin doğrusal eyleyiciler vasıtasıyla kontrol edilmesi řeklinde ifade edilebilir. Böylece verilen bir

yol üzerinde veya bir görev doğrultusunda hareketli platformun yer deęiřtirmesi saęlanır.

Stewart Platform manipölatörlerin kontrolünde genel olarak iki kontrol stratejisi kullanılır. Birincisi, eklem uzayı koordinatlarında geliştirilen eklem uzayı kontrol yapısı, dięeri ise kartezyen koordinatlara dayanarak tasarlanan görev uzayı kontrol yapısıdır. Hareketli platformun verilen yörüngesine göre, ters kinematik sonucu hesaplanan referans bacak yörüngelerinin takibi kontrolünde, eklem uzayı kontrol yapısı kullanılır. Bu kontrol yapısında, 6 bacak uzunluęu uygun bir şekilde kontrol edilirse paralel manipölatörün hareketli platformu, istenilen yörüngeleri takip edebilir. Görev uzayı kontrol yapısı ise, platformun kartezyen koordinatlarında ifade edilen dinamięine dayalı olarak tasarlanır. Her iki kontrol stratejisinin avantajları ve dezavantajları vardır. Manipölatörün yörünge hataları doğrudan elde edilebildięinden görev uzayı kontrol yaklaşımı daha tercih edilebilir görünmektedir. Ancak hareketli platformun hız ve konumu için 6 serbestlik dereceli bir sensör bilgisine ya da platformun ileri kinematik bilgisine gereksinim duyulur. Paralel manipölatörlerde ileri kinematik, çözümü zor ve karmařık bir problemidir.

Yaygın olarak bilinen Stewart Platform'u son derece doğrusal olmayan bir sistem olduęu için seri manipölatörlerle karşılaştırıldıęında kontrol edilmesi daha zordur. 6 serbestlik dereceli Stewart Platform manipölatörün kontrolü için klasik PID kontrolörü pratikte uygulanmaktadır. Ancak, paralel bir manipölatörün arzu edilen yüksek performansını her zaman saęlayamayabilir. Bundan dolayı, yörünge takip kontrol performansını arttıracak ve model belirsizlikleri ve harici bozucuların üstesinden gelecek yüksek seviyeli doğrusal ya da doğrusal olmayan kontrol yapıları geliştirilir. Stewart Platform'un kontrolünde, literatürde PD ve PID kontrol, kayma kip kontrol, ters dinamik ya da hesaplamalı tork kontrol, yapay sinir aęlı kontrol ve bulanık mantık kontrol gibi kontrol yöntemleri kullanılmaktadır.

2. ROBOT KONTROL SİSTEMLERİ

2.1. Giriş

Robot kontrolörün tasarımında, kontrol sistemi için dinamik modelin belirlenmesi gerekir. Doğrusal olmayan dinamikler, eklemler arasındaki dinamik etkileşim, iç parametre değişimleri (sürtünmeler, kütle ve eklemlerdeki değişimler), dış bozucular, çok eksenli kontrol ve de sürekli değişen çalışma çevresinden dolayı özünde karmaşık bir kontrol problemi olabilir. Bunun yanında, bir sistem için elde edilen teorik model, kontrol altındaki sistemin gerçek çalışma çevresi ile tam olarak eşleşmediğinde problemler de artabilir.

Robot kontrol probleminde amaç, robotun uç noktasının verilen yörüngeyi en hassas şekilde izlemesini sağlamaktır. Bunun için uygun bir kontrolör tasarımı yapılmalıdır. Geniş bir uygulama alanı bulunan robotlar için yıllardır birçok kontrol metodu geliştirilmiştir. Bunlar doğrusal ve doğrusal olmayan iki kısımda incelenebilir. Sistemin dinamik modeli doğrusal diferansiyel denklemlerle ifade edilebiliyorsa doğrusal kontrol kullanılabilir. Robot dinamiği doğrusal olmayan diferansiyel denklemlerle ifade edilir. Bazı yaklaşımlarla robotun dinamik modeli, doğrusal diferansiyel denklemlerle elde edilebilmekte ve önemli sonuçlar alınabilmektedir. Bu yüzden endüstriyel ortamlarda sıklıkla doğrusal kontrol yöntemleri uygulanmaktadır.

Endüstride kullanılan robot kontrol sistemlerinin büyük çoğunluğu nispeten daha basit olan PD (Orantı + Türev) ve PID (Orantı + Integral + Türev) tipi kontrolörlerdir. Endüstriyel çevrelerde PID kontrol yöntemi hala en çok kullanılan geri beslemeli kontrolördür. Bu kontrol yönteminin olumlu ve olumsuz yönleri bulunmaktadır. Bu tip kontrol sistemlerinin sadece düşük hızlarda yeterli olabildikleri yapılan çalışmalar ile kanıtlanmıştır [6].

Az sayıdaki tasarım parametresinin olması ve bunların performans ölçüleri ile kolayca ilişkilendirilebilmesi ayrı bir avantaj teşkil etmektedir. Buna karşın değişken sistem parametrelerinin etkisini hesaba katamaz. Bu nedenle doğrusal olmayan sistemler için uygun ve istenilen sonuçlar elde edilemeyebilir. Sonuç olarak, klasik PID kontrolörler, karmaşık derecesi yüksek doğrusal olmayan sistem yapılarında uygun kontrol imkânı sağlayamamaktadır.

Klasik kontrol sistemlerini kullanan endüstriyel robotlarda belirli bir hız limitinin üzerine kolayca çıkılmamakta, sonuç olarak üretim verimliliği sınırlanmaktadır. Robot kontrol sistemlerinden beklenen ve her geçen gün artan performans isteklerinden dolayı daha gelişmiş kontrol tekniklerine gereksinim duyulmaktadır.

Fiziksel sistemlerin matematiksel modellerinin çıkarılması her zaman mümkün olmayabilir. Bazı sistemlerde modelleme doğru şekilde yapılsa bile elde edilen modelin kontrolör tasarımında kullanımı karmaşık problemlere neden olabilir. Bu gibi durumlarda genellikle bir uzman kişinin bilgi ve deneyimlerinden yararlanılma yoluna gidilir. Temeli, insanın bir sistemi kontrol etmedeki düşünce ve sezgilerine bağlı davranışın benzetimine dayanan kontrol yöntemi, bulanık mantık yaklaşımıdır. Bulanık mantık ilk olarak Zadeh tarafından 1960'lı yıllarda önerilmiş, birçok farklı bilim alanında kullanılan dilsel bir tekniktir [7]. Bulanık kontrolün geleneksel kontrol yöntemlerine göre temel avantajı dilsel tanımları kullanarak sistem dinamiği ve onun karmaşıklığının tasarıma yansıttığı güçlükleri rahatlıkla aşabilmesidir. Bir kontrolörün dinamik bir yapıya sahip olması, doğrusal olmayan sistemler için tam olarak bilinmeyen sistem dinamiklerinin ve zamanla değişebilen değişkenlerin kontrolünde çok etkili olmasını sağlar. Bulanık mantık kontrol sisteminde uygun üyelik fonksiyonlarının ve bulanık kuralların bulunmasıyla bu dinamik yapı elde edilebilir. Bu dinamik yapı ile yüksek performanslı bulanık kontrol sistemi endüstriyel alanlarda uygulanabilir.

Kesir dereceli sistemler, sistem derecesi herhangi bir reel sayı olan sistemlerdir. 19. yy'da ortaya atılmış olmasına rağmen kesir dereceli sistemlerin mühendislik alanında uygulamaları, 1990'lı yıllarda gerçekleştirilmiştir. Günümüzde kesirli hesaplamının daha iyi anlaşılması, bilim ve mühendisliğin birçok alanlarında kesir dereceli

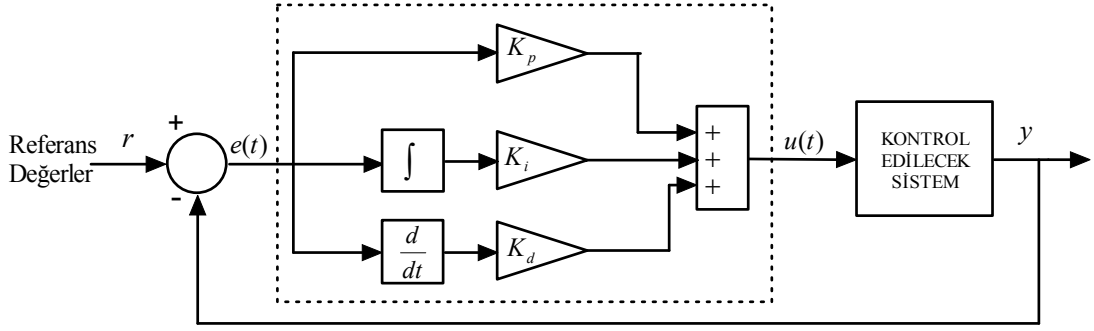
kontrolörlerin uygulanması ile ilgili çalışmaların artması kontrol sistemlerindeki problemlerin çözümü için farklı bir yöntem olarak ortaya çıkmasını sağlamıştır [8]. Bu yönetime göre elde edilen kontrolörlerin analizi, tasarımı, ayarlanması ve uygulanması ile alakalı yapılan çalışmaların önemli olduğu görülmektedir. Bunun yanında, kesir dereceli sistemlerin etkileri elektronik, otomatik kontrol, robotik ve haberleşme gibi birçok alanda incelenmekte ve bu alanlarda uygulamaları gerçekleştirilmektedir.

Kontrol sistemlerinde yaygın kullanılan kontrolörlerin de kesir dereceli yaklaşımdan etkilendiği görülmektedir. Endüstriyel ortamlarda geniş bir şekilde kullanılan PID kontrolörler, kesir dereceli integralleyici ve diferansiyelleyici düşüncesi kullanılarak geliştirilmiştir. Kesir dereceli PID kontrolör olarak tasarlanan kontrolöre ait parametre belirleme çalışmaları büyük ilgi görmüştür. Günümüze bakıldığında, kesir dereceli kontrolörler ve uygulamaları konusunda çok sayıda çalışmalar yapılmıştır.

Robot kontrol sistemlerinde kullanılan kontrolörlerin performanslarının karşılaştırılmasında önemli kriterler, istenen yörüngeyi izleme duyarlılığı, cevap hızı, bozucu dış yükler, kullanılan matematiksel algoritma ve modelleme hatası olarak söylenebilir. Bu çalışmada, Stewart Platform'u için tasarlanacak olan PID, Kesir Dereceli PID ve Bulanık Mantık kontrolörleri hakkında bilgi verilecektir.

2.2. PID Kontrolör

Endüstriyel ortamlarda sıklıkla kullanılan PID kontrolör, diğer kontrolör algoritmalarından daha fazla yaygın ve geniş kullanım alanına sahip bir yapıdır. Birçok geri besleme döngüsü, bu algoritma veya bu algoritmanın benzerleriyle kontrol edilirler. PID kontrolörler, otomatik kontrolün bel kemiğidir. Proses kontrol, motor sürücüleri, otomotiv sektörü, uçuş kontrol gibi endüstri ve kontrol mühendisliği çalışmalarının büyük çoğunluğunda bu kontrolör tipi kullanılmaktadır [9]. Genellikle, sistemde geri besleme varsa basit yapısı ve kullanılabilirliğinden dolayı çözüm için başvurulacak ilk kontrol yöntemi PID kontrolördür.



Şekil 2.1. PID kontrollü geri beslemeli sistem

Şekil 2.1’de gösterilen PID kontrolör yapısı, oransal (Proportional), integral (Integral) ve türev (Derivative) olan üç kontrolörün toplamından oluşmaktadır. Hata değeri (e), referans değeri (r) ile gerçekleşen çıkış değeri (y) arasındaki farkı gösterir. Bu (e) hata sinyali PID kontrolöre gönderilir ve kontrolör bu hata sinyalinin hem türevini hem de integralini hesaplar. Kontrol edilen sistemin girişine uygulanacak olan kontrol sinyali (u); K_p ile (e) hata değeri, K_i ile (e) hata değerinin integrali ve K_d ile (e) hata değerinin türevi çarpımlarının toplamına eşittir. Denklem (2.1)’de PID kontrolörün sürekli zaman formundaki ifadesi verilmektedir.

$$u(t) = K_p e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt} \quad (2.1)$$

Yukardaki denklemin Laplace dönüşümü alındıktan sonra PID kontrolörünün transfer fonksiyonu aşağıdaki gibi elde edilir.

$$U(s) = E(s) \left(K_p + K_i \frac{1}{s} + K_d s \right) \quad (2.2)$$

Bu (u) sinyali kontrol edilen sisteme gönderilir ve yeni çıkış (Y) elde edilmiş olur. Bu (Y) çıkış sinyali geri gönderilerek yeni hata sinyali (e) bulunur. Kontrolör, yeni hata sinyaline aynı işlemleri uygular ve bu işlem devam eder. Şekil 2.1’de gösterildiği gibi 3 temel kontrol etkisini birleştiren PID kontrolörde oransal etki (K_p) sistemin cevap hızını artırır, integral etki (K_i) sistemde ortaya çıkabilecek kalıcı durum hatasını sıfırlar ve türev etki (K_d) de sistemin cevabındaki titreşimleri bastırır, aşımı azaltır. Buna göre PID kontrolör, kontrol edilen sistemde sıfır sürekli rejim hatası olan, aşımsız hızlı bir cevap sağlar.

2.3. Kesir Dereceli PID Kontrolör

Çok önceden bilinen “Kesirli Matematik” konusu, son 10 yılda kullanımının artması ile kesir dereceli kontrol sistemleri de dâhil olmak üzere birçok alanda yapılan çalışmalara öncülük etmiştir. “Kesirli Matematik” aslında, ismi tam doğru olmayan bir ifadedir. Daha doğru bir tanımlama yapılırsa, “derecesi tamsayı olmayan türevleme ve integrasyon” şeklinde ifade edilebilir [10]. Dolayısıyla, kesir dereceli sistemler, türevlerinin dereceleri herhangi bir reel sayı olan diferansiyel denklemlerle ifade edilen dinamik sistemlerdir.

Kontrol sistemleri ile ilgili problemlerin çözümü için farklı bir yöntem olarak ortaya çıkan kesir dereceli dinamik sistemler, kontrol sistemlerinde de geniş kullanım alanı bulmaktadır. Kesir dereceli kontrol sistemleri, güç elektroniği, sistem tanımlama, robotik uygulama, sulama kanal kontrolü, mekatronik sistemler ve ısı transferi sistemleri gibi endüstriyel ve çeşitli kontrol uygulamalarında kullanılmaktadır [11-22].

Geniş bir yelpazede kullanılan kontrolörlerin de kesir dereceli sistemlerden etkilendiği görülmektedir. PID kontrolörlerin endüstriyel ortamlarda yaygın bir şekilde kullanımı, araştırmacıları daha iyi tasarım yöntemleri ya da alternatif kontrolörler bulma konusunda çalışmaya yönlendirmiştir. Buna örnek olarak, geniş bir yelpazede kullanılan PID kontrolörün kalitesini ve sağlamlığını arttırmak amacıyla, kesir dereceli integralleyici ve diferansiyelleyici düşüncesi kullanılarak PID kontrolör yeniden tasarlanmıştır. Tasarlanan kesir dereceli bir PID kontrolör, λ dereceli integratör ve μ dereceli türevi içeren bir PID kontrolörün genelleştirilmiş şekli olan $PI^\lambda D^\mu$ kontrolör olarak ortaya çıkmıştır.

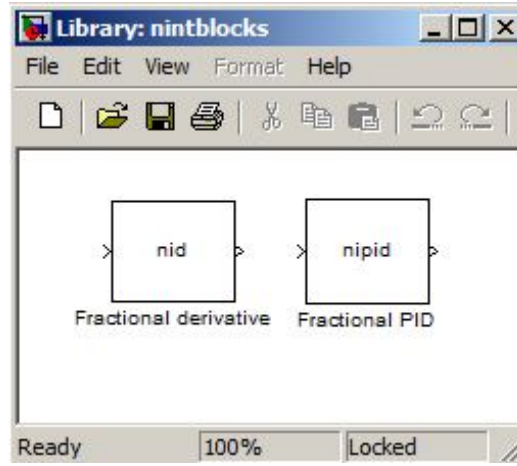
Kesir dereceli PID ($PI^\lambda D^\mu$) kontrolör fikri ilk olarak Podlubny [23] tarafından genelleştirilmiş PID kontrolör olarak sunulmuştur. Bu çalışmadan sonra kesir dereceli kontrolörler ve uygulamaları konusunda çok sayıda çalışmalar yapılmıştır. Bu çalışmalarda, $PI^\lambda D^\mu$ kontrolörün tasarımına, frekans bölgesi yaklaşımlarına, kesir dereceli türev ve integrallerin ayrıştırma şemalarına ve $PI^\lambda D^\mu$ kontrolörü kullanılarak yapılan frekans tabanlı çalışmalara yer verilmiştir [24-29].

$PI^\lambda D^\mu$ (FOPID) kontrolörü tasarlanırken hesaba katılması gereken birçok yön vardır. Tasarım sırasında FOPID kontrolörün içerdiği 5 parametrenin ($K_p, K_d, K_i, \lambda, \mu$) ayarlanması gerekir. Bu sebeple, FOPID kontrolörün, genellikle bozucu etkilere, yüksek frekans gürültüsüne ve sistem modelindeki belirsizliklere karşı sağlam olması istenir. Böylece, FOPID kontrolörün ayarlanma yöntemindeki sınırlamaların hepsi göz önünde bulundurularak K_p, K_d, K_i, λ ve μ parametrelerin en iyi değerleri bulunabilir. Verilen sınırlamalar altında FOPID kontrolörün en iyi parametrelerini bulmak için bir yöntemin geliştirilmesi amaçlanmalıdır. Ancak, geliştirilen yöntem güvenilir ve basit olmalıdır. Bu gereksinimleri sağlayacak iyi bir ayarlama yöntemi olarak esnek programlamayı (Soft Computing) oluşturan yapay sinir ağları, bulanık mantık ve evrimsel algoritmalar gözükmemektedir. Bu yöntemlerden evrimsel algoritmaları, doğal genetiği, yapay sinir ağları insanın sinir ve öğrenme yapısını, bulanık mantık ise insanın düşünme biçimini modelleyerek çözüm arar.

Doğrusal ve doğrusal olmayan sistemlerde FOPID kontrolörün parametrelerini optimize etmek oldukça zordur. Bu parametreleri otomatik olarak optimize edecek verimli ve etkili küresel bir yaklaşıma gereksinim duyulur. Bundan dolayı, evrimsel yaklaşımlara dayanan Genetik Algoritmalar (Genetic Algorithms - GA) [30-32], Diferansiyel Gelişim algoritması (Differential Evolution – DE) [33] ve Parçacık Sürü Optimizasyon algoritması (Particle Swarm Optimisation – PSO) [34-36] gibi son zamanlarda popüler olan popülasyon tabanlı en iyileme algoritmaları, FOPID kontrolörün parametrelerini optimize etmek için önerilmektedir.

Kesir dereceli kontrol sistemlerin bilgisayar ortamında analizini yapabilmek için bir takım programlara gereksinim duyulmuş ve bu konuda çalışmalar yapılmıştır. Literatüre bakıldığında, kesir dereceli kontrol sistemlerinin analizi için Oustaloup tarafından geliştirilen ve yaygın olarak kullanılan ilk uygulama, kesir dereceli sistemlerin dayanıklı kontrolü anlamına gelen CRONE (Commande Robuste d'Ordre Non Entier)'dur [37-40]. MATLAB ortamında kesir dereceli sistemlerin dayanıklı kontrolü için geliştirilmiş olan CRONE programı, Matematik Araçları modülü, Kesirli Model Tanımlama modülü ve CRONE CSD (Control System Design)

modülü olmak üzere 3 modülden oluşmuştur [37]. Her modül kesirli türevin özel bazı uygulamaları ile ilgilenir. Valerio [41], 2005 yılında “Toolbox ninteger for MATLAB v. 2.3” adlı, MATLAB ortamında çalışan bir program yayınlamıştır. Valerio’nun geliştirdiği NINTEGER programı, hem zaman bölgesinde, hem de frekans bölgesinde kesir dereceli, tek giriş tek çıkış kontrolörler tasarlamaya yarayan MATLAB ortamında yazılmış bir programdır. Bu program sayesinde, frekans ve ayrık zaman bölgelerinde kesir dereceli kontrolörler tasarlanabilir. Kontrolör tasarımındaki kesir dereceli türevler oluşturmak için 30’dan fazla formül ve yaklaşım mevcuttur. Ayrıca, kesir dereceli PID yapılarında Toolbox’ın içinde doğrudan vardır ve Şekil 2.2’de gösterilen bir simülink kütüphanesi içermektedir.



Şekil 2.2. NINTEGER programın simülink kütüphanesi

Bu tez aşamasında Stewart Platform’unun kontrolöründe kullanılan, parametreleri optimize edilmiş FOPID kontrolörü için yaklaşık transfer fonksiyonu, NINTEGER programında bulunan “nipid” fonksiyonu kullanılarak elde edilmiştir. Bu Toolbox’ların dışında farklı bir çalışma olarak Şenol [42], tez çalışmasında UFT-FOCS (User Friendly Toolbox for Fractional Order Control) olarak adlandırdığı bir program geliştirmiştir. Geliştirdiği bu program ile tamsayı dereceli ve kesir dereceli sistemlerin zaman bölgesinde birim basamak tepkisi, frekans bölgesinde de Bode, Nyquist ve Nichols grafikleri kolayca elde edilebilmektedir. Bu grafikler parametre belirsizliği içeren tamsayı dereceli ve kesir dereceli sistemler için de elde edilebilmektedir. Ayrıca parametre belirsizliği içeren sistemlerin Bode ve Nyquist zarflarını elde etmek için gerekli olan algoritmalar da programa dâhil edilmiştir.

2.3.1. Kesirli matematiğin geliřimi

Kesir dereceli sistem, kontrol edilmek istenen sistem ya da kontrolörün kesir dereceli diferansiyel denklemlerle ifade edilmesine denir. İlk olarak 1695 yılında L'Hospital, türevin derecesindeki $(\frac{d^n f(x)}{dx^n})$ n deęerinin kesirli olduęu durumda türeve nasıl bir anlam yükleneceęini fikrinden Leibniz'e bahsetmiř ve o zamandan beri kesirli matematik ilgi çekmeye bařlamıřtır. Daha sonraları bu konudaki sistematik çalıřmalar 19. yy bařlarında ve ortalarında Liouville (1832), Holmgren (1864) ve Riemann (1953) tarafından gerçekteřtirilmiřtir [42]. 1884 yılında geliřtirilen genelleřtirilmiř operatör teorisi genişletilerek rasyonel, irrasyonel, pozitif, negatif, reel veya kompleks olabildięi türev operatörlerini de kapsamıřtır [43-44]. Podlubny [44], çalıřmasında diferansiyel ve integralin notasyonlarının genelleřtirilmesi için birkaç yaklařım ele almıř, istenilen keyfi mertebeden kesirli türev ve kesirli integraller ve bunların özelliklerine yer vermiřtir. Ayrıca kesirli diferansiyel denklemler bu çalıřmada bilhassa ele alınmıřtır.

Kesirli türevler ve kesirli integraller doğrusal olmayan kesirli diferansiyel denklemlerin çözümlerinde kullanılmıřtır. Kesirli diferansiyel denklemlerinin çözümleri ayırma yöntemi üzerinden elde edilmiřtir. Genel olarak doğrusal olmayan kesirli diferansiyel denklemlerden tam çözüme ulařılabilir bir yöntem yoktur. Sadece yaklařık çözümler türetilmiřtir.

2.3.2. Kesirli matematiğin temel kavramları

Bir $F(t)$ fonksiyonun kesirli integral veya türevini doğrudan hesaplamak çok zor olduęundan bazı yaklařıklık tanımlarından faydalanılır. Bu konuda en çok kullanılan tanımlar Grünwald-Letnikov, Riemann-Liouville ve Caputo tanımlarıdır [42,44-45]. Tamsayı derecesine karřılık gelen en genel tanım, Denklem (2.3) ile verilen Riemann ve Liouville [44-45] yaklařımlarıdır.

$${}_a D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau, \quad n-1 < \alpha < n \quad (2.3)$$

$\Gamma(\cdot)$ Euler'in gama fonksiyonu, a ve t sırasıyla alt ve üst sınır, α , türev ve integralin derecesi (ondalık veya karmaşık sayı olabilir). ${}_a D_t^\alpha$ ise, sürekli zamanlı bir integro-diferansiyel (integrotürev) olarak adlandırılan operatördür.

Sistem teorisinde dinamik davranışların analizi için genellikle transfer fonksiyonları kullanılır. Kesir dereceli sistemlerin sürekli transfer fonksiyonları Laplace dönüşümü yardımıyla elde edilir. Laplace dönüşümünün kesir dereceli sistemler için en genel formül aşağıda verildiği gibidir [45]. Buna göre, tam sayı olamayan n ($n \in \mathfrak{R}_+$) için $x(t)$ sinyalinin n . dereceden türevinin Laplace dönüşümü:

$$L\{D^n x(t)\} = \int_0^\infty e^{-st} {}_0 D_t^n x(t) dt = s^n X(s) - \sum_{k=0}^{m-1} s^k {}_0 D_t^{n-k-1} x(t)|_{t=0}, m-1 < n < m \quad (2.4)$$

Denklemden kullanılan m tamsayıdır. Hem simülasyonlarda hem de donanımsal uygulamalarda kesir dereceli s 'e sahip fonksiyonu kullanmak pek uygun değildir. Bu yüzden bu fonksiyonu tamsayı dereceli bir transfer fonksiyonuna benzetmek gerekmektedir. Kesir dereceli transfer fonksiyonunun, yaklaşık olarak tamsayı dereceli transfer fonksiyonu şeklinde hesaplanabilmesi için, sonsuz sayıda kutup ve sıfırlar içermelidir. Yine de, sonlu sayıda sıfır ve kutuplar içeren kabul edilebilir yaklaşımlar da elde etmek mümkündür. Yaklaşım teknikleri vasıtasıyla kesir dereceli transfer fonksiyonları, tamsayı dereceli transfer fonksiyonlarına dönüştürülebilir.

En iyi bilinen tekniklerden birisi Oustaloup tarafından önerilen Crone yaklaşımıdır [37]. Bu yaklaşım, N kutuplar ve N sıfırların yinelenmeli bir dağılımını kullanır. Yaklaşık transfer fonksiyonu aşağıda ifade edildiği gibi elde edilir [37].

$$s^v \approx k \prod_{n=1}^N \frac{1 + \left(\frac{s}{\omega_{z,n}} \right)}{1 + \left(\frac{s}{\omega_{p,n}} \right)} \quad (2.5)$$

Yaklaşım hesabı, $[\omega_l; \omega_h]$ frekans aralığında geçerlidir. Denklemdenki eşitliğin her iki tarafı, 1 rad/s birim kazancıma sahip olabilmesi amacıyla k kazancı ayarlanır. Kutup ve sıfırların sayısını gösteren N , önceden seçilir. Sonuçta yaklaşık olarak elde

edilen transfer fonksiyonunun iyi bir performansa sahip olması şunlara bağlıdır: düşük değerler daha basit yaklaşımları üretir, ama hem kazanç hem de faz şekillerinde dalgacıklar (ripples) oluşabilir. N sayısını arttırarak bu dalgacıklar yok edilebilir. Bu defa da, N arttığından dolayı yaklaşım süreci hesapsal olarak ağırlaşacaktır.

Denklem (2.5)'te verilen kutup ve sıfırların frekansları aşağıda verilen ifadeler ile hesaplanır.

$$\omega_{z,1} = \omega_l \sqrt{\eta} \quad (2.6)$$

$$\omega_{p,n} = \omega_{z,n} \varepsilon, \quad n = 1, \dots, N \quad (2.7)$$

$$\omega_{z,n+1} = \omega_{p,n} \eta, \quad n = 1, \dots, N-1 \quad (2.8)$$

$$\varepsilon = \left(\frac{\omega_h}{\omega_l} \right)^{v/N} \quad (2.9)$$

$$\eta = \left(\frac{\omega_h}{\omega_l} \right)^{(1-v)/N} \quad (2.10)$$

Denklem (2.5)'in tersini alarak $v < 0$ durumu göz önünde bulundurulabilir. Eğer $|v| > 1$ ise, yaklaşım yetersizleşir ve genel olarak bu, kesir dereceli s^v 'in aşağıda ifade edildiği gibi bölünmesine neden olur.

$$s^v = s^n s^\delta, \quad n \in Z, \quad \delta \in [0,1] \quad (2.11)$$

Denklemdaki son terim olan s^δ 'nin yaklaşımı hesaplanır. Yaklaşım işlemi, aşağıda belirtilen adımlarla yapılabilir.

- $[\omega_l; \omega_h]$ frekans aralığı ve N , uygun bir şekilde seçilir.
- Denklem (2.7) ve (2.8)'den $\omega_{p,n}$ ve $\omega_{z,n+1}$ parametreleri hesaplanır.
- Denklem (2.5) ile de tamsayı dereceli transfer fonksiyonu elde edilir.

2.3.3. Kesirli matematiğin PID kontrol sistemine uygulanması

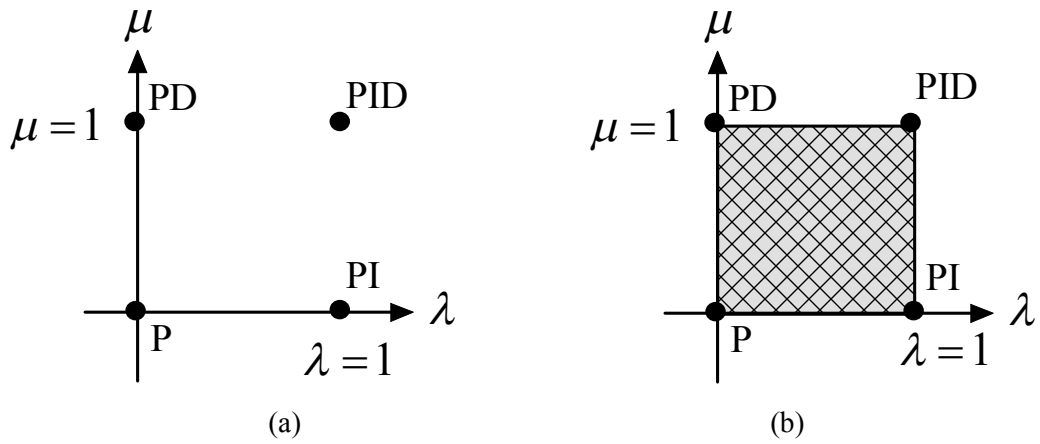
Yapılan çalışmalarda sunulan kesir dereceli PID kontrolör genelleştirilmiş olarak reel dereceli integralleyici ve reel dereceli türevleyici halinde $PI^\lambda D^\mu$ olarak verilmiştir. $PI^\lambda D^\mu$ kontrolörün diferansiyel denklemini aşağıda verildiği gibi ifade edilir.

$$u(t) = k_p e(t) + k_i D_i^{-\lambda} e(t) + k_d D_i^\mu e(t) \quad (2.12)$$

Laplace dönüşümü yardımı ile FOPID'nin sürekli transfer fonksiyonu aşağıdaki gibi elde edilir.

$$G_c(s) = K_p + K_I s^{-\lambda} + K_D s^\mu \quad (2.13)$$

Verilen bir sistem için FOPID kontrolörün tasarımında K_p, K_d, K_i ve ondalık dereceli λ, μ olmak üzere toplam 5 parametre en iyi olarak hesaplanır. Şekil 2.3'te gösterildiği gibi bu tip kontrolör klasik PID kontrolörünü genelleştirir ve noktasaldan yüzeyselere doğru açar. Detaylı bir şekilde ifade edilirse; Şekil 2.3.(a)'da $\lambda=1$ ve $\mu=1$ için tam dereceli PID kontrolörün λ - μ düzlemindeki gösterimi, (b)'de ise kesirli halinin düzlem üzerindeki gösterimi verilmiştir. Tam dereceli kontrolör düzlem üzerinde 4 nokta ile ifade edilirken, kesir dereceli kontrolör taralı alandaki sonsuz nokta ile ifade edilebilir. Bu durum, kontrol edilecek sistemin dinamiği üzerine daha esnek bir müdahalenin söz konusu olduğu anlamına gelmektedir.



Şekil 2.3. Klasik PID (a), Kesir dereceli PID'nin P-I-D düzlemi (b)

2.4. Bulanık Sistemler

Fiziksel sistemlerin matematiksel modellerinin çıkarılması her zaman mümkün olmayabilir. Bu durumlarda fiziksel sistemin yapısı ve kontrolü hakkında bilgiye sahip kişilerin tecrübesine başvurmak gerekir. Bu ise Bulanık Mantık sistemi ile mümkün olabilir. Bulanık mantık sisteminde, uzman kişilerden, dilsel ifadeler olarak alınan bilgiler, bulanık mantık kurallarıyla ifade edilir. Ayrıca, matematiksel modeli tam olarak bilinmeyen ve doğrusal olmayan sistemlere uygulanabilir. Bulanık mantığın amacı, insanların tam ve kesin olmayan bilgiler ışığında, tutarlı ve doğru kararlar vermelerini sağlayan düşünme ve karar verme mekanizmalarının modellenmesidir.

Bulanık mantık kavramı ilk kez 1965 yılında California Berkeley Üniversitesi'nden Lotfi A.Zadeh'in bu konu üzerinde ilk makalelerini yayınlamasıyla duyuldu [46]. Yayımlanan "bulanık kümeler" isimli makalesinde, bir sistemdeki karmaşıklığın yarattığı belirsizliğin farklı görünüşlerini ve kişilerin algılama farklılıklarını ele almıştır. Zadeh'e göre, bir sistemdeki karmaşıklık arttıkça, sistemi betimleyen ifadelerin anlamı azalmakta ve anlamlı ifadeler de belirsizliğe doğru gitmektedir. Zadeh, bulanık küme kavramını ortaya attıktan sonra 1968'de bulanık algoritmalar, 1970'te ise bulanık karar verme yöntemlerini açıklayan bir makale yayımlamıştır. 1973'te Zadeh, "Kompleks sistemlerin ve karar verme işlemlerinin analizine yeni bir yaklaşım" adıyla bulanık kontrolün temelini atan bir makale yayımladı. Zadeh bu makalesinde dilsel değişkenleri ve insan bilgisini formüle etmek için "EGER-O HALDE (IF – THEN)" kurallarını kullanmayı önerdi [47].

Bir kavramı, bir amacı ve bir sistemi tanımlayan ifadelerdeki belirsizliğe veya kesin olmama durumuna bulanıklık denir. Bulanık Mantık insanların her gün kullandığı ve davranışlarının yorumlandığı yapıya ulaşılmasını sağlayan matematiksel bir disiplindir. Dünyadaki bazı olayları açıklamak için kesin tanımlamalar yapabilmek imkânsızdır ve olayların çoğu belirsizlikler ve doğrusal olmama özellikleri taşır. Terimler ya da ölçüler kesin olarak tanımlanıp ölçülemediğinden insanlar günlük hayatta genelde, dilsel niteleyiciler (soğuk, hafif soğuk, ılık, sıcak, çok sıcak vb) gibi belirsiz (kesin olmayan) ifadeler kullanırlar. Bulanık mantık sistemi de bazı sorulara

basitçe evet-hayır cevabı verilemeyen durumları kapsar. Bulanıklığın ve bulanık mantığın temeli de budur.

1965 yılında ortaya çıkan bulanık mantık kavramı, modele dayalı kontrol sistemlerine iyi bir alternatif sunmuştur. İlk olarak 1975 yılında Mamdani tarafından buhar makinesinin kontrolü amacıyla kullanılan bulanık mantık, makinelere insanların özel verilerini işleyebilme ve onların deneyimlerinden, önsezilerinden yararlanarak çalışabilme yeteneği verilmiştir [48]. Bundan bir yıl sonra Mac Vicar-Whelan ise kontrolör tasarımında PD (orantı-türev) ve PI (orantı-integral) yapısına benzeyen standart kurallar kullanılmasını önermiştir [49]. Bulanık kontrol 1980'lerden itibaren endüstride, otomotiv, metro sistemi, nükleer reaktör, asansör ve vinç kontrolü, ev aletleri gibi konularda sıkça uygulama alanı bulmuştur [50-51].

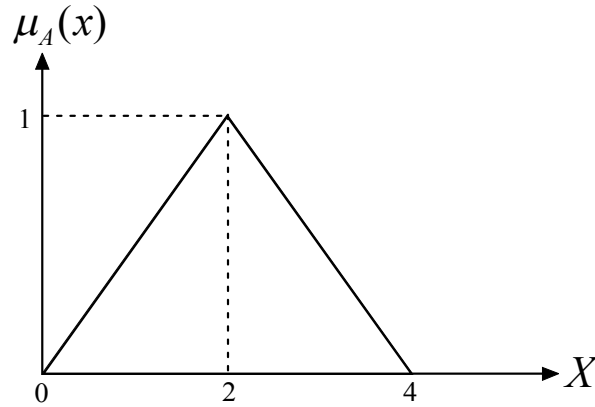
Gelişmiş kontrol sistemlerinde bulanık mantığın başarı ile kullanılması, bu sistemlerin insanların kesin olmayan bilgiler ışığında doğru karar vermelerini sağlayan karar verme mekanizmalarını modelleyebilmesine dayanır. Böylece bulanık mantık kontrol, sistemin kesin matematik modeline ihtiyaç duyan klasik kontrolün yerini alır. Klasik kontrolde gerçek sistem davranışları kesin matematiksel ifadelerle modellenmeye çalışılırken, bulanık mantık kontrolünde insan bilgisi, deneyimi ve mantığı ön plandadır [52].

2.4.1. Bulanık kümeler

Klasik küme teorisinde, kesin sınırlamalar ile verilen bir elemanın tanımı, ya kümeye aittir ya da ait değildir şeklinde yapılır. Ancak doğal diller, tam bir matematik tanımı olmaksızın “kısa, çok sıcak” gibi bir hayli belirsizliğe sahiptirler. Kesin bir küme tam olarak böyle seneryoları barındıramayabilir. Bulanık küme teorisinde, 0 ile 1 arasındaki değerlerde tanımlanan bir üyelik fonksiyonu bu problemi çözebilir. Biraz daha genişletirsek, kesin sınırlamaları olan bir küme, bulanık bir kümenin alt kümesi olarak düşünülebilir. Kesin bir küme ve bulanık bir kümenin matematiksel tanımları aşağıda verilmektedir.

Tanım 1: X bir nesnelere uzayı, x de bu uzaya ait bir eleman olsun. A kümesi de bu X 'in alt kümesi olan bir klasik küme olsun. Bu durumda her bir x için bu A kümesine aittir ya da ait değildir. Her bir x elemanı için bir karakteristik fonksiyon ($I_A(x) \in \{0,1\}$) tanımlayarak, klasik A kümesini $(x,0)$ veya $(x,1)$ sıralı ikililerle temsil edebilir. $A = \{(x, I_A(x)) \mid x \in X\}$. Bu gösterim x elemanının A kümesine ait olup olmadığını gösteren bir gösterimdir.

Tanım 2: X nesnelere uzayı, x 'de bu uzaya ait bir eleman olsun. X uzayı altında tanımlanan bulanık A kümesi, $A = \{(x, \mu_A(x)) \mid x \in X\}$ sıralı ikililerden oluşur. $\mu_A(x)$, bulanık A kümesi için üyelik fonksiyonudur (Membership Function). X 'in her bir elemanını 0 ile 1 arasında bir üyelik derecesine eşlemek için üyelik fonksiyonu belli bir aralığa getirilir.



Şekil 2.4. $\mu_A(x)$ üyelik fonksiyonu

Yukarıdaki tanımlamalara örnek olarak Şekil 2.4'te gösterilen üyelik fonksiyonu ele alınırsa, kümenin elemanı olma veya olmama hali belirli üyelik dereceleri ile ifade edilmektedir. Buna göre X nesnelere uzayı kümesi tamsayılar kümesi olmak üzere “2’ye yakın sayılar” bulanık kümesi $A = \{0/0, 0.5/1, 1/2, 0.5/3, 0/4\}$ şeklinde liste biçiminde gösterilebilir. Bu gösterimde paydadaki sayılar X nesnelere uzayı kümesine ait elemanları, paydaki sayılar ise elemanların üyelik derecesini göstermektedir. Liste gösteriminden anlaşıldığı gibi klasik küme kavramında kümenin elemanı olarak kabul edilmeyecek sayılar belirli bir üyelik derecesi ile A bulanık kümesine dahil olmaktadır. Bir üyelik fonksiyonu, teorik olarak birçok şekilde olabilir. Örneğin; üçgen, yamuk, Gauss ve sigmoidal şeklinde üyelik fonksiyonları vardır. Genel olarak,

bir üyelik fonksiyonu başlangıçta, buluşsal bilgi (heuristic knowledge) ve uzman deneyim (expert experience) vasıtalarıyla tasarlanır. Daha sonrasında ayarlama taktiğine dayalı olarak bu üyelik fonksiyonu ayarlanır.

2.4.2. Bulanık küme işlemleri

Klasik bir küme, bulanık bir kümenin alt kümesi olarak düşünüldüğü için klasik küme üzerinde yapılan ana işlemler bulanık bir kümeye göre de yapılabilir. Bulanık bir küme üzerinde tanımlanan işlemler, klasik küme üzerinde tanımlanan işlemlerle hemen hemen aynıdır.

A ve B bulanık iki küme olsun. Genel olarak, birleşme, kesişme ve tümleme işlemleri kullanılır. A veya B kümesine ait olan nesnelere uzayına A ve B kümesinin birleşimi denir. Hem A ve hem de B kümesine ait olan nesnelere uzayına A ve B kümesinin kesişimi denir. A kümesine ait olmayan nesnelere uzayına A kümesinin tümlenini denir. Bulanık kümeler ile ilgili diğer işlemler [53]'te detaylı olarak verilmektedir. Bu üç işlem aşağıdaki gibi tanımlanır.

Tanım 1: Birleşme: Her $x \in X$ için üyelik fonksiyonu $\mu_{A \cup B}$ aşağıdaki gibi tanımlanır.

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \quad (2.14)$$

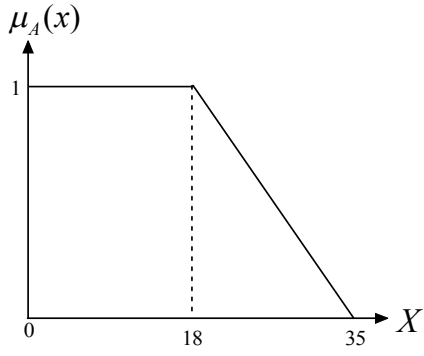
Tanım 2: Kesişme: Her $x \in X$ için üyelik fonksiyonu $\mu_{A \cap B}$ aşağıdaki gibi tanımlanır.

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad (2.15)$$

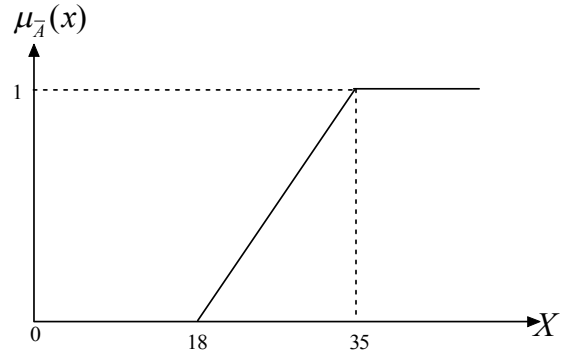
Tanım 3: Tümlenisi: Her $x \in X$ için üyelik fonksiyonu $\mu_{\bar{A}}(x)$ aşağıdaki gibi tanımlanır.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.16)$$

Yukarıda verilen tümlenme, birleşim ve kesişim işlemlerinin örnekleri aşağıda gösterilmektedir. Şekil 2.5, 2.6 ve 2.7 sırasıyla "genç", "orta yaşlı" ve "yaşlı" bulanık kümelerini göstermektedir.

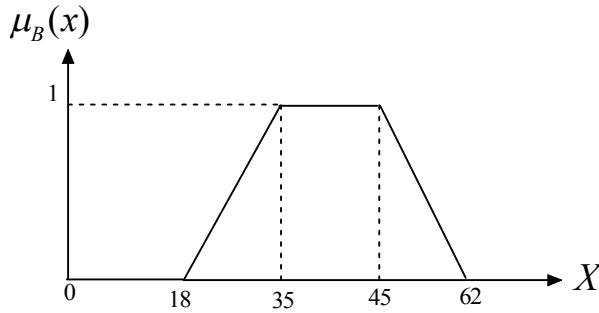


(a)

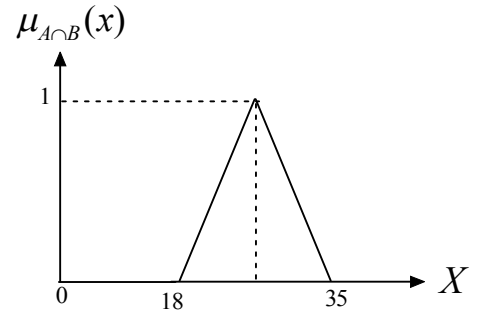


(b)

Şekil 2.5. “Genç” bulanık kümesi (a), “Genç” bulanık kümesinin tümleyeni (b)

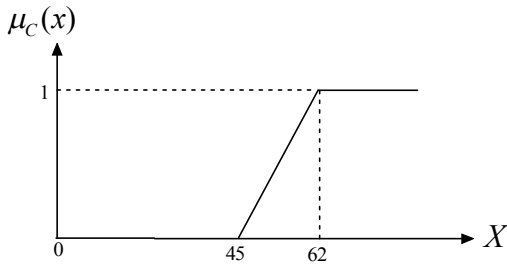


(a)

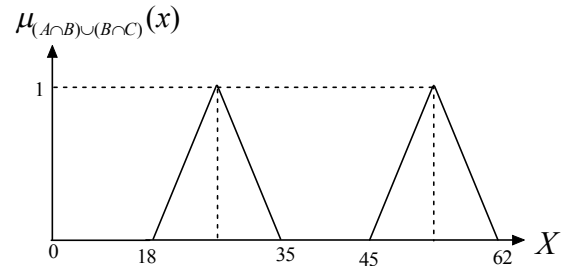


(b)

Şekil 2.6. “Orta Yaşlı” bulanık kümesi (a), “Genç” ve “Orta Yaşlı” bulanık kümelerinin kesişimi (b)



(a)



(b)

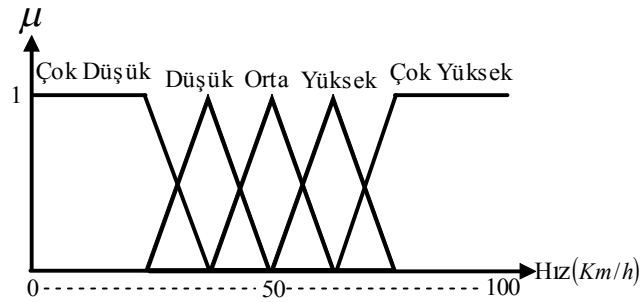
Şekil 2.7. “Yaşlı” bulanık kümesi (a), “Genç” ve “Orta Yaşlı” bulanık kümelerin kesişimi ile “Orta Yaşlı” ve “Yaşlı” bulanık kümelerinin kesişiminin birleşimi (b)

Şekil 2.5'te A kümesi ve tümleyeni, Şekil 2.6'da B kümesi ve A kümesi ile kesişimi, Şekil 2.7'de ise C kümesi, A kümesi ve B kümesinin kesişimi ile B kümesi ve C kümesinin kesişiminin birleşimi örnek olarak verilmektedir.

2.4.3. Kavramsal deęişkenler ve ifadeler

Kavramsal bir deęişken nitelik bakımından, hem bulanık bir kümenin ismi olarak görev yapan dilsel terimli hem de bulanık kümenin anlamını ifade eden üyelik fonksiyonlu bir bulanık kümeyi tanımlar [54]. Bu kavramlar bulanık kontrolde çok önemli anlam kazandırmıştır.

Kavramsal deęişkenler genelde birçok kavramsal ifadelerle açıklanırlar. Bu ifadelerin bulanık kümesi, deęişkenin fiziksel deęer aralığını kaplar. Şekil 2.8’de bakıldığında, sembolik olarak deęişken, hız adlandırılmıştır. Deęişkenin fiziksel deęer aralığı, yani deęişkenin keskin sayısal deęer aralığı [0, 100 km/h] seçilmiştir. Deęişkenin kavramsal deęerleri, yani deęişkenin keskin olmayan dilsel deęer aralığı, çok düşük, düşük, orta, yüksek ve çok yüksek seçilmiştir. Bulanık küme ya da üyelik fonksiyon, kavramsal deęerlerle modellenir. Üyelik fonksiyonun tespit edilmesiyle, dil kümesinden başka bir şey olmayan kavramsal deęerler fiziksel anlam kazanır. Buna uygun olarak ilgili kavramsal deęerin üyelik fonksiyonu, $\mu_{\text{çok düşük}}$, $\mu_{\text{düşük}}$ gibi adlandırılır. Burada deęer aralığının iç kısmı için üçgen şeklinde üyelik fonksiyonu ve deęer aralığının uç kısımları (kenar) için trapez şeklinde üyelik fonksiyonu seçilmiştir.

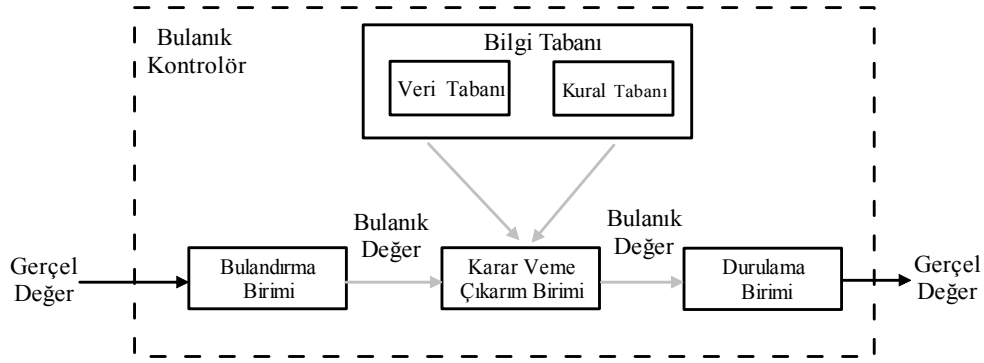


Şekil 2.8. Kavramsal deęişken örneęi

2.5. Bulanık Kontrol Sistemi

Bulanık mantığın en fazla uygulama bulduęu alan kontrol sistemleri olarak görülmektedir. Bunun en büyük nedeni de 1970’li yılların başında Mamdani ve arkadaşlarının bulanık mantık tabanlı bir kontrolör tasarlayarak uygulamaya koymaları olmuştur [48]. Bu uygulama daha sonra oldukça sık başvuru alan bir kaynak

olarak kullanılmış ve bulanık kontrolün diğer bulanık mantık uygulamalarına göre daha hızlı uygulamaya konulmasına neden olmuştur. Özellikle 1990'lı yıllarda bulanık mantık uygulamaları bir patlama kaydetmiş ve artık çalışmalar bulanık mantık tabanlı kontrolör tasarlamaktan çıkıp, bulanık mantık kontrolörlerin performanslarını arttırmaya yönelmiştir. Bulanık mantık kontrolü, kontrol edilen sistemin durum bilgisi ve kontrol işlemi arasındaki bulanık işlem algoritmasına dayanır. Bir sistem için matematiksel modelden ziyade, sistem bilgisinin daha önemli olduğu durumlarda bulanık kontrole başvurulur. Bundan dolayı, bulanık kontrolü geleneksel kontrol algoritmalarından ayıran en belirgin özelliğin sistem bilgisi olduğu söylenebilir. Bir bulanık mantık kontrolör sisteminin temel parçaları Şekil 2.9'da gösterilmektedir.



Şekil 2.9. Bulanık mantık kontrolörün yapısı

Bu sistemde, bulanıklaştırma (fuzzification) arabirimi, bilgi tabanı (knowledge base), karar verme bloğu (decision making logic) ve durulama (defuzzification) arabirimi olarak 4 temel kısım vardır.

1. Kavramsal değişkenlerle giriş ve çıkışın bulanıklaştırılması (Fuzzification)
2. Kural tabanın oluşturulması (Knowledge Base and Rule Base)
3. Bulanık çıkarımsal motor tasarımı (Decision Making Logic)
4. Bulanık çıkışın durulaştırılması (Defuzzification)

Bulanıklaştırma, söylev evrenindeki bir elemanı bulanık bir kümenin üyelik değerine dönüştüren bir matematiksel işlemdir [55]. Sayısal girişi, bulanık kontrol sistemi tarafından kullanılabilen bir üyelik fonksiyonlu bulanık bir kümeye dönüştürecek bir yol sağlar. Örneğin, sayısal giriş olarak 5 tamsayı değerini alalım. Bulanıklaştırma, bu

5 tamsayı deęerini kavramsal deęiřkene gre bulanık bir kmeye dnřtrr. İki trl bulanıklařtırma yapılabilir. Bunlar, bulanık sisteme giren bir sayısal deęeri tekli bulanık kmeye cvirme (Tekli - Singleton) ya da sayısal deęeri cgen, Gauss, cn gibi bulanık kmelere cvirmediir.

Bulanık kurallar ve bilgi tabanı, bulanık mantık kontrolrnn ckerdeęidir ve bulanık verinin iřlenmesi ve dilsel kontrol kurallarının tanımlanması iin kullanılan gerekli tanımlamaları ierir. Kural tabanı, bir dizi dilsel kontrol kuralları vasıtasıyla uzman tarafından belirlenen kontrol ilkelerini ve kontrol amalarını nitelendirir. Genel olarak, bulanık kontrolrn dinamik davranıřı, bir grup bulanık kurallarlarıyla nitelendirilir. Bulanık bir kural ařaęıda verilen formdadır.

IF antecedents Then consequents; If x is A then y is B (Eęer x A ise o halde y B'dir).

A ve B sırasıyla X ve Y uzaylarındaki bulanık kmeler tarafından tanımlanan dilsel deęerlerdir (kavramsal deęiřkenler). x ve y ise sırasıyla bu X ve Y uzayına ait elemanlardır. Genelde "x is A" kısmı gemiř (antecedent) veya řart (premise) olarak adlandırılır. "y is B" kısmı ise sonu (consequence) olarak adlandırılır. Daha da geniřletilirse; antecedents, birleřme (OR) veya keřiřme (AND) iřlemleri ile birleřtirilebilir. rneęin, A1 ve B1 giriř deęiřkenleri zerinde tanımlı iki bulanık kme; C1, cıkıř deęiřkeni zerinde tanımlı bir bulanık kme ve x, y ve z ise bulanık kontrolrn giriř ve cıkıřın dilsel deęiřkenleri olsun. O halde iki giriřli bir cıkıřlı bulanık kontrolr tanımlamıř olunur. Basit bir kural oluřturulmak istendięinde;

If x is A1 and y is B1 Then z is C1.

ıkarım mekanizması ve karar verme mantıęı, bulanık mantık kontrolrnn en temel grevleri yerine getiren kısımdır. Bu kısım, bulanık mantıktaki giriř kuralları ve bulanık cıkıřımı calıřtıran bulanık kontrol eylemlerinin cıkıřıldıęı ve de insan dřncesine gre karar verme yeridir. Bulanık cıkıřım mekanizması, bulanık bir kmenin giriřini bulanık cıkıřa aktarmaya yarayan bir yoldur. Her bir kuralın mevcut durumlarıyla alakalı olduęu lcy belirler. Ayruca, kural tabanındaki bilgi ve mevcut giriřleri kullanarak cıkıřım yapar. Genelde, bu cıkıřım mekanizması, bir dizi mantık

işlemlerinden oluşur. Bulanık çıkarımı uygulamanın birkaç yolu vardır. Bunlardan en çok kullanılanı Mamdani bulanık çıkarım sistemi ve Sugeno bulanık çıkarım sistemidir.

Durulaştırma, çıkış bulanık kümesinden keskin değerleri elde etme yolu olarak adlandırılır. Kısacası; durulayıcı, çıkarım mekanizmasının ürettiği bulanık kümeleri gerçel değerli sistem çıkışına dönüştürür. Durulayıcı seçimi yapılırken aşağıdaki hususlar göz önünde bulundurulmalıdır.

- Durulayıcıdan elde edilen keskin çıkış değeri bulanık kümeyi sezgisel ve mantıksal olarak iyi ifade etmeli.
- Kısa sürede hesaplanabilen basit bir algoritma seçilmeli.
- Bulanık kümedeki küçük bir değişiklik, elde edilen keskin değer çıkışta çok büyük değişikliklere neden olmamalı.

Durulandırmada en sık kullanılan stratejiler; Ağırlık merkezi yöntemi (Centroid of area), Alan açıortayı yöntemi (Bisector of area), Maksimum ortalama yöntemi (Mean of maximum), Maksimum en küçüğü yöntemi (Smallest of maximum) ve Maksimum en büyüğü yöntemi (Largest of maximum).

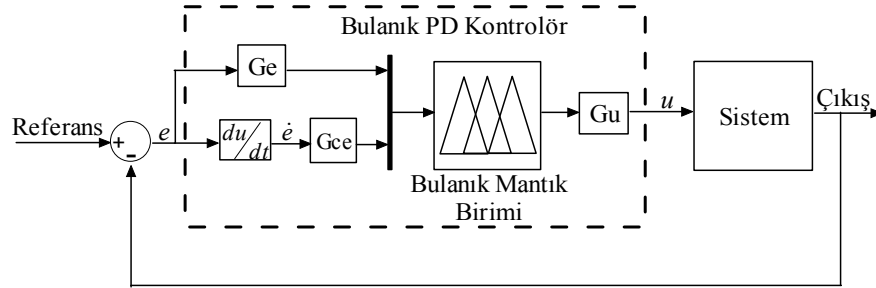
2.6. Bulanık Kontrol Çeşitleri

Özellikle doğrusal olmayan sistem modelleme ve kontrolünde yaşanan zorlukları aşmada bulanık mantık başarılı olmuştur. Bu sebeple kontrol mühendisliği sahasında da çok çeşitli kullanım alanları bulunmaktadır. Bulanık kontrolörler kontrol edilecek sisteme göre çok farklı girişlere ve sistemi kontrol etmekte kullanılan çok farklı çıkışlara sahip olabilirler. Bulanık kontrolörler, klasik kontrolörlerden esinlenerek bulanık PD, bulanık PI, bulanık PID ve karma kontrolörler şeklinde kullanılmaktadır.

2.6.1. Bulanık PD kontrolör

Şekil 2.10'da gösterilen PD tip bulanık kontrolör yapısında, sistemden gelen (e) hata ve (\dot{e}) hatanın türevi giriş bilgisi ve bu girişlere göre üretilecek olan uygun kontrol işareti de çıkış bilgisi olarak verilir. Yapıda yer alan G_e , $G_{\dot{e}}$ ve G_u sırasıyla hata değerinin

kazanç katsayısı, hata değişim değerinin kazanç katsayısı ve çıkış katsayısıdır. Bunlar giriş ve çıkış ölçekleme çarpanları olarak da adlandırılırlar. Sistem yanıtında iyi bir sönüm sağlanması ve maksimum aşım, yükselme zamanı ve yerleşme zamanı değerlerini azaltması, Bulanık PD kontrolörün önemli özelliğidir. Ayrıca PD kontrolör geri beslemeli sistemlerin kararlılığını artırır [56].

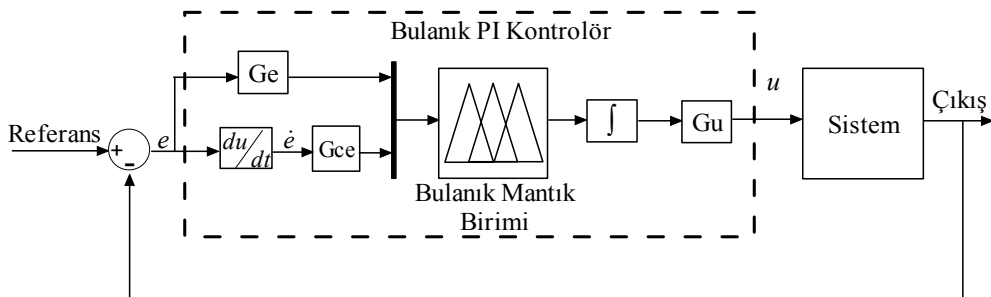


Şekil 2.10. Bulanık PD kontrolör

Sistemin cevabında sürekli hal hatası var ise Bulanık PD kontrolör, bu sürekli hal hatasını ortadan kaldıramayacaktır. Bunu gidermek için Bulanık PD kontrolöre, integral kontrolü eklenerek PI kontrolör yapısı kullanılır.

2.6.2. Bulanık PI kontrolör

Şekil 2.11’de, Bulanık PI yapısı gösterilmektedir. Bu yapıda, Bulanık PD kontrolörde olduğu gibi sistemden gelen (e) hata ve (\dot{e}) hatanın türevi girdi olarak verilerek bir çıkış işareti üretilir. Kontrolörün çıkışında integral alma işlemi uygulanarak sistem için kontrol işareti elde edilir. Sonuçta, Bulanık PI kontrolör giriş olarak hataya bağlı bu iki işareti kullanarak kontrol işaretindeki değişim miktarını belirler [57-58].



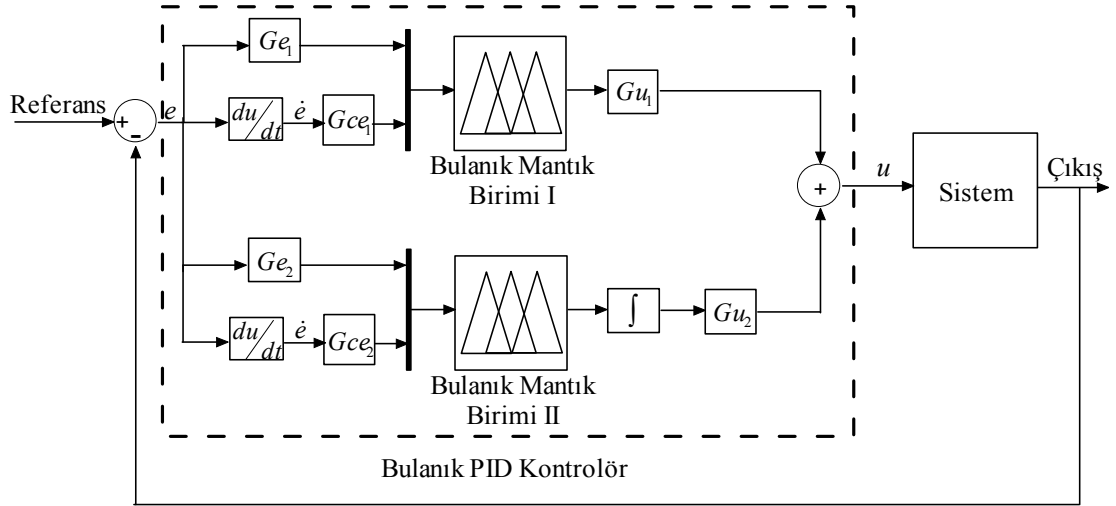
Şekil 2.11. Bulanık PI kontrolör

Bulanık PD kontrolör sürekli hal hatasını ortadan kaldıramadığı için Bulanık PI kontrolör daha etkilidir. Bunun yanında integral etkisinden dolayı, Bulanık PI kontrolör yüksek dereceli doğrusal olmayan sistemlerde salınımın neden olabilmekte ve kararlı hale gelmesi uzun zaman alabilmektedir.

PI tipi bulanık kontrolörler tasarlanırken girişte hatanın integrali alınarak da oluşturulabilir. Ancak hatanın integral bilgisinin kullanılması kuralların oluşturulmasında zorluk yarattığından, Bulanık PD kontrolör yapısına ek olarak kontrolörün çıkışına bir integratör konulması tercih edilir. Bunun dışında, kontrolörden elde edilen performansı daha da artırabilmek için PID kontrolör yapısı şeklinde Bulanık PD ve Bulanık PI kontrolörlere göre daha güçlü bir yapı tasarlanabilir. Böylece, kontrolör girişinde yalnızca (e) hata işaretini kullanarak tek girişli, (e) ile birlikte (\dot{e}) hatanın türevi kullanılarak iki girişli ve (e), (\dot{e}) ile beraber ($\int e$) işareti de kullanılarak üç girişli olarak Bulanık PID kontrolörler oluşturulabilir [59]. Ancak üç girişli Bulanık PID kontrolörde ($\int e$) bilgisiyle kontrol kuralları oluşturmak zor olduğundan hatanın integrali yerine (\ddot{e}), hatanın ikinci türevi bilgisi de kullanılabilir [59-60]. Ayrıca, üç girişli Bulanık PID tipi kontrolörlerde üç boyutlu kuralların oluşturulması zor olduğundan genelde iki girişli Bulanık PID yapıları, uygulamalarda tercih edilmektedir.

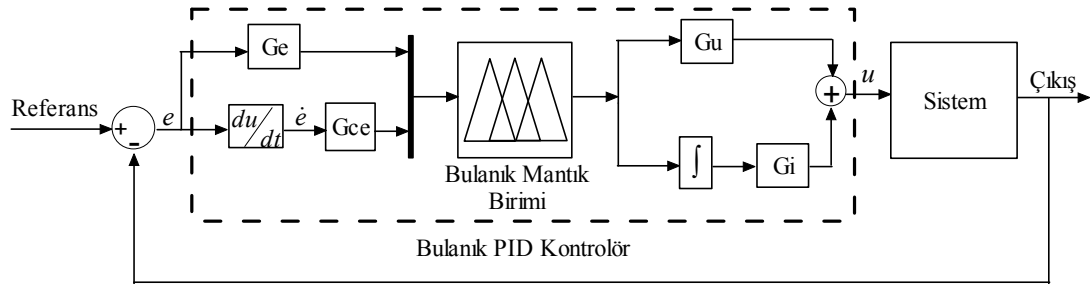
2.6.3. Bulanık PID kontrolör

Burada, Bulanık PID kontrolöründe sıkça kullanılan iki yapı verilecektir. Birincisi, Şekil 2.12'de gösterildiği gibi Bulanık PD ve Bulanık PI kontrolörlerin birleştirilmesi sonucu elde edilen kontrolör yapısıdır. Bu yapı, hata, hatanın türevi ve hatanın integralini kullanarak tasarlanan üç girişli Bulanık PID kontrolüne göre daha kullanışlıdır. Bunun nedeni ise, bu üç girişli Bulanık PID kontrolör yapısında örnek olarak her giriş için yedi adet dilsel değişken kullanıldığını varsayarsak, toplamda 343 tane kural tanımlanması gerekirken, Şekil 2.12'de gösterilen yapıda ise her bir kural tablosu için 49 olmak üzere toplam 98 kural tanımlanır.



Şekil 2.12. İki farklı kural tablosu kullanılarak oluşturulan iki girişli Bulanık PID kontrolör yapısı

İkincisi ise yukarıdaki yapıya göre daha kullanışlı olan Şekil 2.13'te gösterilen Bulanık PID kontrolör yapısıdır. Bu yapıda kural tabanı tek olduğundan her girişe ait 7 dilsel değişken seçilmesi durumunda toplamda sadece 49 adet kural yazılması gerekir. Ancak bu Bulanık PID kontrolör yapısında bir önceki yapıya göre daha az sayıda kazanç katsayıları veya ölçekleme faktörleri olmasından dolayı bu katsayıların doğru ayarlanması daha önemlidir.



Şekil 2.13. İki girişli tek kural tabanlı Bulanık PID kontrolör yapısı

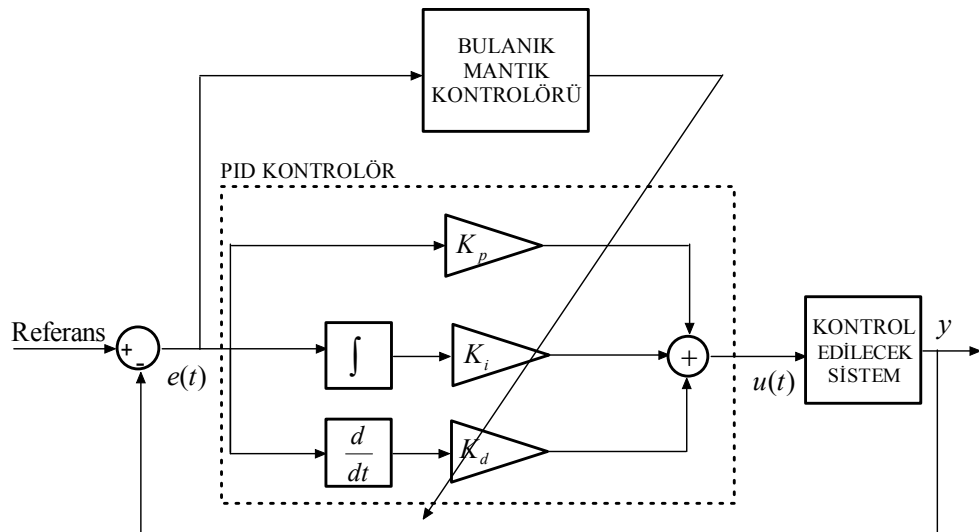
Şekil 2.13'te görüldüğü üzere kontrolör çıkışındaki PD kontrol işaretinin önce bir çıkış kazanç katsayısı ile çarpımından sonra da bir integratör ilavesiyle diğer bir çıkış kazanç katsayısı ile çarpılarak bu iki işaretin toplanmasıyla (u) kontrol işareti üretilir. Şekilden de anlaşılacağı üzere bu yapı, iki girişli tek kural tablosu kullanılarak oluşturulan Bulanık PID kontrolör yapısıdır.

Sonuç olarak, Bulanık PID kontrolör yapısı, Şekil 2.10 ve 2.11’de gösterildiği gibi, Bulanık PI ve Bulanık PD kontrolörlerin biraraya getirilmesiyle de elde edilebilir [58, 60-61]. Böylece, üç girişli Bulanık PID yapılarında olduğu gibi üç boyutlu tek kural tabanı oluşturma güçlüğü de ortadan kaldırılmış olur. Böylelikle, Bulanık PI ve Bulanık PD kontrolörlerin birarada kullanılması ya tek kural tablosu oluşturularak Bulanık PD kontrolörün çıkışına bir integratörle birlikte bir toplama elemanın eklenmesiyle ya da iki farklı kural tablosu oluşturularak Bulanık PI ve Bulanık PD kontrolörlerin çıkışlarının toplanmasıyla Bulanık PID kontrolör yapıları gerçekleştirilir.

2.6.4. Bulanık kazanç ayarlamalı PID kontrolör

Son yıllarda, bulanık mantık kontrol etkin bir şekilde araştırılmakta ve yararlanılmaktadır. Literatüre bakıldığında, bulanık ve gelenekselin değerlerini birleştirerek oluşturulan Bulanık PID kontrolör, bulanık ve PID’den oluşan karma kontrol ve bu kontrolün sağlamlığını geliştirmek için oluşturulan öz ayarlamalı bulanık kontrolörler geliştirilmiştir [62-63].

Klasik bir PID kontrolöründeki K_p , K_i ve K_d parametrelerin Şekil 2.14’te gösterildiği gibi bir bulanık kazanç ayarlamalı sistem tarafından çevrimiçi olarak ayarlanması ile oluşturulmuş bir kontrol yapısıdır [64-65].

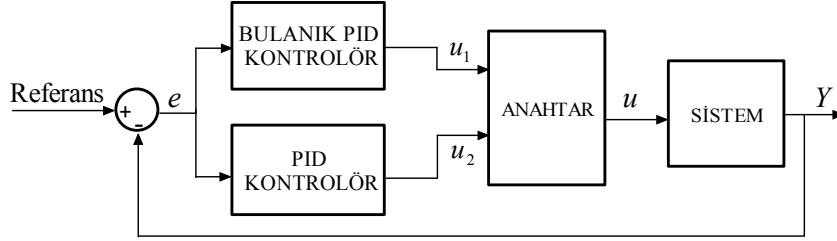


Şekil 2.14. PID kontrolör parametrelerini ayarlayan bulanık kontrolör yapısı

2.6.5. Karma tip Bulanık PID kontrolör

Karmaşık yapıya sahip gerçek sistemler için istenilen hedeflere ulaşmada yalnızca tek kontrolör yetersiz kalabilir ya da bunu sağlayabilen karma tipi bir kontrolör tasarımı gerekir. Bu karma tipi kontrolörde, yüksek hız bölgesinde etkin olan genellikle alt seviyeli kontrolör, düşük hız bölgesinde ise yüksek seviyeli kontrolör etkin olur.

Bu tip kontrol yöntemi, klasik PID ve Bulanık PID kontrolörlerin bir araya getirilmesinden oluşmaktadır. Bu tip kontrolde, klasik PID kontrolörün doğrusal sistemlerdeki başarımına ek olarak yüksek dereceden doğrusal olmayan sistemlerin kontrolünde de aktif rol oynaması sağlanmaktadır [66-68].



Şekil 2.15. Karma PID kontrolör yapısı

Şekil 2.15, karma tipi bir PID kontrol yapısını göstermektedir. Literatüre bakıldığında ise farklı karma PID yapıları, bir Bulanık PI kontrolörün klasik bir D kontrolör ile birleştirilmesi veya bir klasik PID kontrolörde P kontrolörün Bulanık P kontrolör ile değişiminden oluşan kontrolör gibi yapılar da mevcuttur [57, 67-68].

Şekil 2.15'te görüldüğü gibi iki farklı kontrolörün bir arada kullanılmasıyla oluşturulan bu yapıda, daha az üyelik fonksiyonlu ve daha küçük kural tabanlı bulanık kontrol algoritması tasarlanır. Temel olarak, bu kontrolör yapısında, PID kontrolör ve Bulanık PID kontrolör, sistemde oluşan anlık hataya bağlı bir fonksiyon ile ilişkili anahtarlama elemanı vasıtasıyla birleştirilmektedir. Şekilde gösterilen anahtar elemanı, sistem için gerekli olan kontrol sinyalini (u), u_1 ve u_2 den faydalanarak oluşturur. Ayrıca, bu anahtarlama elemanı, hangi kontrolörün öncelikli olarak kontrole başlayacağını da kararını verir.

2.7. Bulanık Mantık Kontrolörünün Tasarım Aşamaları

Bulanık kontrol tasarımında sırasıyla şu adımlar uygulanmalıdır:

- Kontrol edilecek sistemin giriş, durum ve çıkış değişkenleri belirlenir.
- Kontrolör içerisinde yer alan her bir değişkenin (giriş/çıkış) değişim aralığı alt bölgelere ayrılır ve her bir alt bölge dilsel olarak nitelendirilir.
- Bu alt bölgelerinin her biri birer bulanık küme üyelik fonksiyonu ile nitelendirilir.
- Giriş değişkenlerine ait bulanık kümelerle çıkış değişkenine ait bulanık kümeler arasında bulanık ilişkiler kurularak kurallar belirlenir.
- Giriş ve çıkış değişkenler için gerekli ölçeklemeler yapılır.
- Kontrolöre gelen girişler bulanıklaştırılır.
- Belirlenen kurallar ile bulanık çıkarım yapılır.
- O andaki girişe göre her bir kural tarafından belirlenen sonuç çıkış bulanık kümeleri elde edilir.
- Sonuç bulanık kümesi durulanır ve o andaki girişler için keskin çıkış değeri elde edilir.

2.8. Sonuç

Bu bölümde PID kontrolör, FOPID kontrolör ve Bulanık Mantık kontrolörü hakkında genel bilgiler verilmiştir. Kontrolörlere ait parametrelerin özelliklerinden, kontrole etkisinden ve doğrusal veya doğrusal olmayan sistemlerdeki başarımından söz edilmiştir.

Klasik kontrol yöntemleri arasında yer alan PID kontrolörün her bir giriş için oransal, türevsel ve integral olmak üzere üç kontrol parametresi ile ayarlandığı görülmektedir. Kontrolörde yer alan bu üç parametre iyi bir şekilde ayarlandığında, kontrol edilen sistemin tepki verme süresi, keskinliği ve kontrolörün dayanıklılığı ve performansı konusunda iyi sonuçlar alınabilmektedir.

PID kontrolörün genelleştirilmiş şekli olan λ kesir dereceli integratör ve μ kesir dereceli türevi içeren $PI^\lambda D^\mu$ (FOPID) kontrol sistemleri, güç elektroniği, sistem tanımlama, robotik uygulama, sulama kanal kontrolü, mekatronik sistemler ve ısı transferi sistemleri gibi endüstriyel ve çeşitli kontrol uygulamalarında geniş kullanım alanı bulmaktadır. FOPID kontrolörü tasarlanırken kontrolörün içerdiği 5 parametre $(K_p, K_d, K_i, \lambda, \mu)$ iyi bir şekilde ayarlanırsa, kontrolörün bozucu etkilere, yüksek frekans gürültüsüne ve sistem modelindeki belirsizliklere karşı dayanıklı olması sağlanabilir. Böylece, FOPID kontrolörün ayarlanma yöntemindeki sınırlamaların hepsi göz önünde bulundurularak K_p, K_d, K_i, λ ve μ parametrelerin en iyi değerleri bulunabilir.

Son yıllarda artan bir ilgiyle kullanılan bulanık mantık, sistemlerin kontrolünde yaygın bir şekilde kullanılmaktadır. Bulanık mantık kontrolün yüksek başarımlı gösterdiği uygulama alanları doğrusal olmayan, iyi tanımlanmamış, zamanla değişen sistemlerdir. Bu sistemlerin karmaşıklığı arttıkça bulanık mantık kontrol, klasik kontrolün yerini almaktadır. Ancak, bulanık mantık kontrolörünün istenilen sonuçlara ulaşabilmesi için sahip olduğu parametrelerin iyi bir şekilde ayarlanmış olması gerekmektedir. Uzman kişinin bilgisini sisteme yansıttığı için, bulanık kontrolörün sahip olduğu giriş ve çıkış kazanç katsayılarının, kural tabanının, üyelik fonksiyonlarının sahip olduğu parametrelerin iyi bir şekilde ayarlanmış olması şarttır. Parametreleri iyi ayarlanmış bir bulanık mantık kontrolör, zaman bölgesinde hızlı tepki verirken, hata değeri olarak da en az miktarda cevaba ulaşması beklenir.

Teknolojinin gelişmesiyle birlikte, doğada var olan birçok sistem yeniden keşfedilmektedir ve gerçekleştirilen çalışmalara esin kaynağı olmaktadır. Bu sayede canlılar üzerinde var olan öğrenme, akıl yürütme ve karar verme gibi özelliklerden esinlenerek geliştirilen evrimsel algoritma yöntemlerin, kontrol uygulamalarında kullanılması son yıllarda yaygınlaşmış ve sunduğu çözümlerden dolayı tercih nedeni olmuştur. Bir sonraki bölümde ve de tez çalışmasında tasarlanan kontrolörlere ait parametrelerin optimizasyonunda, doğrusal ya da doğrusal olmayan fonksiyonların evrimsel optimizasyonu için türevsel optimizasyon yöntemlerine alternatif ve iteratif bir yöntem olan parçacık sürü optimizasyon algoritması kullanılmıştır.

3. PARÇACIK SÜRÜ OPTİMİZASYON ALGORİTMASI

3.1. Giriş

Parçacık sürü optimizasyonu (PSO), son yıllarda sıkça kullanılan evrimsel optimizasyon yöntemlerinden biridir. 1995 yılında ilk olarak Kennedy ve Eberhart tarafından uçan kuş sürülerinin sosyal davranışlarından esinlenerek geliştirilen popülasyon temelli bir tekniktir [69]. PSO, kuş ve balık sürülerindeki gibi sosyal etkileşim ve iletişime dayanır.

Algoritma, evrimsel ve popülasyon temelli olduğundan, PSO algoritmasında yer alan üyeler, en iyi performansı gösteren grup liderini izleme eğilimi içerisindedirler. PSO algoritması, genetik algoritmalara benzer şekilde çalışan bir yöntemdir. Her iki algoritma da gelişmiş güzel üretilmiş bir popülasyon grubu ile çalışır ve bu popülasyon grubunu değerlendirecek uygunluk değerlerine sahiptir. Her iki algoritma popülasyon grubunu güncelleştirir ve arama yapar. Ancak PSO, çarpazlama ve mutasyon gibi operatörlerin kullanılmadığı diğer evrimsel tipi yöntemlerden farklıdır. Arama uzayı içerisinde en iyi konuma doğru aramayı yönlendirmek için fertler arasında bilgiyi sosyal bir şekilde paylaşmak amacıyla popülasyonu oluşturan sürü içerisindeki üyelerin arama yöntemi boyunca korunması ve hem kendilerinin hem de tüm sürünün geçmiş deneyimlerinden faydalanılması, algoritmanın farklılığını gösterir.

PSO algoritmasının bellek ve işlemci gereksinimleri düşük olduğundan bu algoritma kolayca uygulanabilir ve hesapsal olarak ağır değildir. Ayrıca göz önünde bulundurulmuş amaç fonksiyonunun değişimini elde etmek için matematiksel işlemlere ihtiyaç duymayan, yalnızca değerinin bilinmesinin yeterli olduğu algoritmadır.

PSO algoritması, birçok global optimizasyon problemleri için etkili bir yöntem olarak ortaya çıkmaktadır ve bazı durumlarda diğer evrimsel hesaplama teknikleri ile rastlanılan zorlukların üstesinden gelebilmektedir [69-70].

Evrimsel hesaplama tekniklerinde başlıca üreme (yeni popülasyon), mutasyon ve seçim operatörleri olmak üzere üç ana işlem vardır. PSO algoritmasında doğrudan bir üreme operatörü yoktur. Fakat hem parçacığın bir önceki en iyi konumuna doğru hem de sürünün en iyi parçacığına doğru olan bir parçacığın hızı, evrimsel hesaplamadaki üreme işlemine benzerdir [71-72].

PSO algoritmasında, her bir üye “parçacık” olarak çağırılır. Her bir parçacık, genetik algoritmalarındaki gibi uyumlu iki ebeveynden yeni bir fert oluşturma yerine, parçacığın sahip olduğu kendi deneyimi ve tüm sürünün ya da komşu parçacığın deneyimi ile devamlı olarak güncelleştirilen bir hız sayesinde, çok boyutlu arama uzayı içerisinde gezinir. Bunun yanında, PSO algoritmasının konum güncelleme işlemi, genetik algoritmanın mutasyon işlemine benzerdir.

PSO aslında, “en uyumlunun hayatta kaldığı” kavramını kullanmayan tek evrimsel algoritmadır. Algoritma, evrimsel algoritmalarda kullanılan bir seçim fonksiyonunu kullanmaz. Böylece daha düşük uygunluklu parçacıklar, optimizasyon süreci boyunca hayatta kalabilir ve potansiyel olarak arama uzayının herhangi bir noktasını ziyaret edebilirler [72].

3.2. Algoritmanın İçeriği

Evrimsel algoritmalara benzer şekilde PSO algoritmasında da her bir aday çözüm, parçacıkla (particle); popülasyon ise sürü ile ifade edilir. Her bir aday çözüm (particle) bir sonraki konumunu hız (velocity) vektörü, kendi tecrübesi (yerel en iyi - pbest) ve sürü tecrübesine (küresel en iyi - gbest) göre ayarlar.

PSO algoritmasında, yerel en iyi ve küresel en iyi olmak üzere iki değişken geliştirilir. Her parçacık, o ana kadar başardığı en iyi çözümlü kordinatlarını problem uzayında takip eder. Bu değer, pbest olarak adlandırılır. PSO tarafından takip edilen diğer bir en iyi değer, o ana kadar sürü içerisinde bulunan parçacıklar arasındaki en iyi değere sahip parçacığın elde ettiği değerdir. Bu parçacık, gbest olarak adlandırılır. Özetlemek gerekirse PSO, her bir iterasyonda parçacığın pbest ve gbest konumuna doğru olan hızının değişmesinden ibarettir.

PSO, çok boyutlu bir arama uzayında sürü parçacıklarının arama davranışına dayalı iteratif bir yöntemdir. Her bir iterasyonda, tüm parçacıkların hızları ve pozisyonları güncellenir. Her parçacık, sahip olduğu o anki konum ile pbest arasındaki mesafeye ve de gbest arasındaki mesafeye göre o anki konumunu ve hızını ayarlar. Burdan yola çıkılarak i. parçacığın konumundaki değişim hızı, yani parçacık hızı (V_i), aşağıda verildiği gibi hesaplanır.

$$V_i(n) = \chi(V_i(n-1) + \varphi_1 r_1 (pbest_i - P_i(n-1)) + \varphi_2 r_2 (gbest - P_i(n-1))) \quad (3.1)$$

P_i , i. parçacığın konumunu r_1 ve r_2 , 0 ile 1 arasında rastgele seçilmiş sayıları ve φ_1 ve φ_2 'de pozitif öğrenme katsayılarını gösterir. χ ise, aşağıda verilen deklemlerle belirlenen ve sınırlayıcı faktör olarak adlandırılan değişkendir.

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \quad \varphi = \varphi_1 + \varphi_2, \quad \varphi > 4 \quad (3.2)$$

Bu tanımlamalar yapıldıktan sonra tüm parçacıklar hız vektörleriyle bu iki en iyi konuma (pbest ve gbest) ulaşmaya çalışırlar. Bir parçacığın hızı V_i , konumu P_i olarak tanımlanırsa algoritmanın n. iterasyonu sonucunda i. parçacığın yeni konum bilgisi şu şekilde güncellenir [73].

$$P_i(n) = P_i(n-1) + V_i(n) \quad (3.3)$$

3.3. Pseudo Kodu ve Notasyon

PSO algoritmasında optimizasyon işlemi için izlenecek adımlar aşağıda açıklandığı gibi yapılabilir.

- Optimize edilecek parametrelerin alt ve üst sınırları belirlenir.
- Her bir parçacığın arama uzayındaki başlangıç konumları ve hızları, belirlenen sınırlar içerisinde rastgele atanır.
- Parçacıkların uygunluk değerleri ile pbest ve gbest konumları belirlenir.
- Her parçacığın uygunluk değeri, pbest'in uygunluk değeri ile karşılaştırılır. Eğer parçacığın o anki konumunun uygunluk değeri daha iyi ise parçacığın konumu ve uygunluk değeri, pbest'e ve pbest'in uygunluk değerine aktarılır.

- pbest vektörünün en iyi uygunluk değeri, gbest'in uygunluk değeri ile karşılaştırılır. Eğer pbest vektörünün en iyi uygunluk değeri daha iyi ise o parçacığın konumu ve uygunluk değeri, gbest ve gbest'in uygunluk değerine aktarılır.
- Parçacıkların hızları ve konumları Denklem (3.1) ve (3.3)'e göre güncellenir.
- Süreç, iterasyon sayısı tamamlanmaya ya da belirlenen en iyi uygunluk değeri elde edilinceye kadar üçüncü adımdan itibaren tekrarlanır.
- Evrim sonunda elde edilen gbest en iyi konum, problemin çözümü olarak alınır.

PSO algoritması ile elde edilen çözüm, optimal bir çözümdür. Ancak elde edilen çözüm her zaman en uygun çözüm olmayabilir. PSO, diğer evrimsel optimizasyon yöntemlerine göre yazılımı daha basit, daha az kod satırına ihtiyaç duyan ve daha az iterasyonla yakınsayabilen bir algoritmadır. Şekil 3.1'de standart bir PSO algoritmasının komut dizimi görülmektedir.

```

Rasgele başangıç sürüsünü oluştur.(P adet parçacık)
Tekrarla

    i=1'den P'ye kadar yap
        pi parçacığı ile sistemi çözümle
        Eğer uygunluk (pi) < uygunluk ( pbesti ) ise pbesti = pi
        Eğer uygunluk (pi) < uygunluk ( gbest ) ise gbest = pi
        i. parçacığın hızını Denklem 3.1 ile hesapla
        i. parçacığın konumunu Denklem 3.3 ile güncelle

    son

Sonlandırma kriteri sağlanıncaya kadar

```

Şekil 3.1. PSO algoritmasının genel işleyiş yapısı

PSO algoritması ile optimizasyonu yapılacak bir probleminin PSO yöntemine uyarlanması için çözümün gösterimi ve uygunluk fonksiyonun tanımlanması gerekmektedir. Algoritmanın en büyük avantajlarından biri parçacıkların reel sayı olarak tanımlanabilmesidir. Genel bir optimizasyon problemi için PSO algoritmasında yer alacak değişkenlerin gösterimi ile ilgili notasyon aşağıda verildiği gibi belirlenebilir.

P_i : Sürüdeki i.ci parçacık; $P_i = [p_{i1} p_{i2} p_{i3} \dots p_{im}]$

p_{ij} : i.ci parçacığa ait j.ci elemanın değeri ($j = 1,2,3,\dots,m$)

V_i : Sürüdeki i.ci parçacığın hızı; $V_i = [v_{i1} v_{i2} v_{i3} \dots v_{im}]$

v_{ij} : i.ci parçacık hızına ait j.ci elemanın değeri ($j = 1,2,3,\dots,m$)

$swarm$: ρ adet parçacıktan oluşan sürü (sürü boyutu, ρ); $swarm = [P_1 P_2 P_3 \dots P_\rho]^T$

$pbest_i$: i.ci parçacığın en iyi konumu; $pbest_i = [pbest_{i1} pbest_{i2} pbest_{i3} \dots pbest_{in}]$

$pbest_{ij}$: j.ci boyuta göre i.ci yerel en iyi parçacığın konum değeri ($j = 1,2,3,\dots,n$)

$gbest$: Küresel en iyi parçacığın konumu; $gbest = [gbest_1 gbest_2 gbest_3 \dots gbest_n]$

Yukarıdaki adımlar oluşturulurken başlangıç parçacık miktarı, tipik olarak 20–50 arasında seçilen miktarda belirlenir. Bu miktar problemin büyüklüğüne göre daha düşük ya da daha yüksek olabilir. Parçacıkların boyutu, optimize edilecek probleme bağlı bir değişkendir. Dolayısıyla, optimize edilecek fonksiyon kaç boyutlu ise parçacıkların boyutları da bu miktarda olacaktır. Parçacıkların değer aralığı, yine probleme bağlı bir parametredir. Optimize edilecek değişkenlerin değer aralığını belirtir.

3.4. Sonuç

Bu bölümde, Stewart Platform'un kontrolü için tasarlanan kontrolörlere ait parametrelerin ayarlanmasında kullanılan parçacık sürü optimizasyon algoritmasından bahsedilmiştir. Kuş ve balık sürülerinin sosyal davranışları gözlemlenerek geliştirilen PSO algoritması, global optimumu hızlı bir şekilde yakalayan, basit ve kolay kodlanabilir bir algoritmadır.

PSO algoritmasında ideal olarak tüm tasarım parametreleri bağımsız birer değişken olarak kabul edilir ve en iyi çözüm aranır. Ayrıca, PSO yöntemin en büyük üstünlüğü klasik optimizasyon algoritmalarından farklı olarak türev bilgisine, ağır matematiksel hesaplamaya ihtiyaç duymamasıdır. Bu üstünlük, çalışılan sistem modelinin bilinmediği durumlarda çok belirgin olarak ortaya çıkmaktadır.

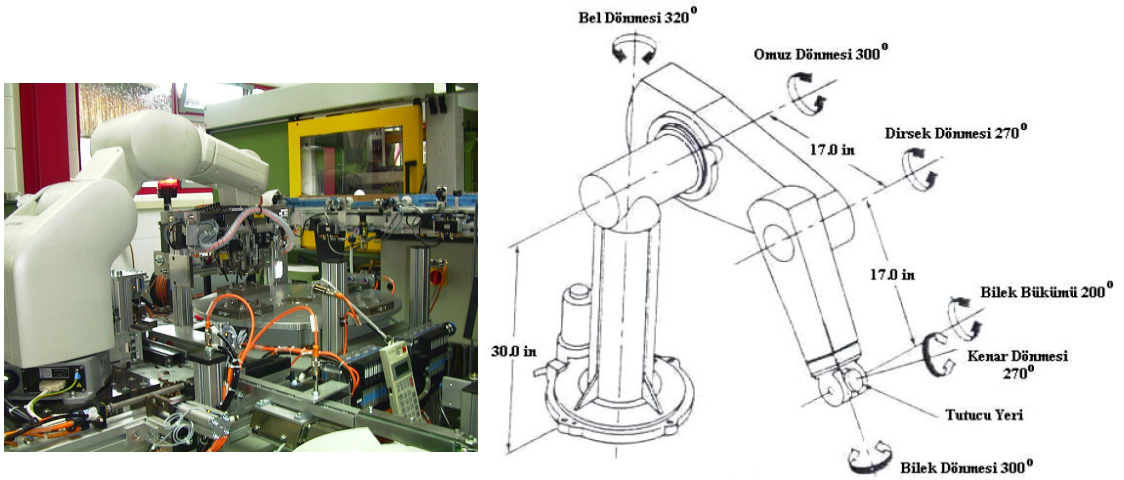
4. SERİ ROBOTLAR

4.1. Giriş

Robot sistemlerin endüstri alanlarında kullanılmalarıyla ilgili olarak robot tanımı, Amerikan Robot Enstitüsü (RIA) tarafından şu şekilde yapılmıştır. "Robot, çeşitli görevleri yapmak amacıyla değişik şekillerde programlanmış hareketlerle; nesnelerin, gereçlerin ya da özel düzeneklerin taşınması için tasarlanmış çok işlevli bir manipülatördür."

Robot manipülatörü, uç-işlevcisi yardımıyla kaynak yapma, kesme, tutma, delme, konumlama gibi uygulamalarda kullanılan bir robot koludur. Bir robot manipülatörü, temel olarak eklemlerle birbirine bağlanan bir dizi katı bağlardan oluşur. Bu bağlarda bulunan eklemler genelde doğrusal ve dönel şeklindedir. Son bağ ise uç işlevci olarak adlandırılır. Adından da anlaşılacağı gibi bu bağa kesme işlemi ya da kaynak işlemi gibi işlemleri yerine getirebilen aletler takılabilir. Genellikle endüstride yaygın olarak kullanılan manipülatörler, seri bağlanmış bağlardan oluşan, açık döngülü mekanizmalardır. Bir seri manipülatör, her bir bağın bir sonraki bağa hareketli bir eklemlerle bağlandığı ve uç işlevciyi referans eksene (base coordinate frame) bağlayan seri bağlardan oluşur. Bu tip manipülatörler; serbestliği sağlayan bir koldan, el becerisi sağlayan bir bilekten ve robotun yapması gereken görevi tamamlayan sonladırıcıdan oluşmaktadır. Ayrıca, manipülatörün hareketini yöneten ve kontrol eden kontrol sistemi, manipülatörün ve çevrenin durumunu gözetleyerek veri üreten algılayıcılar (sensors) ve eklemlerin hareketi ile manipülatörün hareketini sağlayan hareketlendiriciler (actuators-motors) robot manipülatörünün içerisinde yer almaktadır. Endüstriyel robotlar olarak da adlandırılan bu robot kolu manipülatörlerin daha çok 6 eksenli vardır. Şekil 4.1'de endüstriyel bir robot kolunu ve buna ait eksenler ve hareket yönleri gösterilmektedir.

Bu seri robot kolları yapısal olarak insan koluna benzediklerinden dolayı, insan kolunun avantaj ve dezavantajlarına sahiptirler. Robot kolunun yetiřebileceđi toplam alana alıřma uzayı denir. Seri robotlar alıřma uzaylarına gre sınıflandırılırlar. Bir robot hareketinin kapasitesi, kontrol edilebilmesi mmkn olan eksenlerdeki hareketlerle belirlenir. Genel olarak kresel, silindirik, mafsallı kol, kartezyen ve Scara tipi robot kolları řeklinde dirler. alıřma uzayı ve eriřebildikleri mesafe aısından seri mekanizmalar paralel mekanizmalara gre daha stn yeteneklere sahip olmasına rađmen dinamik zellikleri karřılařtırıldıđında paralel mekanizmalar kadar iyi deđildir.



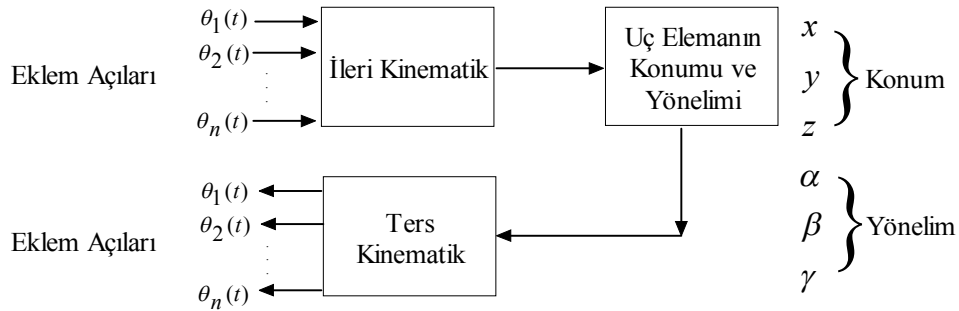
řekil 4.1. Endstriyel bir maniplatrn eksenleri ve hareket ynleri [78]

Robot kinematiđi; bađların konumları, hızları ve ivmeleri arasındaki iliřkidir. Kısacası, robotun uzaysal yer deđiřtirmesinin zamana bađlı bir fonksiyonunun analitik tanımıyla uđrařır. Robot kolu kinematiđi, bir robotun sabit bir referans koordinat sistemine gre hareket etmesiyle bu hareketten kaynaklanan kuvvet/moment etkisini gze almadan robotun hareketini inceler. zellikle eklem deđiřkenleri uzayı ile robot kolunun u-iřlevcisinin konumu ve ynlenmesi (orientation) arasındaki bađıntılıları inceler.

Herhangi bir robot kolunda, her bir bađ iin drt byklk verilerek kinematike tanımlanır. Bu byklkler vasıtasıyla mekanizmanın tanımı geleneksel olarak Denavit-Hartenberg gsterimi ile ifade edilir [74]. Denavit-Hartenberg ynteminde drt ana deđiřken kullanılarak robot kolu kinematiđi ıkarılır. Bu deđiřkenler, iki eksen arasındaki bađ uzunluđu (link length) , (i-1) ile i eksenleri arasındaki bađ aısı

(link twist), üst üste çakışan bağlar arasındaki eklem kaçıklığı (joint offset) ve iki bağ arasında oluşan eklem açısı (joint angle)'dır. Bu dört değişkene D-H değişkeni denir [75].

Robot kolu kinematiği, ileri ve ters kinematik olarak ikiye ayrılır. İleri kinematik probleminde, robot eklemlerinin yapacağı açılar belirlenir ve istenen her açı değeri kadar eklemler hareket ettirilirse robotun uç-elemanı, robotun çalışma uzayı içinde bir noktaya gelir. Buna ileri kinematik denir. İleri kinematik probemi çözümünde genel olarak iki yaklaşım vardır. Bunlardan birincisi, geometrik yaklaşım; bu yaklaşım manipülatör duruşuna bağlı olarak geometrik şekilden yararlanır. Diğer yaklaşım ise cebirsel yaklaşımdır. Bu yaklaşım manipülatörün parametreleri ve eklem değişkenleri arasındaki cebirsel ilişkilerden yararlanır. Ters kinematik problemi, manipülatörün uç işlevcisinin konumu ve yönelimi verildiğinde bu konum ve yönlenmeyi elde etmek için kullanılan olası tüm eklem açılarının hesaplanmasına denir. Şekil 4.2, ileri ve ters kinematik problem arasındaki ilişkiyi göstermektedir.



Şekil 4.2. İleri ve ters kinematiğin şematik gösterimi

Ters kinematik problemi için iki temel çözüm tekniği vardır: Birincisi, verilen konfigürasyon verisine göre analitik olarak eklem değişkenlerinin çözümüdür. İkinci tip çözüm ise sayısal tekniklere dayanarak eklem değişkenlerinin elde edilmesidir [76]. Analitik çözümler, geometrik ve cebirsel çözümleri ve de bir uç nokta konfigürasyonu için hızlı ve etkin bir hesaplamayı sağlayan kapalı form çözümleridir. Sayısal çözümler ise etkileşimli bir işlem sırasının uygulandığı doğrusal olmayan aritmetik eşitliği, geometrik ve aritmetik özellikler kullanarak çözüme giden çözümlerdir. Bundan dolayı, ters kinematik, ileri kinematikten çok daha zor bir problemdir. Ters kinematik probleminin çözümü hesapsal olarak

büyüktür ve gerçek zamanlı manipülatör kontrolünde genellikle çok uzun zaman alır. Ayrıca, çözümsel olarak karmaşık, doğrusal olmayan denklemler içermesi, eklem yapısına bağlı olması ve tekilliklerin var olması sebebiyle ters kinematik, çözülmesi oldukça zor olan problemleri içerir. Ters kinematik problemi için matematiksel çözümler her zaman fiziksel çözümleri karşılamayabilir ve çözüm yöntemi robotun yapısına bağlıdır.

Robot tasarımı ve kontrolü için yapılacak ilk adım, robotun kolunun dinamik modelini çıkarmak olacaktır. Serbestlik derecesi üç veya daha fazla olan bir seri robotun dinamik denklemlerini çıkarmak oldukça zordur. Bu denklemleri oluşturacak bir yöntem gereklidir. Bu yöntem; robotun hareket denklemlerini çıkartırken kolayca formüle edilebilmeli ve çok karışık olmamalı, elde edilen model gerçek sistemle ilgili sonuçları doğru vermeli, robotun hem ileri hemde ters dinamik problemlerini çözebilmeli, genel olmalı ve elde edilen dinamik model, sistemin tüm kısıtlarını göz önüne almalı.

Robot kol dinamiği, robot kol hareketindeki denklemlerin matematiksel formülasyonu ile uğraşır. Bundan dolayı manipülatörün hareket denklemi, manipülatörün dinamik davranışını tanımlayan bir dizi matematiksel denklem ifadeleridir. Bu dinamik denklem, doğrusal olmayan, birbirleriyle bağlantılı ve ikinci dereceden bir diferansiyel denklem sistemini gösterir. Robot kolu dinamiğini formüle edebilmek için Lagrange-Euler (L-E) [77,78], Recursive-Lagrange (R-L) [79], Newton-Euler (N-E) [80], Genelleştirilmiş D’Alambert Prensipli (G-D) [81] ve Kane metodu [82] gibi yaklaşımlar önerilmiştir. Bu yaklaşımlar içinde en çok tercih edileni L-E ve N-E yaklaşımlarıdır. Buradan elde edilecek hareket denklemleri aynı fiziksel robotik manipülatörün dinamik davranışını tanımlayacağından bu anlamda birbirine eşdeğerdir. Bununla beraber, bu denklemlerin yapısı değişik elde edilme nedenlerine bağlı olarak farklılık gösterebilir. Bu denklemler, robot kolunun hareketinin bilgisayar simülasyonu, robot koluna uygun kontrol kurallarının tasarımı, kinematik tasarım ve robot kolunun yapısının hesaplanması için oldukça faydalıdır.

L-E formülasyonuna göre yapılan manipülatörün dinamik modelinin elde edilmesi, diğerlerine göre daha basit ve sistemattir. Bu yaklaşım sistemdeki toplam enerji ve

iş ile tanımlanır. Denklem (4.1)'de verildiği gibi bu işlemde kinetik ve potansiyel enerjiden faydalanılır.

$$L = K(q, \dot{q}) - P(q) \quad (4.1)$$

L-E hareket denklemleri aşağıda verilen ifadeye dayanır.

$$\frac{d}{dt} \frac{dL}{d\dot{q}} - \frac{\partial L}{\partial q} = \frac{d}{dt} \left(\frac{\partial K(q, \dot{q})}{\partial \dot{q}} \right) - \frac{\partial K(q, \dot{q})}{\partial q} + \frac{\partial P(q)}{\partial q} = \tau \quad (4.2)$$

Denklemlerde geçen ifadeler;

L , Lagrange Fonksiyonu,

$K(q, \dot{q})$, Robot eklemlerin toplam kinetik enerjisi,

$P(q)$, Robot eklemlerin toplam potansiyel enerjisi,

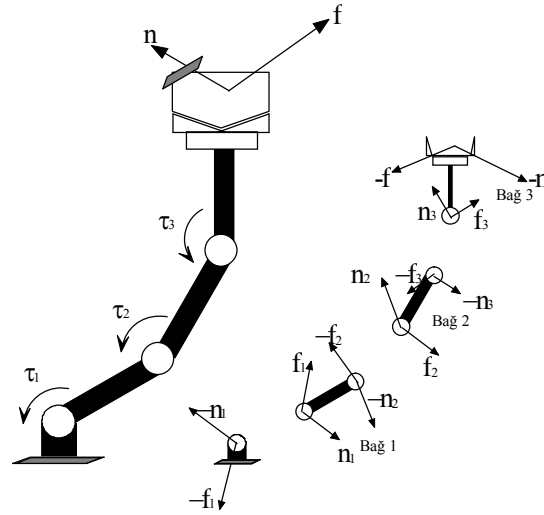
q , Robotun genelleştirilmiş koordinatı. Dönel eklem için θ , Kayar eklem için d ,

\dot{q} , Genelleştirilmiş koordinatın zaman göre birinci türevi,

τ , Robotun ekleminden sisteme uygulanan genelleştirilmiş kuvvet ya da moment,

şeklinde tanımlanırlar.

Diğer bir yaklaşım olan N-E, hareket denklemleri kullanılarak genelleştirilmiş kuvvetlerin/torkların hesaplanmasına dayanır. Denklemleri türetmek basittir, fakat karışık ve vektörel çarpımlı ifadeler içerir. Newton-Euler formülasyonuna ait denklemler literatürde gösterildiği gibi robot eklemlerine ardışık olarak uygulanır [75, 76, 83-84]. Şekil 4.5'te gösterildiği gibi ardışık olarak ileri yönde tekrarlanan denklemler sayesinde lineer hızlar, açısal hızlar, lineer ivmeler, açısal ivmeler gibi kinematik bilgiler elde edilir. Şekil 4.3 ve Şekil 4.4'te gösterilen geriye doğru hesaplanan denklemlerde ise, her bir eklem üzerine manipülatörün tutucusundan uygulanan momentlerin ve kuvvetlerin elde edilmesi sağlanır.

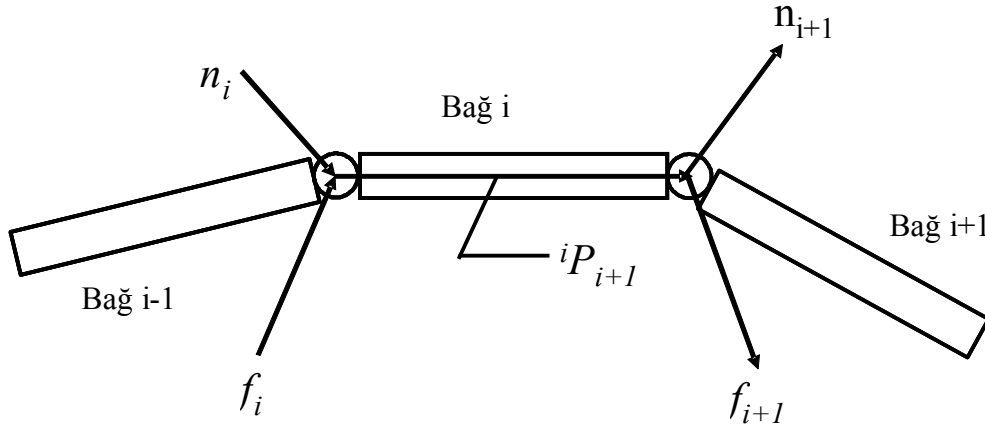


Şekil 4.3. Bağlar üzerinde oluşan kuvvet ve torklar

$${}^i f_i = {}^i R^{i+1} f_{i+1} \quad (4.3)$$

$${}^i n_i = {}^i R^{i+1} n_{i+1} + {}^i P_{i+1} \times {}^i f_i \quad (4.4)$$

Denklemlerde yer alan f_i bağ (i-1) tarafından bağ (i) üzerinde harcanan kuvveti göstermektedir. n_i , bağ (i-1) tarafından bağ (i) üzerinde sarf edilen torku göstermektedir. R , dönme matrisini P ise konumu göstermektedir (Şekil 4.4).

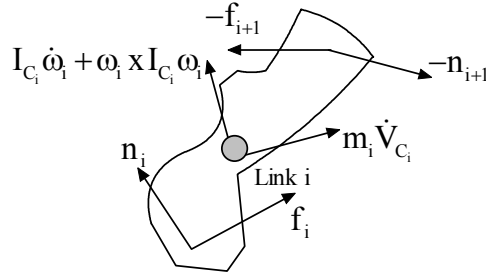


Şekil 4.4. Kuvvet ve tork yayılımı

Newton-Euler yaklaşımına göre bir bağ üzerinde oluşan kuvvet ve moment denklemleri aşağıda verildiği gibi hesaplanabilir (Şekil 4.5).

$$F_i = m_i \dot{V}_{C_i} \quad (4.5)$$

$$N_i = I_{C_i} \dot{\omega}_i + \omega_i \times I_{C_i} \omega_i \quad (4.6)$$



Şekil 4.5. Bağ (i) üzerindeki dinamik kuvvetler

Denklem (4.5) ve (4.6)'da yer alan \dot{V}_{C_i} doğrusal ivmeyi, I_{C_i} atalet tensörünü, ω_i ve $\dot{\omega}_i$ sırasıyla açısal hız ve açısal ivmeyi gösterir. Bu denklemler, bağın atalet kuvveti ve momentidir.

Yukarıdaki denklemlerden yola çıkarak çıkarak Newton-Euler yaklaşımı bağ (i) üzerinde uygulandığında, bu bağ üzerinde oluşan kuvvet ve tork, aşağıda verilen denklemlerle ifade edilebilir.

$$f_i = F_i + f_{i+1} \quad (4.7)$$

$$n_i = N_i + n_{i+1} + P_{C_i} \times F_i + P_{i+1} \times f_{i+1} \quad (4.8)$$

Bu denklemlerden eklem torkunu elde etmek için eklem eklem değişkenine bakılır. Eğer eklem kayar ya da dönel eklem olması durumunda aşağıdaki denklem kullanılır.

$$\tau_i = \begin{cases} n_i \cdot Z_i & \text{Dönel} \\ f_i \cdot Z_i & \text{Kayar} \end{cases} \quad (4.9)$$

Newton-Euler yaklaşımı kullanılarak en son bağdan geriye doğru taban (base) eksenine kadar olan kuvvet veya momentler hesaplanır. Bu hesaplama için gerekli olan denklemler aşağıda verilmiştir.

$${}^i f_i = {}^i R^{i+1} f_{i+1} + {}^i F_i \quad (4.10)$$

$${}^i n_i = {}^i N_i + {}^i R^{i+1} n_{i+1} + {}^i P_{C_i} \times F_i + {}^i P_{i+1} \times {}^i R^{i+1} f_{i+1} \quad (4.11)$$

Yukarıdaki denklemlerin tümü kullanılarak eklem değişkeni türüne göre manipülatörün Newton-Euler hareket denklemleri çıkartılabilir.

Manipülatörün dinamik modelinin elde edilmesinden ve bu modeller kullanılarak istenen sistem tepkisini veya performansını başarmak üzere kontrol stratejilerine başvurulur. Bir robot kolunun kontrol analizi iki şekilde yapılır. Birincisi kolun bir başlangıç noktasından istenen hedef pozisyonun yakınına belirli bir yörünge üzerinden yapılan hareket kontrolüdür. İkincisi ise uç işlevcisinin sensörlerinde gelen geri beslemeyi kullanarak bir işi tamamlamak üzere nesneyle dinamik olarak etkileşirken yaptığı hassas hareket kontrolüdür.

Robot kontrolü robotik biliminin omurgasıdır. Bir robot manipülatörünün arzu edilen görevi otomatik olarak nasıl yapacağı konusuna dayanır. Dolayısıyla, robot kontrolü, robot kontrolörlerinin tasarlanmasından ibarettir. Genellikle bunlar, belirtilen bilgisayar programları vasıtasıyla bir denklem ya da bir algoritma formunda olur. O halde kontrolörler, robotun kontrol sistemi diye adlandırılan kısmını oluşturur. Bu sistem, fiziksel olarak bilgisayar, data toplama birimi, eyleyiciler (genellikle elektrik motorları), robotun kendisi ve bir takım elektronik devrelerden meydana gelir. Böylece, bir robot kontrolörünün tasarımı ve tamamen uygulanabilmesi bahsedilen bu disiplinlerin her birine bağlıdır.

Robot kolu kontrolünde en basit kontrolör, orantı-integral-türev tipi olan (PID) yapıdır. Genel olarak, kontrolörlerin bu tipi, dinamik denklemlerden oluşan robot modeline dayalı tasarlanır. İcra edeceği göreve bağlı olarak yaygın bir şekilde robotik kollarında kullanılan bu kontrolörler, her zaman en iyi performans veremeyebilir. Performansı arttırmak için manipülatörün fiziksel parametre değerlerini içeren dinamik modelin tam bir bilgisinin gerektirdiği model tabanlı kontrolörler tasarlanır. Diğer yandan detaylı dinamik modele ihtiyaç duymayan ve doğrusal olmayan sistemlerde gayet iyi sonuçlar veren değişken yapılı kontrolörler (Variable Structure Controllers), bulanık mantık kontrolörleri (Fuzzy Logic Controllers), yapay sinir ağı kontrolörler (Neural Network Controllers), sinirsel bulanık kontrolörler (Neuro-Fuzzy Controller) gibi literatürde yaygınlaşmış ve uygulama alanı bulmuş kontrolörlerdir.

Literatürde robot kolu kontrolü üzerine yapılan bazı çalışmalar özetlenecek olursa; Duran ve Ankaralı [85], üç serbestlik dereceli PUMA tipi robot kolunun üç elemanının PID yöntemi ile yörünge kontrolünü yapmışlardır. Robota ait kinematik ve dinamik denklemleri çıkardıktan sonra yörünge planlanmasını kartezyen koordinatlarda yaparak istenen yörüngeyi gerçekleştirmek için gerekli mafsal değişkenlerini ters kinematik analizle hesaplamışlardır. Yörüngeyi izlemek üzere tasarlanan PID kontrolörüne dış bozucu etkiler altında sikloid, harmonik ve polinom yapılı üç farklı yol alma fonksiyonu giriş olarak verilmiş ve elde edilen sistem cevapları istenen yörüngeye yakınsaması açısından karşılaştırma yapılmıştır. Elde edilen çıkış yörüngeleri değerlendirilerek hata analizi yapılmıştır. Çıkan hata miktarlarının kabul edilebilir seviyede olduğunu ifade etmişlerdir.

Endüstride yaygın olarak kullanılan PID kontrolörünün tasarımı sırasında üç parametrenin belirlenmesi gerekmektedir. Fakat özellikle robotikte sistem nonlineerlik içerdiğinden dolayı sabit katsayılı PID kontrolörü her zaman iyi neticeler vermeyebilir. Dolayısıyla sistemdeki değişimleri dikkate alabilmek için bu katsayıların da ayarlanması gerekmektedir. Bunun için Zhao ve arkadaşları [86] PID kontrolörünün katsayılarını belirlemek için bulanık mantık kullanmışlardır. Bulanık mantıklı kontrolörün girişleri hata ve hatanın türevi, çıkışları ise PID kontrolörünün katsayılarıdır. Bu yapıya göre elde edilen sonuçlara bakıldığında önerilen kontrolör yöntemi, sabit katsayılı kontrolörden daha iyi olduğu görülmüştür. Santos ve Dexter [87] kontrol kuralını uyguladıkları sistemin kontrol performansını iyileştirmek için bulanık mantık ile PID katsayılarını ayarlamışlardır. Bu katsayıların hangi aralıkta değişebileceğini deneyler yaparak belirlemişlerdir. Zulfatman ve Rahmat [88], elektro-hidrolik bir eyleyicinin performansını arttırmak için öz ayarlamalı bir bulanık PID kontrolör geliştirmişlerdir. PID kontrolörünün her bir parametresi, uygun bulanık kurallar seçilerek ayarlanmıştır. Geleneksel PID kontrolörüyle karşılaştırılarak hidrolik sistemin performansı önemli ölçüde geliştirilmiştir. Khoury ve arkadaşları [89], 5 serbestlik dereceli bir robot kolu üzerinde bulanık PID kontrol uygulaması yapmışlardır. Ayrıca, aynı robot kolunun dinamik modeli kullanılarak hesaplamalı tork kontrol ve direk ayarlamalı kontrol metotlarına göre karşılaştırma yapılarak önerilen bulanık PID kontrolörünün başarısını doğrulamışlardır. Alassar ve arkadaşları [90], bağımsız eklem kontrol yöntemine dayanan 5 serbestlik dereceli

robot kolu (Lynx6) için klasik PID kontrolörün dezavantajlarının üstesinden gelecek bir bulanık mantık kontrolörü önermişlerdir. Bunun yanında, Lynx6 robotuna uygulanan hem PID hem de bulanık mantık kontrolörlerinin cevaplarına bakılarak simülasyon yapmışlardır. Simülasyon sonuçlarına göre önerilen bulanık mantık kontrolörünün PID kontrolöründen daha iyi sonuçlar ürettiğini göstermişlerdir. Ham ve arkadaşları [91], robot manipülatörleri için kontrol sistemin performans gereksinimlerine ve kararlı bölgelerine dayalı olarak seçilen bulanık kümelerden oluşan sağlam bir bulanık kontrolör geliştirdiler. Önerilen bu kontrolörü, robot yörünge kontrol problemine uyguladılar ve doğrusal olmayan standart bir sağlam kontrolör ile karşılaştırdılar. Simülasyon sonuçlarına bakıldığında önerilen kontrolörün verimli olduğunu ve daha üstün yörünge izleme performansına sahip olduğunu göstermişlerdir. Malki ve arkadaşları [92], zamanla değişen yüklerden dolayı belirsizlikler altında esnek eklemlili bir robot kolu için bulanık bir PID kontrolörün tasarımı ve deneyini sunmuşlardır. Yapılan deneysel sonuçlar, önerilen bulanık PID kontrolörünün kayda değer izleme performansı gösterdiği ve de belirsizliklerin olduğu esnek eklemlili robot kollarının kontrolünde çok sağlam bir kontrolör olarak kullanılabilceğini göstermişlerdir. Ayrıca, laboratuvarında bir DC motor ile sürülen esnek eklemlili bir robot kolunu kullanarak geleneksel ve bulanık PID kontrolörlerin kontrol performansı karşılaştırılmıştır. Yapılan tüm deneylerde bulanık PID kontrolörünün, geleneksel PID kontrolöründen daha iyi sonuçlar ürettiğini göstermişlerdir.

Robot kollarının kontrolünde kullanılan bulanık mantık gibi detaylı bir dinamik modele ihtiyaç duymayan ve doğrusal olmayan sistemlerde gayet iyi sonuçlar veren diğer bir kontrolör tipide yapay sinir ağlarıdır. Daha az ön bilgiye ve daha çok değişen sistem dinamiğine karşı çalışma performansı belirli sınırlar içinde kalabilen yapay sinir ağları kontrol sistemlerinin tasarımında kullanılmaya başlanmıştır. An ve arkadaşları [93], parametrik belirsizliklere sahip robot sistemi için sağlam bir kontrol yapısı sunmuşlardır. Robot dinamiğindeki bu belirsizlikleri karşılamak için katmanlarında bulunan sinirlere ait doğrusal olmayan fonksiyonlu yapay sinir ağı bir kontrolör kullanılmıştır. Önerilen kontrolörün hem yapısal belirsizliklere karşı (yük parametreleri gibi) hem de yapısal olmayan belirsizliklere (sürtünme modeli ve bozucu etki gibi) karşı sağlamlığını göstermişlerdir. Ayrıca, kontrol yapısının

güvenirligini, 8 eksenli bir robot manipülatörünün bilgisayar simülasyonu ve deneyi ile göstermişlerdir. Hesselroth ve arkadaşları [94], iki video kameradan geri beslemeyle 5 eklemlili pnömatik robot kolunu (Soft-Arm) ve tutucuyu kontrol etmek için 200 sinirden oluşan üç boyutlu çalışma uzayını temsil eden yapay sinir ağı bir kontrolör geliştirmişlerdir. Zannatha ve arkadaşları [95], yapay sinir ağlarına dayanan iki dönel eklemlili düzlemsel bir robot manipülatörü için 2 kinematik kontrol düzeneği geliştirmişlerdir. Birinci olarak uç işlevcinin konum kontrolünü kontrol etmek için 3 katmanlı yapay sinir ağını tasarlayıp eğitmişlerdir. İkinci düzenekte ise yapay sinir ağına dayanan diferansiyel kinematik kontrolör için ağı tasarlayıp eğitmişlerdir. Kullanılan sinir ağlarının gizli katmanlarında doğrusal olmayan fonksiyon ve çıkış katmanlarında ise doğrusal fonksiyona sahip sinirler kullanılmıştır. Bu sinir ağları, geriye doğru yayımlı hata algoritması ile eğitilmiş 3 katmanlı ileri doğru ağlardır.

Zhang ve arkadaşları [96], robot yörünge takibi için uyarlamalı sinirsel değişken yapılı iki PD kontrolör düzeneği sunmuşlardır. Birinci düzeneğin için kapalı çevrimli sistemin kararlılığını ve izleme hatasının asimtotik yakınsamasını sağlayacak RBF tipi yapay sinir ağı ve değişken yapılı bir PD kontrolör geliştirilmiştir. İkincisinde kontrol keskinliğini ve yakınsama hızını arttıracak ve yerel yapay sinir ağlarının yetersizliklerini giderecek ve de değişken yapılı kontrol girişinin gürültülerini azaltacak yapay sinir ağı ve değişken yapılı karma tipli bir kontrolör geliştirilmiştir. Simülasyon sonuçlarında bu iki tür kontrol düzeneğinin verimliliğini ve de ikinci düzeneğin daha avantajlı olduğunu göstermişlerdir.

Marichal ve arkadaşları [97], mobil bir robotun engelli yollarda engele takılmadan hedefine ulaşabilmesi için sinirsel bulanık kontrolör geliştirmişlerdir. Önerdikleri sinirsel bulanık kontrolör tarafından otomatik olarak üretilen üyelik fonksiyonları ve kurallar sayesinde robotun engele takılmadan hedefine ulaşabildiğini göstermişlerdir. Low ve arkadaşları [98], gerçek zamanlı endüstriyel uygulamalar için dinamik bir bulanık sinir ağı kontrolör yaklaşımını sunmuşlardır. Bu bulanık sinir ağı kontrolör, SCARA tipi bir robota gerçek zamanlı uygulamışlar ve simülasyon sonuçlarında, ters kinematik probleminin öğrenebildiğini ve hatayı hızlı bir şekilde sıfıra düşürebildiğini göstermişlerdir. Alavandar ve Nigam [99], PD+I kontrolör yapısına

dayanan uyarlanabilir yapay sinirli bulanık çıkartım sistemi (Adaptive Neuro Fuzzy Inference System – ANFIS)’ni kullanarak 6 serbestlik dereceli bir PUMA 560 robot kolunun kontrolünü sunmuşlardır. Sayısal simülasyona bakıldığında yörünge takip probleminde önerilen yaklaşımın verimliliğini göstermişlerdir. Ayrıca, PID kontrolörüne göre karşılaştırmalı değerlendirme yaparak kontrolör tasarımını geçerli kılmak için bulanık PID+I kontrolörü sunmuşlardır. Sunulan sonuçları, ANFIS kontrolörü kullanılarak PID ve bulanık PD+I kontrolörlerinden daha tatmin edici bir izleme doğruluğunun başarılabilildiğini göstermiştir.

Durmaz [100] tezinde, dinamik denklemleri çıkarılan bir, iki ve üç serbestlik dereceli robot kollarının kontrolünde kullanmak üzere yapay sinir ağı, bulanık mantık ve model referans adaptif kontrol yöntemlerini incelemiş, elde edilen bilgilerin ışığı altında model referansa dayalı dinamik sinirli bulanık mantık kontrolör (DSBMK) önermiştir. Robotun dinamik modeli çıkartılırken daha gerçeğe yakın simülasyon sonuçları verebilen bir model elde edebilmek için yerçekimi ivmesi ve sürtünme kayıplarını dinamik denklemlere dâhil etmiştir. Çalışmanın sonunda bir, iki ve üç serbestlik dereceli robot kollarına verilen yörüngeler izlettirilmiş ve performans değerleri gözlemlenmiştir. Verilen yörüngeler takip edilirken bozucu momentler uygulanmış ve yörüngelerde kayda değer bir bozulma meydana gelmemiş ve sistemin dayanıklı olduğu gözlemlenmiştir. Ayrıca üyelik fonksiyonların sayısı iki kat artırılarak yapılan simülasyonda yörünge takibinin hassasiyetinin de arttığı gözlemlenmiştir.

Wai ve Chen [101], yüksek hassasiyetli konum izlemeyi başarmak için n-eklemlerli bir robot manipülatörünün kontrolünü yapacak dört katmanlı yapay sinir bulanık ağı bir gürbüz kontrolör (Robust Neural Fuzzy Network Control - RNFNC) tasarlamışlardır. Ayrıca, önerilen RNFNC yöntemin verimliliğini ve sağlamlığını doğrulamak için DC servo motorlarla sürülen 2 eklemlerli bir robot manipülatörü üzerinde bulanık mantık kontrolörü, hesaplamalı tork kontrolörü ve de PD kontrolörü dâhil edilerek uygulamalar yapıp deneysel sonuçlar ve sayısal simülasyonlar elde etmişlerdir. Diğer kontrolörlere göre RNFNC kontrolörün, verilen referans eklem yörüngelerini daha yakın takip edebildiğini verilen sonuçlarla göstermişlerdir. Wai ve Yang [102], DC servo motorlarla sürülen 2 eklemlerli bir robot manipülatörünün yüksek hassasiyette

konum izlemeyi başarabilmesi için uyarlamalı bulanık yapay sinir ağı (Adaptive Fuzzy Neural Network Control - AFNNC) bir kontrolör geliştirmişlerdir. Önerilen kontrolörün sağlamlığını ve verimliliğini doğrulamak ve de üstünlüğünü göstermek için robot üzerinde diğer PD kontrolör, Takagi-Sugeno-Kang tipi bulanık yapay sinir ağı kontrolör (T-FNNC), gürbüz yapay sinir bulanık ağı kontrolör (R-NFNC) ve de bulanık model tabanlı kontrolör (F-MBC) sistemleriyle karşılaştırmalar yapmışlardır. Simülasyon sonuçlarına göre AFNNC sisteminin eklem konum izleme cevapları, diğer kontrol sistemlerinden daha yakın bir şekilde belirtilen referans yörüngelerini takip edebildiği görülmektedir.

Günümüz kontrol sistemlerinde optimizasyona olan talebin artması, problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle kontrol parametrelerin ayarlanmasında sert (hard) optimizasyon teknikleri yerine, esnek programlama (soft computing) ve evrimsel algoritma (evolutionary algorithm) kullanımı ön plana çıkmıştır. Evrimsel algoritmalar, doğal genetik yapıyı modelleyerek çözüm arayan bir arama algoritmalarıdır. Evrimsel yaklaşımlara dayanan Genetik Algoritmalar (Genetic Algorithms (GA)), Parçacık Sürü Optimizasyon algoritması (Particle Swarm Optimisation – PSO) ve Diferansiyel Gelişim algoritması (Differential Evolution – DE) gibi son zamanlarda popüler olan popülasyon tabanlı en iyileme algoritmalarından birileridir.

Literatürde, yukarıda sayılan robot kolu kontrolünde kullanılan bulanık mantık, PID ve yapay sinir ağı (neural network) gibi kontrolörlere ait parametrelerin evrimsel algoritmalar ile ayarlanmasına yönelik birçok çalışma vardır. Kwok ve arkadaşları [103], 6 serbestlik dereceli bir Puma 560 robot kolu için PID kontrolörünün parametelerini optimize etmek için genetik algoritmaları kullanmışlardır. Puma robot kolu üzerinde GA ile parametreleri ayarlanan PID kontrolörü kullanılarak çeşitli referans yörüngeleri için simülasyonlar yapmışlardır. Elde edilen simülasyon sonuçları, geleneksel optimizasyon teknikleri ile karşılaştırılmıştır. Elde edilen sonuçlar, robot kolu kontrolü için GA tabanlı parametre ayarlama tekniğinin verimli ve etkili bir şekilde çalışabildiğini ve de genel bir en uygun ayarlama yaklaşımı olabileceğini göstermiştir. Homaifar ve arkadaşları [104], iki eklemlilik bir robot kolunun konum ve yörünge kontrolü için aşamalı karma tipi bir bulanık-PID

kontrolör tasarlamışlardır. Genetik algoritmayı kullanarak büyük ebatlı bir manevra için kontrolöre ait üyelik fonksiyonların parametrelerini optimize etmişlerdir. Tasarladıkları bu kontrolör üzerinde simülasyonlar yaparak kontrolörün kararlılığını ve yörünge takip performansının çok iyi olduğunu göstermişlerdir. Ahn ve Anh [105], iki eklemli pnömatik yapay kas manipülatörün (PAM) parametrelerini belirlemek için geleneksel GA'dan daha doğru bir aramayı ve daha hızlı bir yakınsamayı üretecek değiştirilmiş bir genetik algoritma önermişlerdir. Geleneksel GA ve en küçük karesel ortalama yöntemleriyle karşılaştırma yaparak elde ettikleri deneysel sonuçlar, PAM manipülatörünün modellenmesi ve tanımlanmasında önerilen algoritmanın mükemmel performans sergilediğini göstermiştir.

Wong ve arkadaşları [106], iki tekerlekli mobil bir robotun sağ ve sol tekerlek motor hızlarını belirlemek için mesafe bulanık kontrolörü ve açı bulanık kontrolöründen oluşan bir hareket kontrol yapısı önermişlerdir. Kontrol edilen robotun iki boyutlu uzayda arzu edilen konuma etkili bir şekilde hareket edebilmesi amacıyla bulanık kontrolörlerin uygun üyelik fonksiyonlarını belirlemek için PSO tabanlı bir yöntem sunmuşlardır. PSO tabanlı bulanık kontrolör tasarımıdaki yöntemde bulanık sistemlerin giriş ve çıkış üyelik fonksiyonları otomatik olarak seçilebilmektedir. Ayrıca, karşılaştırma yapmak amacıyla GA tabanlı yöntem de geliştirilmiştir. PSO tabanlı yöntem, GA tabanlı yöntemden daha iyi bir arama kabiliyetine sahip olduğunu ve de kontrol performansının daha iyi olduğunu göstermişlerdir. Bunun yanında uygulamalı olarak önerilen bulanık kontrolör tasarım yöntemini, FIRA Robot Soccer turnuvası için iki tekerlekli mobil robotların kontrolünde başarılı bir şekilde uygulamışlardır.

Bingul ve Karahan [107], iki serbestlik dereceli düzlemsel bir robotun kontrolü için PID kontrolör, bulanık mantık kontrolörü (FLC) ve kesir dereceli PID kontrolör (FOPID) tekniklerine ait kontrol parametrelerini PSO algoritmasını kullanarak optimize etmişlerdir. Daha iyi karşılaştırma yapmak amacıyla verilen yörünge için kontrolörlere ait parametreler, üç farklı uygunluk fonksiyonuna göre optimize edilmiştir. Kontrolörlerin sağlamlığını sınamak amacıyla verilen yörünge ve model parametreleri değiştirilmiş ve de sisteme gürültü eklenmiştir. Verilen yörünge simülasyon sonuçlarıyla, önerilen PSO tabanlı FOPID kontrolörünün diğer PSO

tabanlı kontrolörlerden daha iyi bir performans sergilediğini göstermişlerdir. Ayrıca, sağlamlık sınavından elde edilen sonuçlarla FOPID kontrolörün daha üstün olduğunu işaret etmişlerdir. Mazhari ve Kumar [108], PUMA 560 Robot kolu kontrolü için GA, GPS (Genelleştirilmiş Örüntülü Arama - Generalized Pattern Search), PSO, SA (Benzetimli Tavlama - Simulated Annealing), karma tipli GA-GPS ve GA-NM (Genetic Algorithm with Nelder-Mead) gibi farklı sezgisel arama algoritmalarını kullanarak Bulanık PD+I kontrolör tasarlayıp parametrelerini ayarlamışlardır. Yaptıkları karşılaştırmalarda, GA ve PSO algoritmaların başlangıç yakınsama hızının, yüksek olduğunu fakat optimum noktaya erişmek için iterasyon sayısının daha uzun olduğunu ve aynı şekilde GPS'nin başlangıç yakınsama hızının ise düşük fakat arama uzayına erişir erişmez çok hızlı olduğunu göstermişlerdir. Bu yaklaşımların hepsi için bozucu etkili ve etkisiz ayarlama performanslarını karşılaştırdıklarında GA-GPS karma tipli algoritmanın, diğerlerine göre en iyi performansa sahip olduğunu göstermişlerdir.

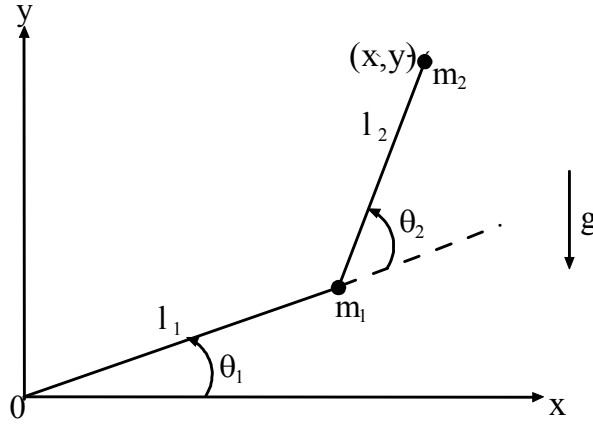
Bingul ve Karahan [109], iki serbestlik dereceli düzlemsel bir robotu kontrol etmek için 3 farklı maliyet fonksiyonlu PSO algoritmasını kullanarak Mamdani tipli bir bulanık mantık kontrolörü geliştirmişlerdir. Aynı zamanda bu optimizasyon tekniğini kullanarak PID kontrolör de tasarlamışlardır. Parameteleri ayarlanan kontrolörlerin sağlamlığını test etmek için sistemin model parametreleri ve verilen yörünge değiştirilmiş ve de sisteme gürültü eklenmiştir. Simülasyon sonuçlarında, robot kontrolü için PSO tabanlı bulanık mantık kontrolörünün PSO tabanlı PID kontrolöründen daha iyi ve daha sağlam olduğunu göstermişlerdir.

4.2. İki Serbestlik Dereceli Robotun Bulanık Mantık Yörünge Kontrolü

Bu bölümde yapılan çalışmada, iki serbestlik dereceli düzlemsel bir robot kolunun eklem uzayında bulanık mantık ve PID kontrol yöntemleri ile yörünge kontrolü yapılmıştır. Robot koluna ait dinamik denklemler verilmiştir. Kontrolörlere ait parametreler parçacık sürü optimizasyon algoritması kullanılarak ayarlanmıştır. Ayrıca kontrolörlerin performanslarını gözlemlemek amacıyla simülasyon çalışmaları ve sonuçlarına da yer verilmiştir.

4.2.1. Dinamik modeli

İki serbestlik dereceli robot kolunun dinamik modeli, yapının analizi ve kontrol algoritmasının tasarımı için önemlidir. Robot kolunun dinamik analizi, eklemlere, tahrik elemanları tarafından uygulanan moment veya kuvvet büyüklükleri ile robot kolunun zamana göre konumu, hızı ve ivmesi arasındaki ilişkilerin incelenmesidir. Robot manipülatörleri, doğru ve sağlam kontrolün zor yapılabildiği karmaşık doğrusal olmayan dinamik denklemlere sahiptir. Bu sebeple, burada tasarlanacak kontrolörlerin performansını test etmek için iyi bir örnektir.



Şekil 4.6. İki serbestlik dereceli robot kolu modeli

Şekil 4.6’da gösterilen iki serbestlik dereceli robotun dinamik denklemlerinin çıkartılmasında hem sistematik ve kolay olması hem de robot dinamik denklemlerinde çok yaygın kullanılması sebebiyle Lagrange-Euler (L-E) metodu kullanılmıştır. Doğrusal olmayan diferansiyel denklemlerin çözümden oluşan bir seri robot kolunun dinamik denklemleri, genel olarak aşağıda verildiği gibidir.

$$\tau = D(q)\ddot{q} + C(q, \dot{q}) + G(q) \quad (4.12)$$

Denklemden, birinci terim manipülatörün genel atalet tensörü veya kütle matrisini, ikinci terim Coriolis ve merkezkaç (centrifugal) kuvvet vektörünü ve üçüncü terim yerçekimi ivmesini temsil etmektedir. Denklem (4.12)’de verildiği gibi iki eklemlilik düzlemsel robotun her bir eyleyicisine etki eden tork ifadesi kütle matrisi, Coriolis ve merkezkaç kuvvet vektörü ve yerçekimi vektörünün toplanmasıyla aşağıdaki gibi bulunur.

$$\begin{aligned}
\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{pmatrix} (m_1 + m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2 \cos\theta_2 & m_2l_2^2 + m_2l_1l_2 \cos\theta_2 \\ m_2l_2^2 + m_2l_1l_2 \cos\theta_2 & m_2l_2^2 \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} \\
&+ \begin{pmatrix} -m_2l_1l_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2)\sin\theta_2 \\ m_2l_1l_2\dot{\theta}_1^2 \sin\theta_2 \end{pmatrix} + \begin{pmatrix} (m_1 + m_2)gl_1 \cos\theta_1 + m_2gl_2 \cos(\theta_1 + \theta_2) \\ m_2gl_2 \cos(\theta_1 + \theta_2) \end{pmatrix}
\end{aligned} \tag{4.13}$$

Denklemdede, m_i bağın kütesini, l_i bağ uzunluğunu, g yerçekimi ivmesini, $\theta, \dot{\theta}$ ve $\ddot{\theta}$ sırasıyla eklem konumunu, hızını ve ivmesini göstermektedir.

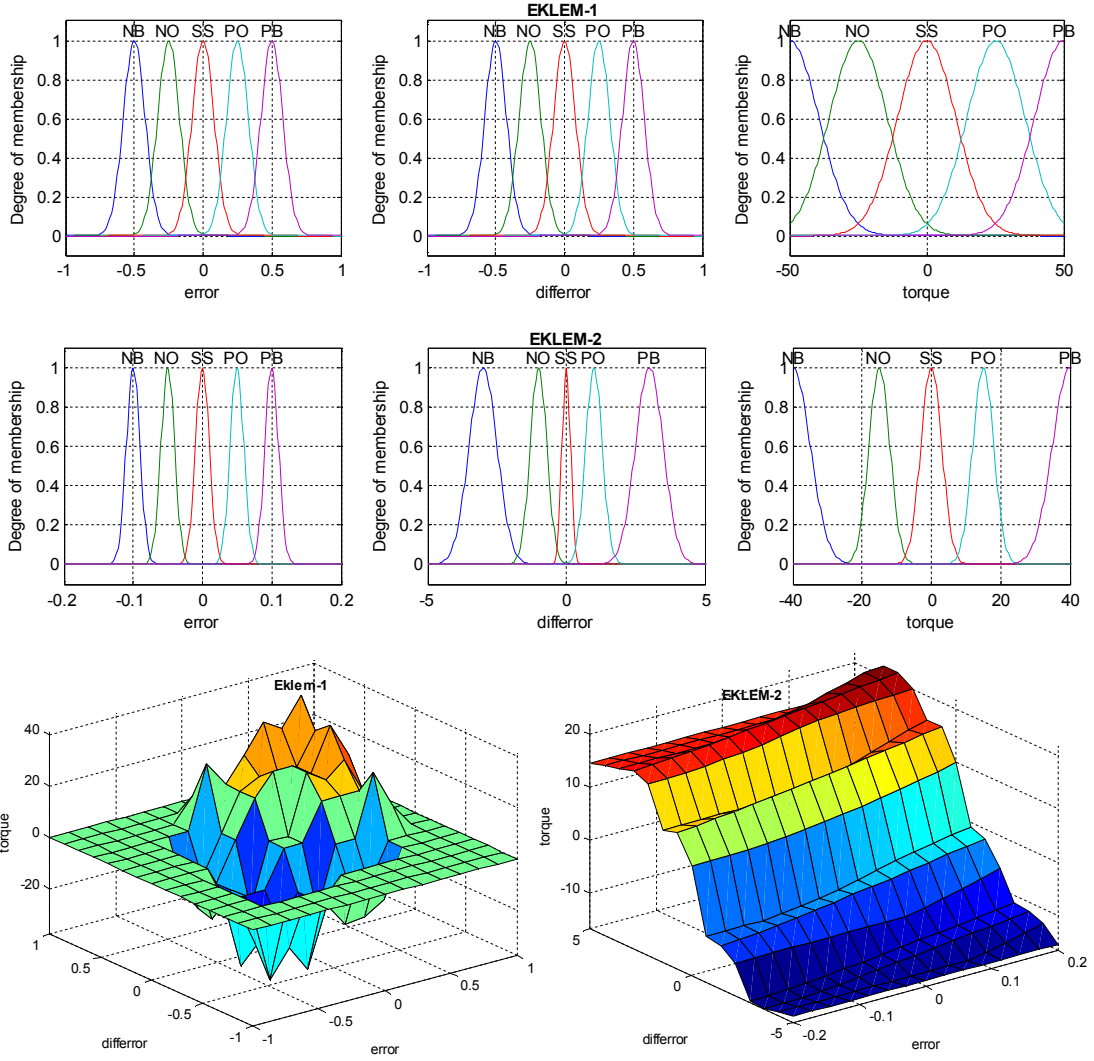
4.2.2. Bulanık mantık kontrolörü

Burada, robot yörünge kontrolünde yapılan bulanık mantık kontrolörünün giriş değişkenleri, eklem konumunun hatası (e) ve hata değişimi (\dot{e})'dir. Bulanık kontrolörünün çıkış değişkeni ise eklem kontrol sinyali olan tork (u -torque)'tur. e ve \dot{e} , bulanıklaştırıcı tarafından "error" ve "differror" dilsel değişkenler olarak belirlenmiştir. Ayrıca, durulaştırma ve kuralların uygulanması sırasında her bir bulanık sayı grubunda beşer adet bulanık küme kullanılarak bulanık değişkenler dilsel olarak temsil edilmişlerdir. Bu dilsel değişkenler sırasıyla Negatif Büyük (NB), Negatif Orta (NO), Sıfır (SS), Pozitif Orta (PO) ve Pozitif Büyük (PB) olarak tanımlanmıştır. Kullanılan bu dilsel değişkenler, giriş ve çıkış uzayının bulanık kısmını oluşturmaktadır. Şekil 4.7, eklem konum hataları, konum hatasındaki değişimler ve kontrol sinyalleri için bulanık kümelerin başlangıç üyelik fonksiyonlarını ve de bulanık kontrol yüzeylerini göstermektedir.

Burada kullanılan bu sözel değişkenler için Şekil 4.7'de gösterildiği gibi Gaussian biçimli üyelik fonksiyonları kullanılmıştır. Gaussian üyelik fonksiyonu aşağıdaki gibi tanımlanır.

$$\mu(x) = \exp\left[\frac{-(x-c)^2}{\sigma^2}\right] \tag{4.14}$$

Denklemdede c ve σ sırasıyla Gaussian üyelik fonksiyonunun merkez değeri ve standart sapmasıdır.



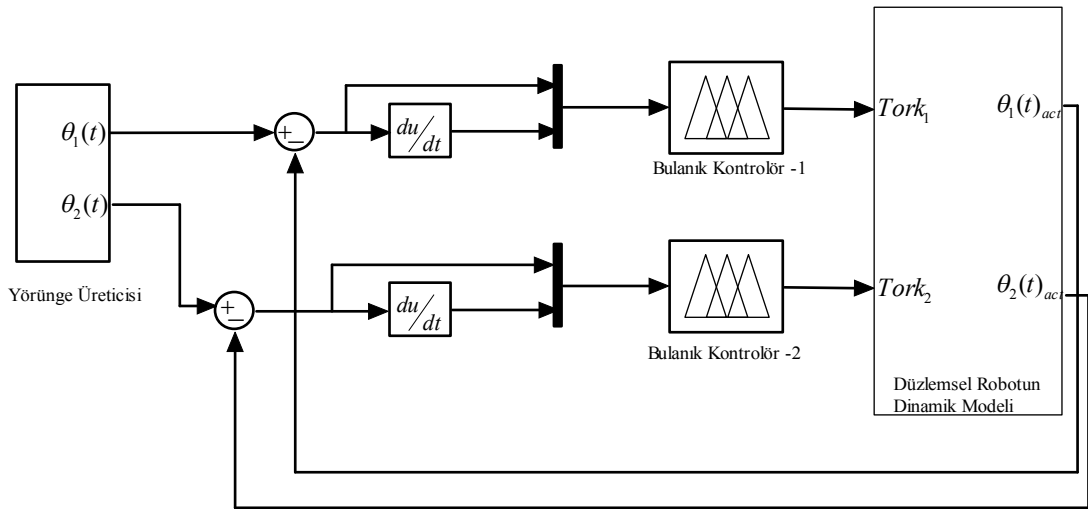
Şekil 4.7. Bulanık kümelerin başlangıç üyelik fonksiyonları ve kontrol yüzeyleri

Bulanık işlemlerin yapıp bulanık kararların verildiği birimin temel elemanları bulanık kurallardır. Bu bulanık kurallar, eldeki bulanık verilerden yeni bulanık sonuçlara varılmasını sağlarlar. Burada, robot yörünge kontrolü için kullanılan bulanık kurallar (IF-THEN) Tablo 4.1’de verilmektedir. Tablodan anlaşılacağı üzere toplam kural sayısı 25’dir.

Tablo 4.1. Bulanık mantık kural atama tablosu

		Hatanın Değişimi (differror)				
		NB	NO	SS	PO	PB
Hata (error)	NB	NB	NB	NO	NO	SS
	NO	NB	NO	NO	SS	PO
	SS	NO	NO	SS	PO	PO
	PO	NO	SS	PO	PO	PB
	PB	SS	PO	PO	PB	PB

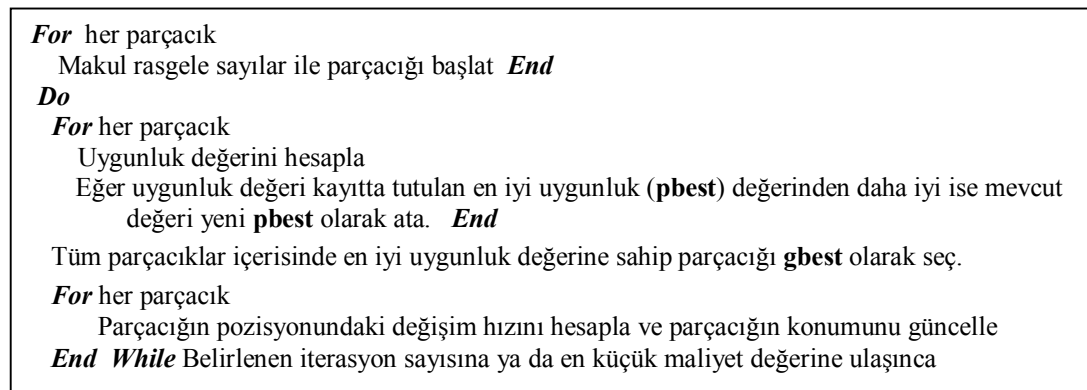
Bütün kurallar işlenip, her bir kurala ilişkin bulanık sonuç elde edildikten sonra, bu sonuçlar durulaştırılarak istenilen gerçek sonuç belirlenir. Durulaştırma işlemi, bulanık kontrolörünün durulaştırma biriminde gerçekleştirilir. Bulanık kontrol algoritmasının son aşaması olan durulaştırıcıda bilinen durulaştırma yöntemlerinden uygun olan biri kullanılarak bu işlem yapılır. Bu çalışmada alanların merkezi yöntemi kullanılarak kontrolörün çıkış sinyalindeki değişim elde edilmiştir. Robot kolu için bulanık mantık kontrolörünün simülink modeli Şekil 4.8'de gösterilmektedir.



Şekil 4.8. Robot kolu için bulanık mantık kontrolörün simülink modeli

4.2.3. PSO ile bulanık mantık kontrolörün ayarlanması

Robot kolu eklemlerinde yer alan her bir bulanık kontrolör 15 üyelik fonksiyonuna ve 25 kurala sahip olduğundan dolayı bu çalışmada toplam 60 parametre optimize edilecektir.



Şekil 4.9. PSO için pseudo kodu

PSO algoritması, 60 boyutlu uzayda giriş ve çıkış parametrelerinin tümünü arar. Böylece, tüm eklem için kontrolörün parametreleri Şekil 4.9’da gösterildiği gibi temel kod yapılı PSO’yu kullanarak optimize edilir. PSO algoritmasında kullanılan parçacığın yapısı aşağıda gösterilen denklem ile ifade edilebilir.

$$P_i = \begin{bmatrix} \sigma_{11} c_{11} & \dots & \sigma_{15} c_{15} & \sigma_{21} c_{21} & \dots & \sigma_{25} c_{25} & \sigma_{31} c_{31} & \dots & \sigma_{35} c_{35} \\ \sigma_{41} c_{41} & \dots & \sigma_{45} c_{45} & \sigma_{51} c_{51} & \dots & \sigma_{55} c_{55} & \sigma_{61} c_{61} & \dots & \sigma_{65} c_{65} \end{bmatrix} \quad (4.15)$$

Denklemde c_{ij} ve σ_{ij} parametreleri, Gaussian üyelik fonksiyonların merkezini ve genişliğini temsil ederler. Yukarıdaki denklemde, birinci ve ikinci satırlar, sırasıyla birinci ve ikinci bulanık kontrolörlerin parametrelerini oluştururlar. Parçacıklara ait bu parametrelerin başlangıç değerleri algoritmanın birinci adımı için rastgele üretilir.

PSO algoritmasını uygularken en önemli adım, her bir parçacığın uygunluğunu değerlendirmek için kullanılan en iyi maliyet fonksiyonunun seçilmesidir. Bu çalışmada kontrolörlerin parametrelerini PSO algoritması ile ayarlama sürecinde 3 farklı maliyet fonksiyonu kullanılmıştır. Aşağıdaki denklemlerde verildiği gibi Mean of Root of Squared Error (MRSE), Mean of Absolute Magnitude of the Error (MAE) ve Mean Minimum Fuel and Absolute error (MMFAE)’dur. Parçacık (P_i) için maliyet fonksiyonları aşağıdakiler gibi hesaplanır.

$$E_{MRSE}(k) = \frac{1}{N} \sum_{i=1}^N \sqrt{e_1^2(i) + e_2^2(i)} \quad (4.16)$$

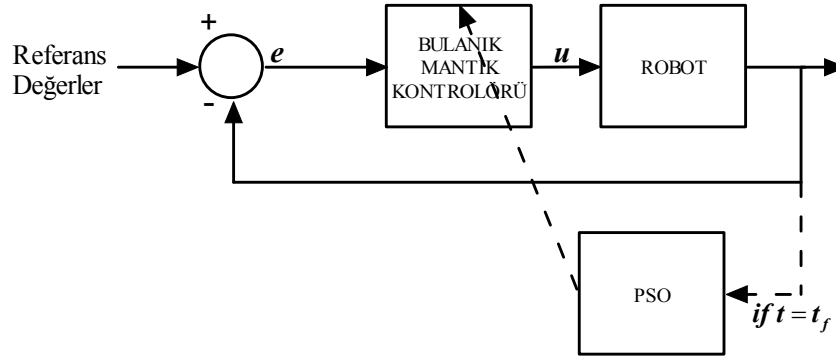
$$E_{MAE}(k) = \frac{1}{N} \sum_{i=1}^N |e_1(i)| + |e_2(i)| \quad (4.17)$$

$$E_{MMFAE}(k) = \frac{1}{N} \sum_{i=1}^N |e_1(i)| + |e_2(i)| + |u_1(i)| + |u_2(i)| \quad (4.18)$$

Denklemde $e_1(i)$ birinci eklem için i .ci örneğin yörünge hatası, $e_2(i)$ ikinci eklem için i . örneğin yörünge hatası, N , örnek sayısı, k , adım sayısı $u_1(i)$ ve $u_2(i)$ sırasıyla birinci ve ikinci eklem için kontrol sinyalleridir. Aşağıdaki denklemde, kontrolörün her bir eklemde ürettiği kontrol sinyallerinin RMS değerlerinin ortalaması verilmektedir.

$$U_{MRMS} = \frac{\sqrt{\frac{1}{N} \sum_i^N (u_1(i))^2} + \sqrt{\frac{1}{N} \sum_i^N (u_2(i))^2}}{2} \quad (4.19)$$

Şekil 4.10, PSO algoritmasını kullanarak bulanık mantık kontrolörünü optimize etme sürecinin blok yapısını göstermektedir. Bulanık mantık kontrolörünün tüm parametreleri her simülasyon zamanı bitiminde (t_f) güncellenir.



Şekil 4.10. PSO ile bulanık kontrol parametrelerinin ayarlanma süreci

Burada, hesapsal yükü göz önünde bulundurarak popülasyon büyüklüğü 10 parçacığa göre ayarlanmıştır. Dolayısıyla, her bir parçacık P_i , 60 eleman içerdiğinden sürünün büyüklüğü 10x60'tır. Şekil 4.9'da verilen koda göre Şekil 4.10'da gösterilen PSO bloğunda, her bir parçacık için maliyet fonksiyonu hesaplanır ve her simülasyon zamanı sonunda (t_f) pbest ve gbest değerleri elde edilir. Her bir parçacık için parçacığın hızı hesaplandıktan sonra parçacığın konumu güncellenir. Bu algoritma, verilen maksimum adım sayısına ya da minimum maliyet ölçüt değerine ulaşıncaya kadar çalıştırılır.

Algoritmada yer alan χ , φ_1 ve φ_2 sırasıyla 0.76, 2.05 ve 2.05 olarak ayarlanmıştır [109]. Parçacıktaki elemanların değerleri makul aralığı aşmamaları için üyelik fonksiyonların merkezleri (c_{ij}) ve genişlikleri (σ_{ij}) sırasıyla $[c_{\min}, c_{\max}]$ ve $[\sigma_{\min}, \sigma_{\max}]$ aralıkları dâhilinde sınırlandırılmıştır. Tablo 4.2'de her bir bulanık kontrolördeki girişlerde (error – differror) ve çıkış (torque)'ta bulunan üyelik fonksiyonlarına ait parametrelerin sınır aralık değerleri verilmektedir.

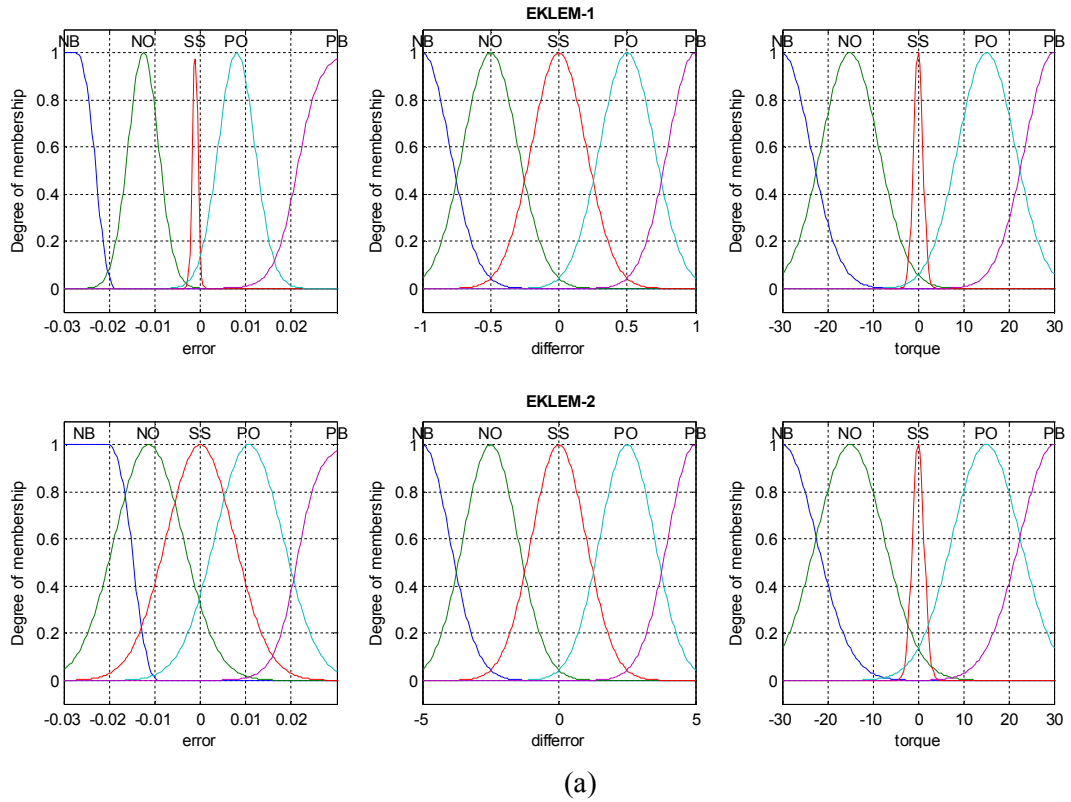
Tablo 4.2. Üyelik fonksiyon parametrelerinin sınır değerleri

Parametre	error		differror		torque	
	Kontrol-1	Kontrol-2	Kontrol-1	Kontrol-2	Kontrol-1	Kontrol-2
$[c_{\min}, c_{\max}]$	[-1, 1]	[-0.19, 0.19]	[-1, 1]	[-5, 5]	[-50, 50]	[-40, 40]
$[\sigma_{\min}, \sigma_{\max}]$	[0.02, 0.1]	[0.006, 0.014]	[0.02, 0.1]	[0.2, 0.6]	[1, 12]	[2, 10]

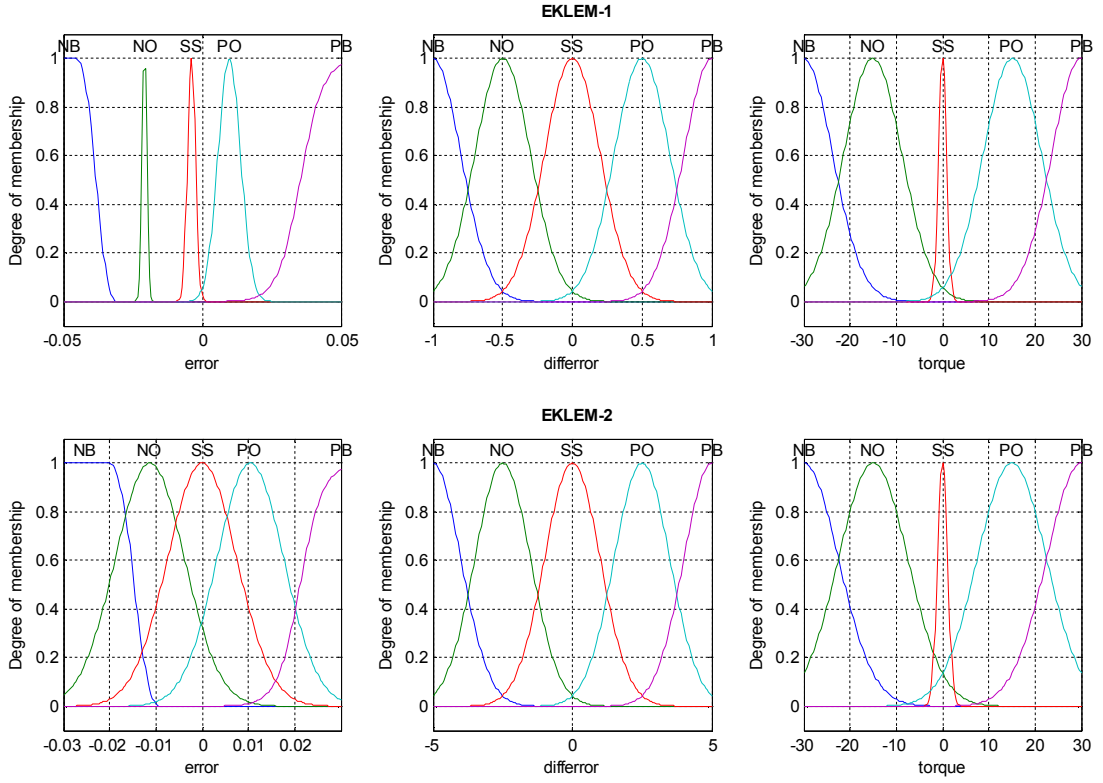
Aşağıda verilen yörünge denklemine göre, PSO algoritması, 3 farklı maliyet fonksiyonundan birisi ile her bir adımda 10 parçacık çalıştırmak suretiyle toplam 60 adım boyunca çalıştırılmıştır. Tüm simülasyonlar, Matlab 7.0.1'de yazılan bir program kullanılarak yerine getirilmiştir.

$$\theta_1(t) = 0.1524 + 0.24384 \cos\left(\frac{2\pi t}{5} - \frac{\pi}{2}\right) \quad (4.20)$$

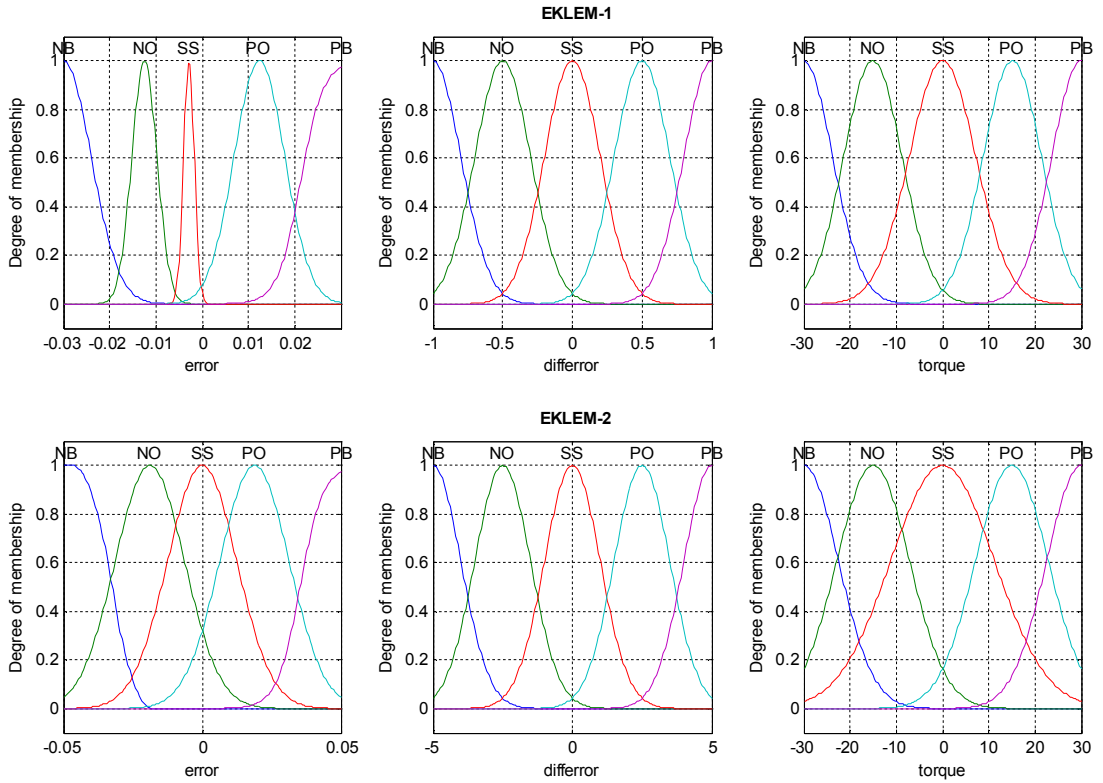
$$\theta_2(t) = 0.39624 + 0.24384 \sin\left(\frac{2\pi t}{5} - \frac{\pi}{2}\right) \quad (4.21)$$



Şekil 4.11. MRSE (a), MAE (b) ve MMFAE (c) ile optimizasyon sonrası üyelik fonksiyonları



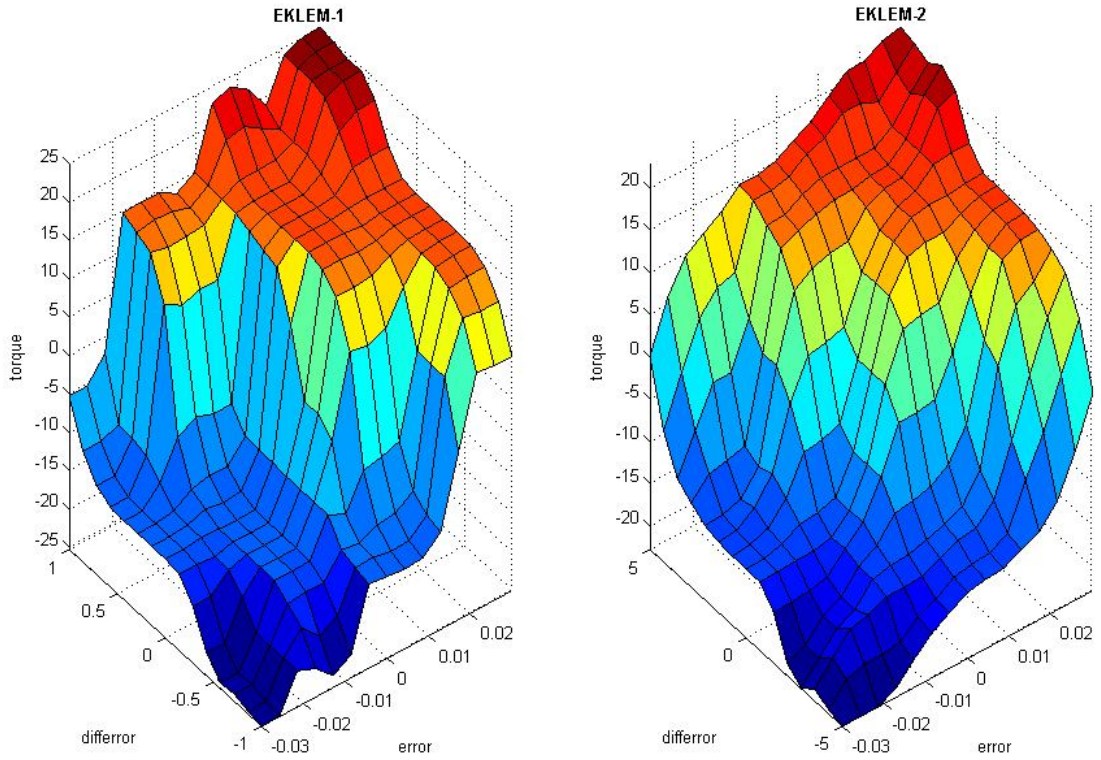
(b)



(c)

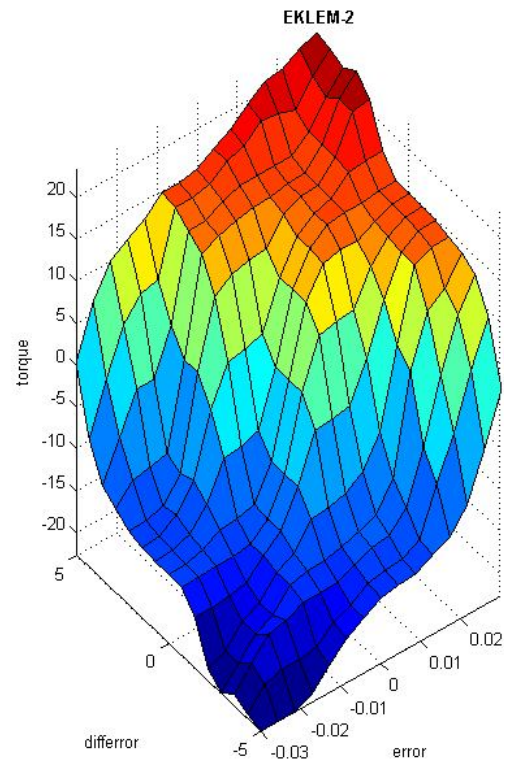
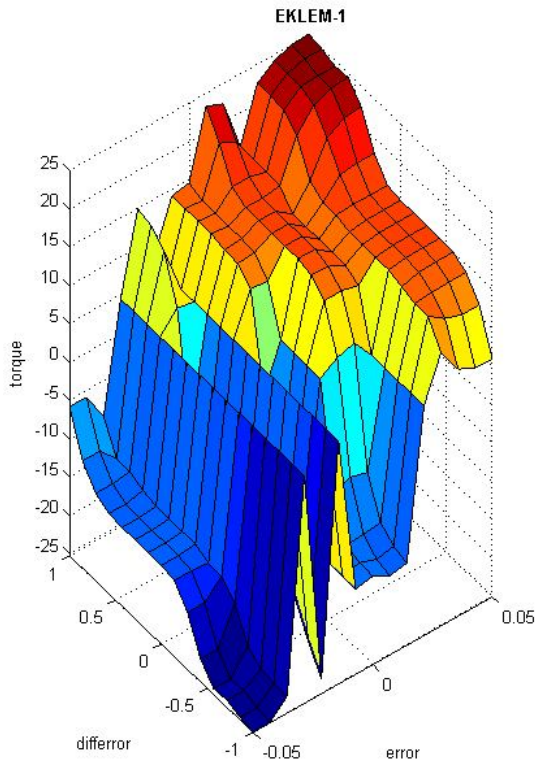
Şekil 4.11. (Devam) MRSE (a), MAE (b) ve MMFAE (c) ile optimizasyon sonrası üyelik fonksiyonları

MRSE, MAE ve MMFAE maliyet fonksiyonlarının her biri ile ayarlanan bulanık mantık kontrolörünün üyelik fonksiyonları ve kontrol yüzeyleri her iki eklem için sırasıyla Şekil 4.11 ve Şekil 4.12’de gösterilmektedir. Şekil 4.11.a, b, c’de görülebildiği gibi verilen robot yörüngesine dayanarak PSO ile ayarlanan kontrolörlerin başlangıç üyelik fonksiyonları değişmiştir. Optimizasyon süreci sonunda, değişimler özetlenirse; error girişine ait NB (Negatif Büyük) ve PB (Pozitif Büyük) dilsel değişkenli üyelik fonksiyonları, spline ve sigmoidal fonksiyon olarak değişmiştir. Ayrıca, konum hatalarına (error) ve kontrol girişlerine (torque) ait tüm maliyet fonksiyonlarında optimize edilmiş üyelik fonksiyonların bulunduğu uzayın sınırları azalmıştır. MRSE ve MAE maliyet fonksiyonlarından elde edilen sonuçlar karşılaştırıldığında her iki eklemdaki kontrolörlere ait üyelik fonksiyonların merkez ve genişlik parametrelerinin hemen hemen aynı değerlere sahip olduğu görülmektedir. Diğer taraftan, MMFAE kullanılarak çalıştırılan PSO algoritması, üyelik fonksiyonların farklı değerleri üretmiştir.

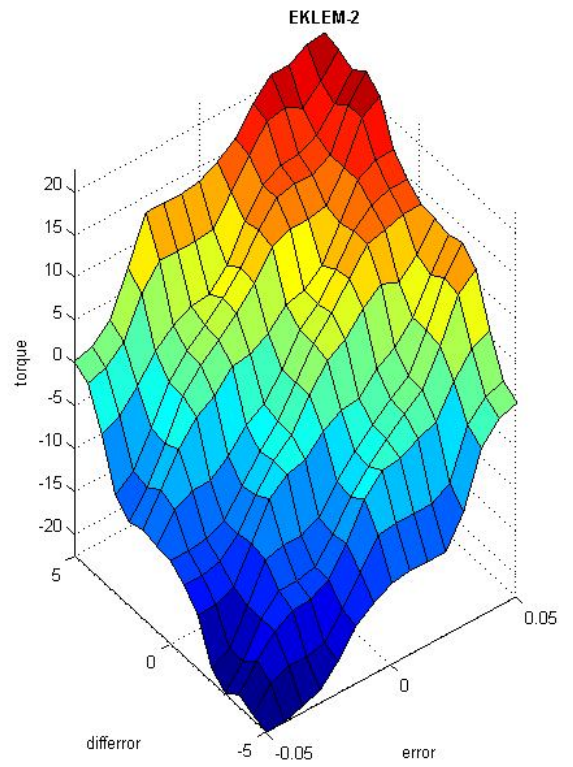
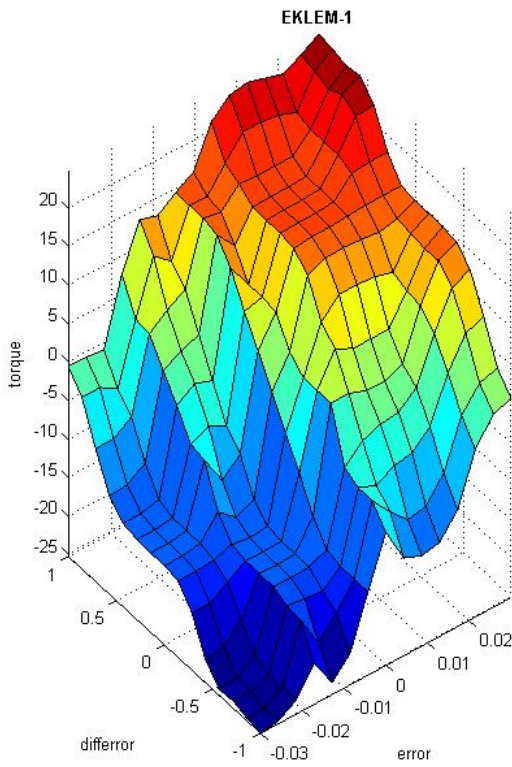


(a)

Şekil 4.12. MRSE (a), MAE (b) ve MMFAE (c) ile optimizasyon sonrası bulanık kontrol yüzeyleri



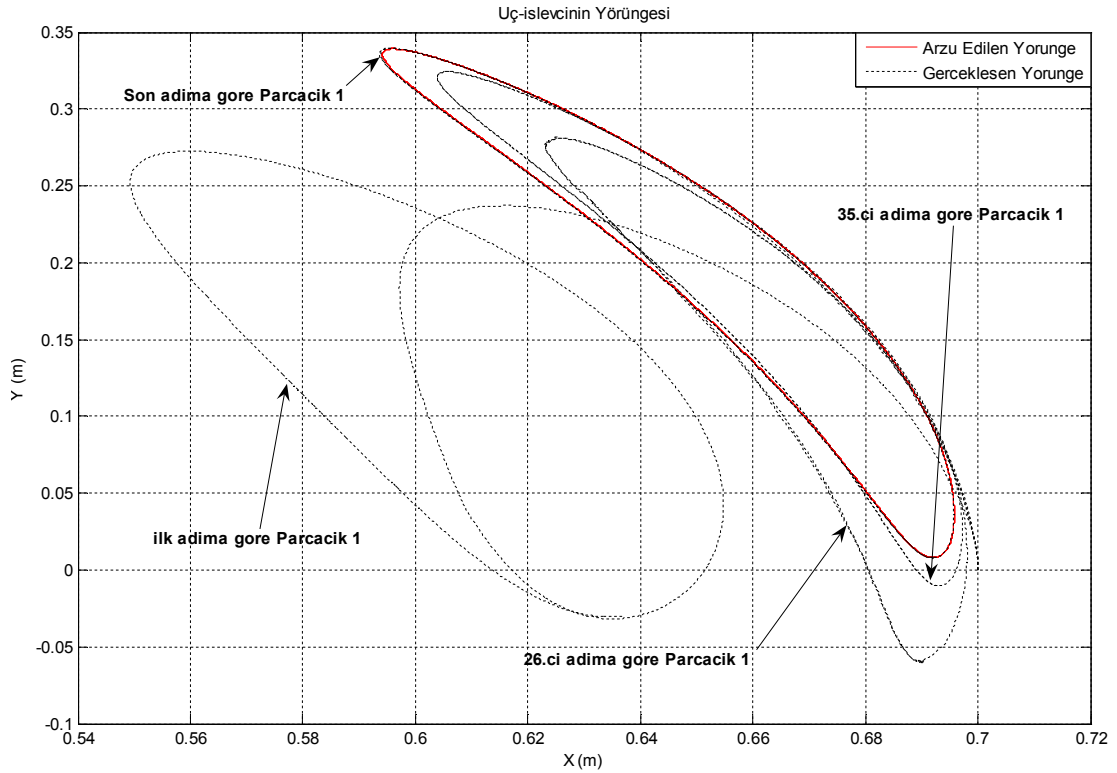
(b)



(c)

Şekil 4.12. (Devam) MRSE (a), MAE (b) ve MMFAE (c) ile optimizasyon sonrası bulanık kontrol yüzeyleri

Şekil 4.12’de farklı maliyet fonksiyonu ile ayarlanan kontrol yüzeyleri karşılaştırıldığında her birinin farklı olduğu görülmektedir. Eklem 2’ye bakıldığından ayarlı yüzeyler arasında farklılıkların küçük olduğu fakat Eklem 1’de ise büyük olduğu anlaşılmaktadır. Şekil 4.13, verilen yörünge boyunca birinci adımdan son adıma kadar sürü içindeki birinci parçacığın (P_1) hareketini göstermektedir.



Şekil 4.13. Optimizasyon süreci boyunca 1 nolu Parçacığın (P_1) hareketi

4.2.4. PSO ile PID kontrolörün ayarlanması

PSO tabanlı bulanık mantık kontrolörün (Bulanık-PSO) performansını göstermek amacıyla 3 farklı maliyet fonksiyonunu kullanarak PSO ile optimizasyon işlemi PID kontrolör için de uygulanmıştır. Böylece PSO tabanlı PID kontrolör (PID-PSO) ile Bulanık-PSO arasındaki farklılıklar gözlemlenebilmiştir. Tam bir karşılaştırma yapabilmek amacıyla hem bulanık mantık kontrolörü hem de PID kontrolör için optimizasyon şartları aynı olması sağlanmıştır. Bu sebeple, optimizasyon işleminde kontrol sinyalinin mümkün değer aralığı, üst sınır olarak Eklem 1 için 26 N.m, Eklem 2 için 20 N.m ve alt sınır olarak Eklem 1 için -14 N.m, Eklem 2 için ise -3 N.m olarak verilmiştir.

PSO algoritmasını kullanarak farklı maliyet fonksiyonları ile ayarlanan PID katsayıları, Tablo 4.3'te verilmektedir.

Tablo 4.3. Farklı maliyet fonksiyonları için PID katsayıları

PID Katsayıları	PSO-MRSE		PSO-MAE		PSO-MMFAE	
	Ekleml-1	Ekleml-2	Ekleml-1	Ekleml-2	Ekleml-1	Ekleml-2
K _p	973	1064	756	1526	358	465
K _i	591	26	586	96	259	220
K _d	30	9	25	15	84	11

Tabloya bakıldığında ayarlanan PID katsayılarının farklı maliyet fonksiyonlarında ve eklemde farklı değerlere sahip olduğu görülmektedir. Eklem 1 için MRSE ve MAE maliyet fonksiyonlarına göre ayarlanan katsayılar, hemen hemen aynı değerlere sahiptir. Eklem 2 için ise PID katsayıları arasındaki farklılıkların büyük olduğu ve oransal katsayının diğerlerinden daha yüksek çıktığı görülmektedir.

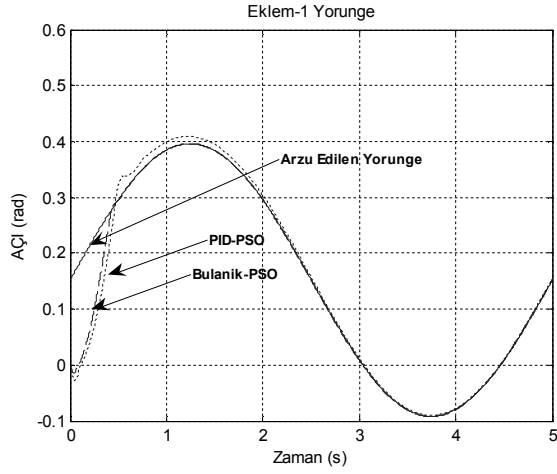
4.2.5. Bulanık ve PID kontrol için deneysel sonuçlar

Bu kısımda, Bulanık-PSO ve PID-PSO kontrolörlerin sağlamlığını test etmek ve bu kontrolörlerin performansını göstermek amacıyla simülasyon çalışmalarına yer verilecektir. İlk olarak, herhangi bir bozucu etki olmaksızın kontrolörler için elde edilen maliyet değerleri ve Denklem (4.19)'a göre hesaplanan ortalama RMS değerleri Tablo 4.4'te verilmektedir. Tabloda verilen maliyet değerlerine göre PID-PSO kontrolörünün tüm maliyet fonksiyonlarında Bulanık-PSO kontrolöründen daha iyi olduğu görülmektedir. Ortalama RMS değerlerine bakıldığında Bulanık-PSO kontrolörün daha düşük kontrol sinyali ürettiği anlaşılmaktadır.

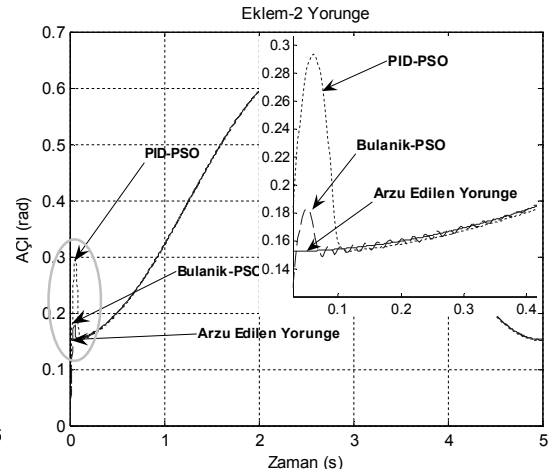
Tablo 4.4. Herhangi bir bozucu etki olmaksızın kontrolörlerin yapmış olduğu maliyet ve ortalama RMS değerleri

Kontrolör	MRSE	U_{MRMS}	MAE	U_{MRMS}	MMFAE	U_{MRMS}
PID-PSO	0.0029	4.1478	0.0032	4.0967	7.2339	3.7957
Bulanık-PSO	0.0041	3.9049	0.0040	3.9254	7.2920	3.8564

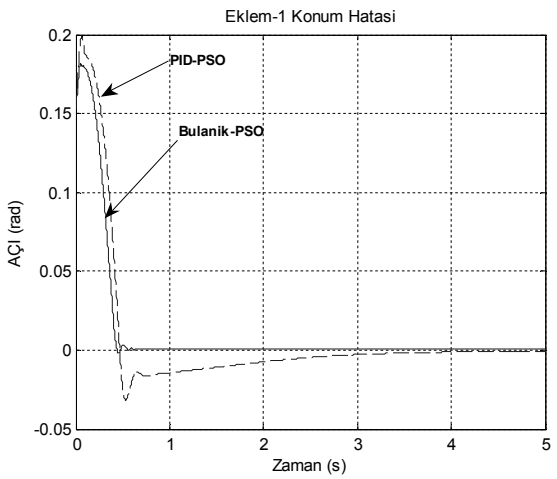
PSO tabanlı kontrolörlerin performanslarını karşılaştırmak ve sağlamlığını test etmek için ilk olarak, her bir eklemdeki robot kolunun kütlesi 3 katına çıkartılır. Şekil 4.14, her eklem için arzu edilen konumu, gerçekleşen konumu, konum hatasını ve kontrol sinyalini gösterir.



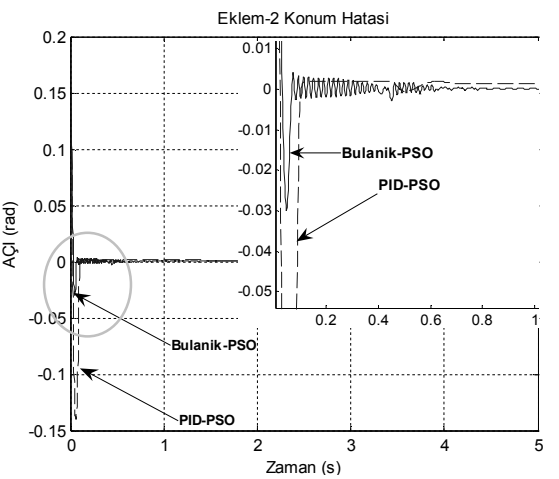
(a)



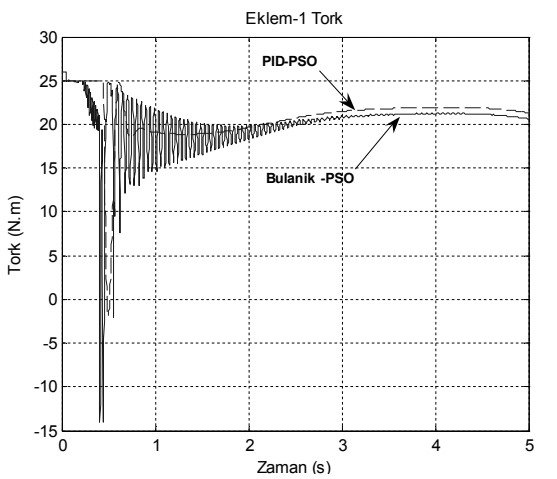
(d)



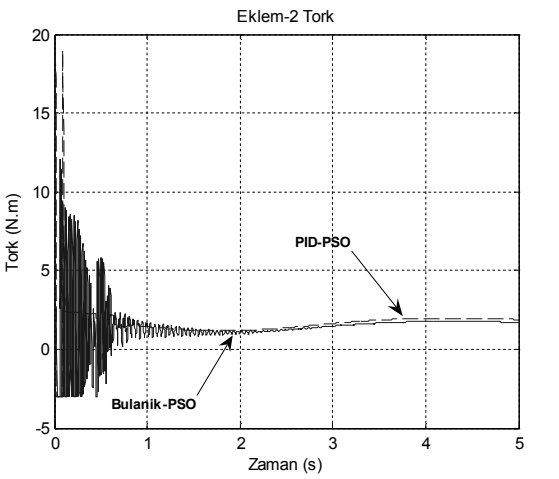
(b)



(e)



(c)

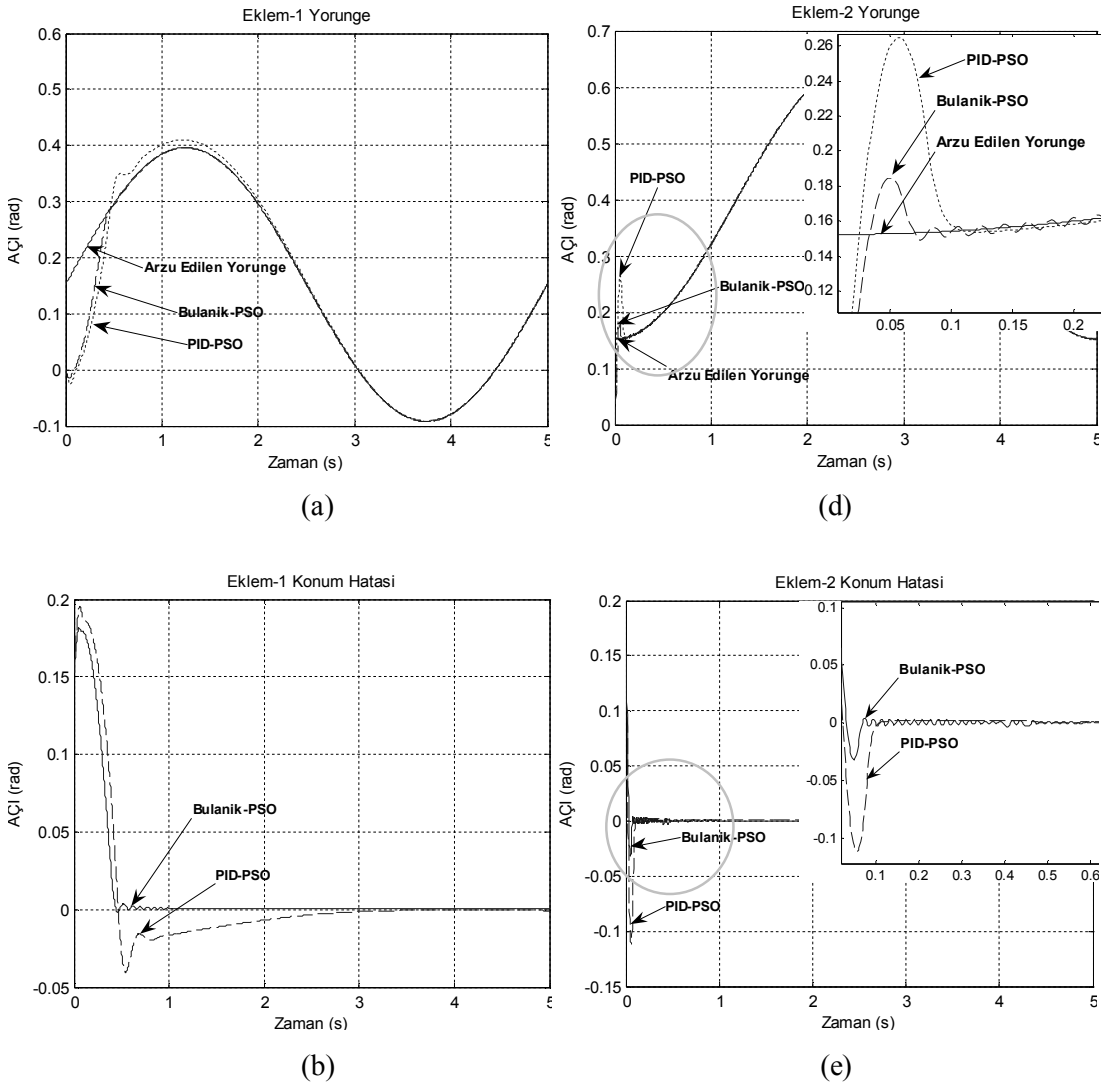


(f)

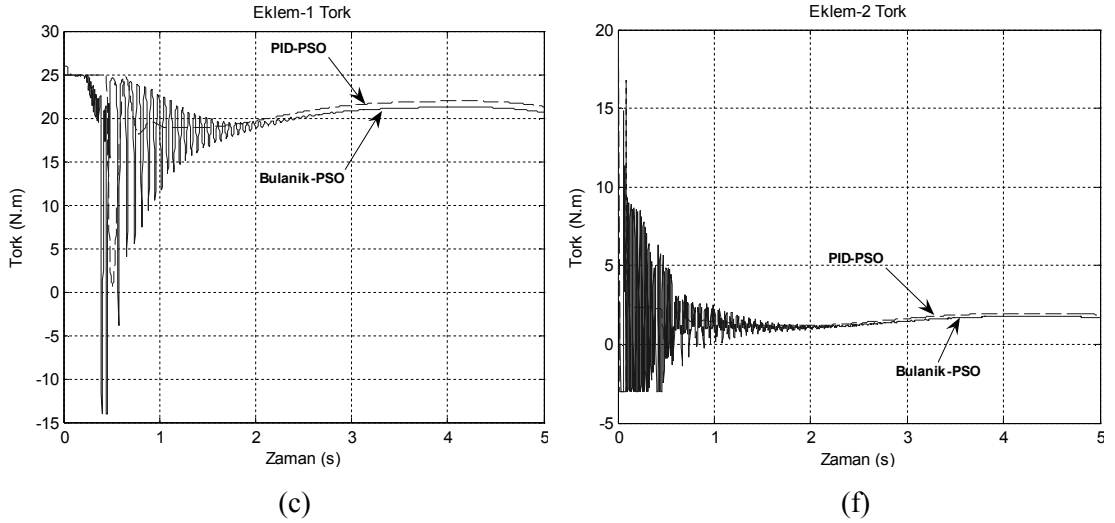
Şekil 4.14. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için kütle değişiklikleri altında MRSE maliyet fonksiyonlu Bulanik-PSO ve PID-PSO kontrolörlerin performansı

Şekil 4.14 incelendiğinde, aşım açısından Bulanık-PSO kontrolörü, PID-PSO kontrolöründen daha iyi performans gösterdiği görülmektedir. Öte yandan, Bulanık-PSO kontrolörü, yörünge üzerinde dalgalanmalar üretmiştir. Bu dalgalanmalar gittikçe azaldığı ve çok ufak olduğundan dolayı bunlar göz ardı edilebilir.

Şekil 4.15, MAE maliyet fonksiyonu kullanılarak optimize edilen kontrolörlere göre elde edilen sonuçları gösterir. Şekil incelendiğinde Bulanık-PSO kontrolörün performansı, PID-PSO kontrolörü ile karşılaştırıldığında daha iyi olduğu görülmektedir. Ayrıca, Şekil 4.15'te görülebildiği gibi, Bulanık-PSO kontrolörü, daha kısa oturma zamanına ve hızlı cevap zamanına sahiptir.

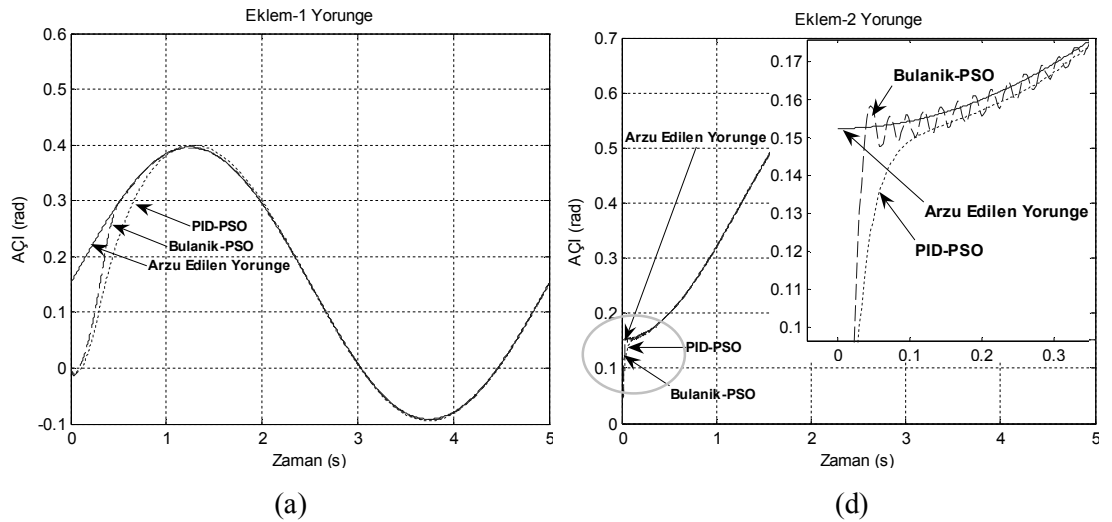


Şekil 4.15. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için kütle değişiklikleri altında MAE maliyet fonksiyonlu Bulanık-PSO ve PID-PSO kontrolörlerin performansı

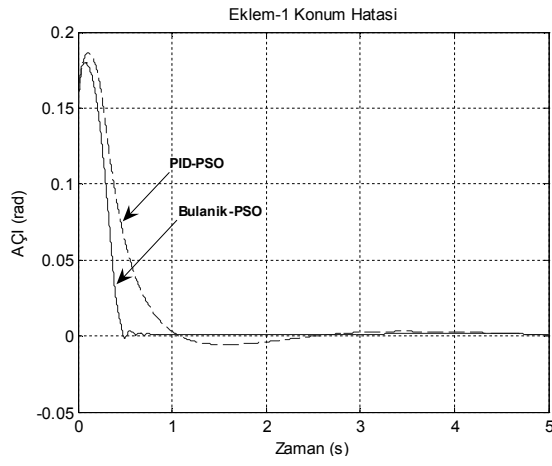


Şekil 4.15. (Devam) Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için kütle değişiklikleri altında MAE maliyet fonksiyonlu Bulanik-PSO ve PID-PSO kontrolörlerin performansı

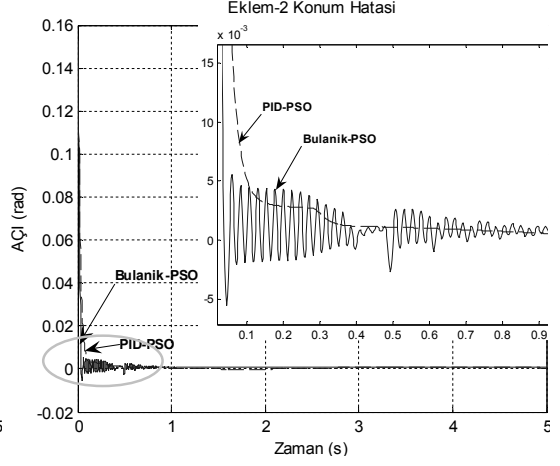
Şekil 4.16, MMFAE maliyet fonksiyonuna göre optimize edilen kontrolörlerden elde edilen simülasyon sonuçlarını gösterir. Şekil 4.16'da görülebildiği gibi Bulanik-PSO kontrolörün cevap zamanı ve yakınsama eğilimi, PID-PSO kontrolörün cevap zamanı ve yakınsama eğiliminden daha hızlıdır.



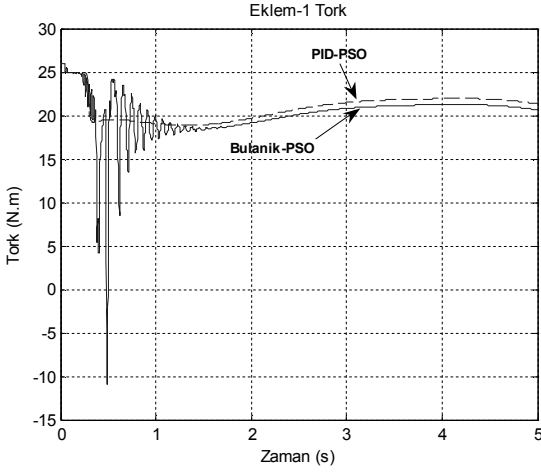
Şekil 4.16. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için kütle değişiklikleri altında MMFAE maliyet fonksiyonlu Bulanik-PSO ve PID-PSO kontrolörlerin performansı



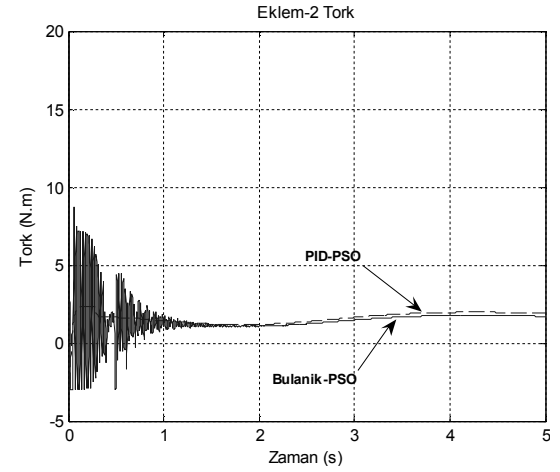
(b)



(e)



(c)



(f)

Şekil 4.16. (Devam) Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için kütle değişiklikleri altında MMFAE maliyet fonksiyonlu Bulanik-PSO ve PID-PSO kontrolörlerin performansı

Tablo 4.5'te, kütle değişikliği altında kontrolörlerde meydana gelen hata değerlerine göre hesaplanan maliyet fonksiyon değerleri ve ortalama RMS değerleri verilmektedir. Tablodan da anlaşılacağı üzere bu iç bozucu etki altında, hemen hemen aynı kontrol sinyallerin üretilmesine rağmen Bulanik-PSO kontrolörü, PID-PSO kontrolöründen daha iyi sonuçlar vermiştir.

Tablo 4.5. Kütle değişikliği altında kontrolörlerin maliyet ve ortalama RMS değerleri

Kontrolör	MRSE	U_{MRMS}	MAE	U_{MRMS}	MMFAE	U_{MRMS}
PID-PSO	0.0194	11.5833	0.0211	11.5466	22.5424	11.3762
Bulanik-PSO	0.0115	11.2755	0.0124	11.3255	21.7528	11.1573

Kontrolörlerin sağlamlığını değerlendirmek için ikinci olarak, robot sistemine harici bozucu etki yapan farklı varyanslı gürültü eklenir. Bu gürültü sırasıyla birinci eklem sonra ikinci eklem ve son olarak her iki eklem uygulanır. Buna göre kontrolörlerin tüm farklı gürültü varyanslarında yapmış olduğu maliyet değerleri Tablo 4.6, 4.7 ve 4.8'de verilmektedir. Tablolar incelendiğinde gürültü varyansı arttığında verilen yörüngeden sapmalar meydana geldiği ve de Bulanık-PSO kontrolörün, PID-PSO kontrolöründen daha iyi sonuçlar ürettiği görülmektedir.

Tablo 4.6. Farklı varyanslı gürültü eklendiğinde MRSE fonksiyonuna göre maliyet değerleri

Gürültü Varyansı	Eklem 1		Eklem 2		Eklem 1 ve 2	
	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.01	0.0115	0.0183	0.0124	0.0188	0.0122	0.0183
0.1	0.0115	0.0174	0.0134	0.0191	0.013	0.0176
0.5	0.0127	0.0166	0.0167	0.0208	0.0178	0.0186
0.7	0.014	0.0169	0.0194	0.0223	0.0198	0.0205
0.8	0.0147	0.0172	0.0209	0.0232	0.02	0.0218

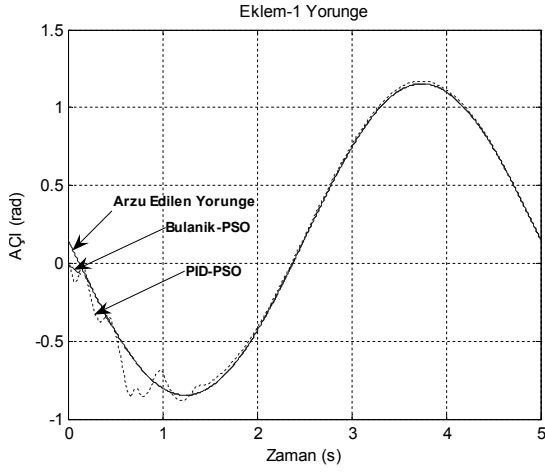
Tablo 4.7. Farklı varyanslı gürültü eklendiğinde MAE fonksiyonuna göre maliyet değerleri

Gürültü Varyansı	Eklem 1		Eklem 2		Eklem 1 ve 2	
	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.01	0.0124	0.0192	0.0134	0.0197	0.0134	0.0192
0.1	0.0129	0.0184	0.0146	0.0201	0.0144	0.0188
0.5	0.0146	0.018	0.0185	0.0227	0.0197	0.0206
0.7	0.0155	0.0184	0.0222	0.0248	0.0232	0.0232
0.8	0.0159	0.0187	0.0244	0.0261	0.0239	0.0248

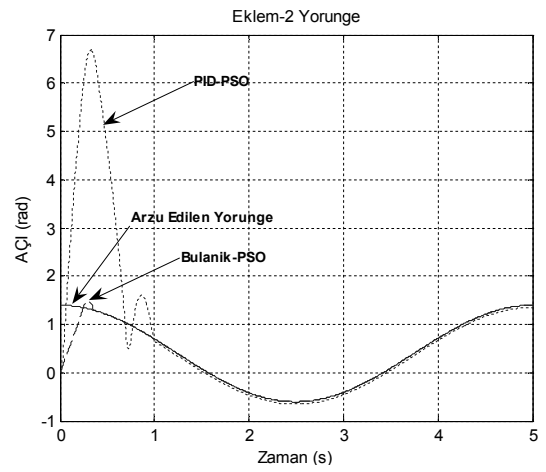
Tablo 4.8. Farklı varyanslı gürültü eklendiğinde MMFAE fonksiyonuna göre maliyet değerleri

Gürültü Varyansı	Eklem 1		Eklem 2		Eklem 1 ve 2	
	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.01	21.6959	22.4923	22.4423	22.6614	22.3042	22.5636
0.1	21.6143	22.3863	23.7178	24.7061	23.3599	24.1921
0.5	21.8345	22.2701	26.9391	29.2854	26.0566	28.0575
0.7	21.9327	22.2752	28.0276	30.7149	27.0011	29.321
0.8	21.972	22.2839	28.4338	31.2986	27.4521	29.8447

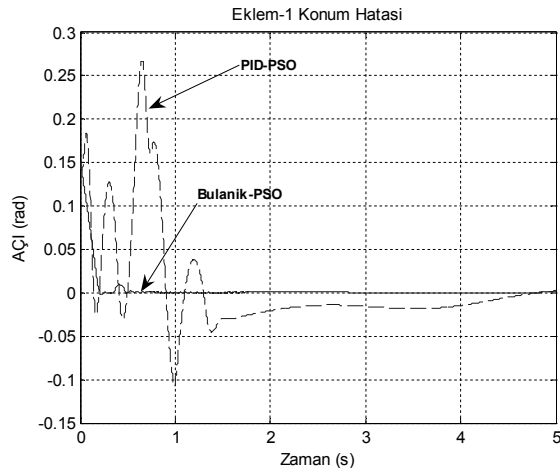
Üçüncü olarak yapılan sağlamlık testinde, verilen yörüngedeki genlik (A) ve faz (ϕ) değerlerinin birinci eklem, ikinci eklem ve her iki eklem için değiştirilmesidir.



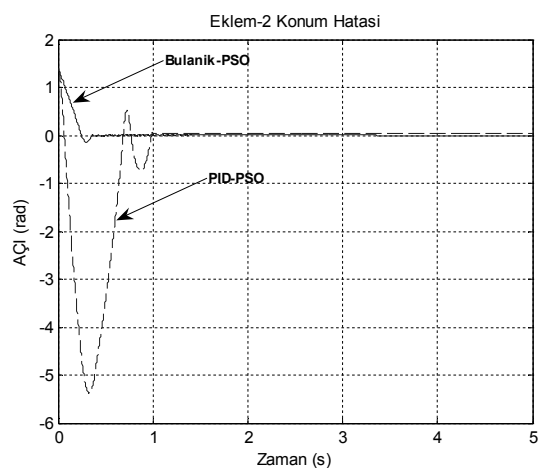
(a)



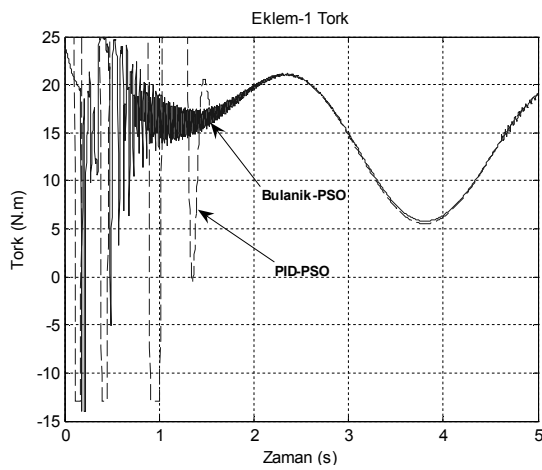
(d)



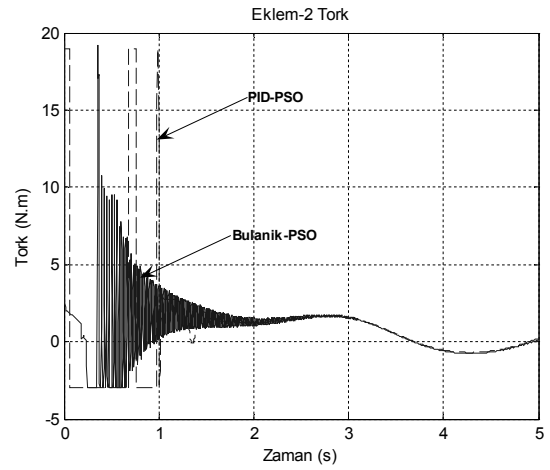
(b)



(e)



(c)

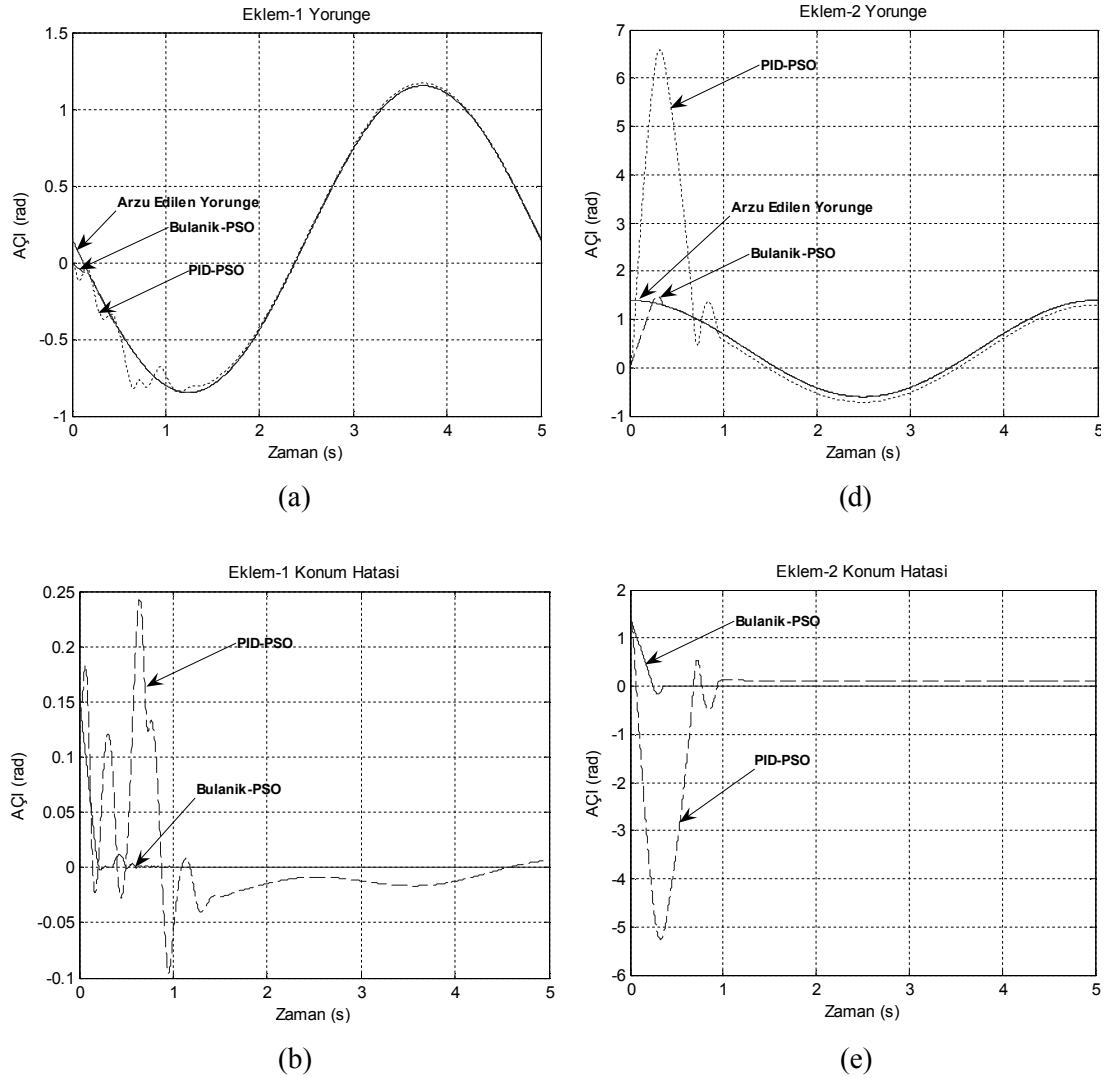


(f)

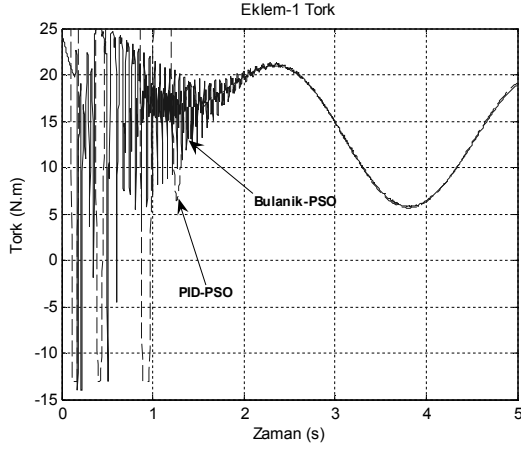
Şekil 4.17. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için hem genlik hem de faz değişiklikleri altında MRSE maliyet fonksiyonlu Bulanik-PSO ve PID-PSO kontrolörlerin performansı

Bu sağlamlık testinde, kontrol edilen eklemlerin arzu edilen ve gerçekleşen konumları, konum hataları ve kontrol sinyalleri Şekil 4.17’de gösterilmektedir. Şekil’de görülebildiği gibi MRSE maliyet fonksiyonlu Bulanık-PSO kontrolörün, aynı maliyet fonksiyonlu PID-PSO kontrolöründen daha etkili kalıcı durum cevabına ve hızlı bir yakınsama oranına sahiptir.

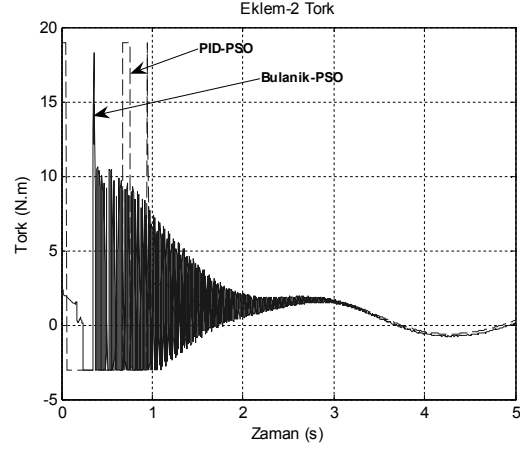
Şekil 4.18 ve Şekil 4.19, sırasıyla MAE ve MMFAE maliyet fonksiyonlarıyla ayarlanan kontrolörlere göre eklemlerin arzu edilen ve gerçekleşen konumlarını, konum hatalarını ve kontrol sinyallerinin gösterir.



Şekil 4.18. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için hem genlik hem de faz değişiklikleri altında MAE maliyet fonksiyonlu Bulanık-PSO ve PID-PSO kontrolörlerin performansı

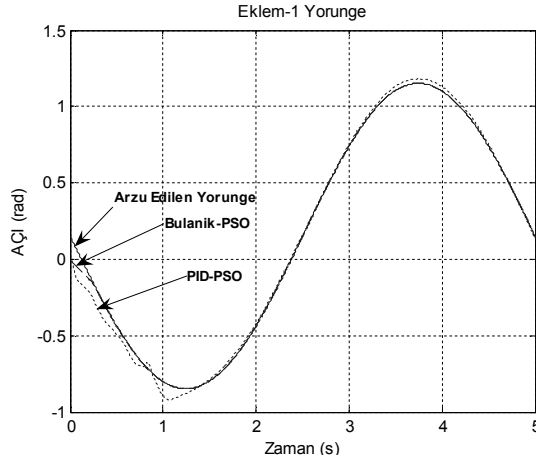


(c)

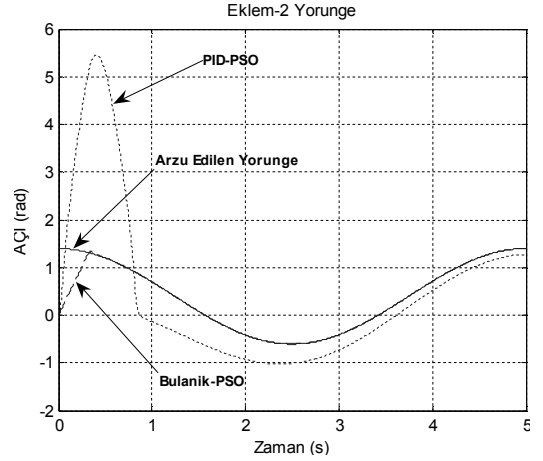


(f)

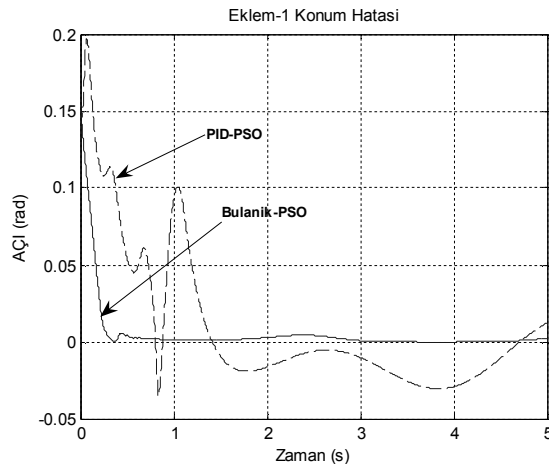
Şekil 4.18. (Devam) Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için hem genlik hem de faz değişiklikleri altında MAE maliyet fonksiyonlu Bulanik-PSO ve PID-PSO kontrolörlerin performansı



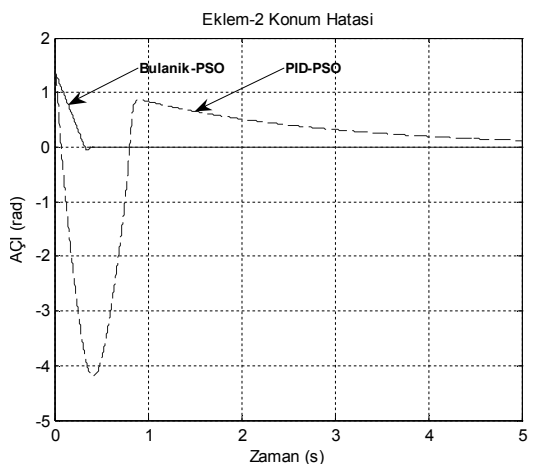
(a)



(d)

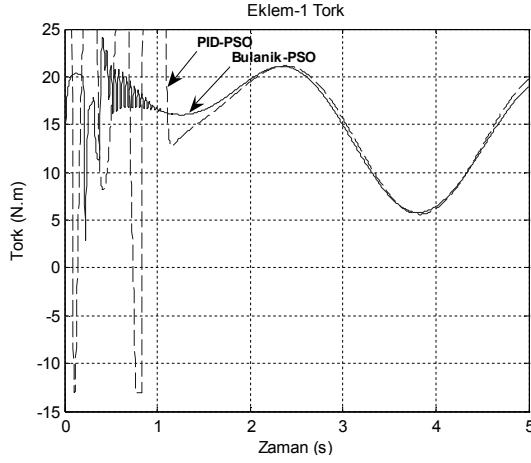


(b)

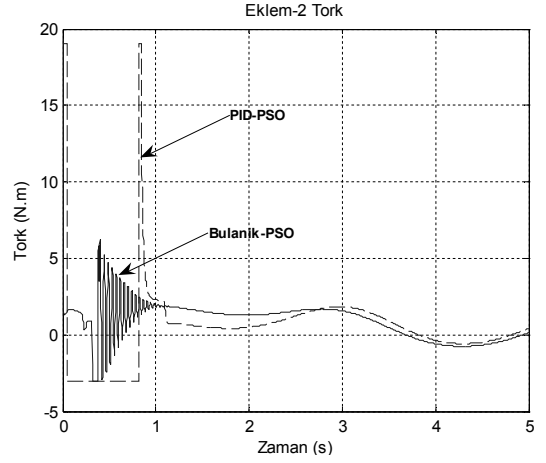


(e)

Şekil 4.19. Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için hem genlik hem de faz değişiklikleri altında MMFAE maliyet fonksiyonlu Bulanik-PSO ve PID-PSO kontrolörlerin performansı



(c)



(f)

Şekil 4.19. (Devam) Yörünge takibi (a, d), konum hataları (b, e) ve kontrol sinyali (c, f) için hem genlik hem de faz değişiklikleri altında MMFAE maliyet fonksiyonlu Bulanik-PSO ve PID-PSO kontrolörlerin performansı

Üç farklı maliyet fonksiyonuna göre PSO ile parametreleri ayarlanan kontrolörlerin bu sağlamlık testindeki (genlik ve fazın değiştirilmesi) maliyet değerleri Tablo 4.9, 4.10 ve 4.11’de verilmektedir. Tablolar incelendiğinde Bulanik-PSO kontrolörü, MRSE, MAE ve MMFAE maliyet fonksiyonların tümünde ve tüm yörünge değişikliklerinde PID-PSO kontrolöründen daha iyi olduğu ortaya çıkmaktadır.

Tablo 4.9. Verilen yörünge değişikliği durumunda MRSE fonksiyonuna göre maliyet değerleri

A	Eklem 1		Eklem 2		Eklem 1 ve 2	
	Bulanik-PSO	PID-PSO	Bulanik-PSO	PID-PSO	Bulanik-PSO	PID-PSO
0.1	0.0092	0.0112	0.013	0.0223	0.0106	0.0177
0.5	0.0171	0.0284	0.0098	0.0118	0.0147	0.0233
1	0.0369	0.0723	0.0125	0.0121	0.031	0.0543
ϕ	Bulanik-PSO	PID-PSO	Bulanik-PSO	PID-PSO	Bulanik-PSO	PID-PSO
0.3927	0.0304	0.0451	0.0135	0.0239	0.0327	0.0518
0.7854	0.0265	0.0401	0.0125	0.0191	0.0277	0.0432
4.7124	0.0059	0.008	0.019	0.178	0.0132	0.1819
A & ϕ	Bulanik-PSO	PID-PSO	Bulanik-PSO	PID-PSO	Bulanik-PSO	PID-PSO
0.1	0.0154	0.0213	0.014	0.028	0.0178	0.0326
0.3927						
0.5	0.058	0.0879	0.0112	0.015	0.0576	0.0883
0.7854						
1	0.0037	0.0086	0.0455	0.5935	0.0395	0.4839
4.7124						

Tablo 4.10. Verilen yörünge değişikliği durumunda MAE fonksiyonuna göre maliyet değerleri

A	Eklem 1		Eklem 2		Eklem 1 ve 2	
	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.1	0.0102	0.0122	0.0146	0.0266	0.0122	0.0216
0.5	0.018	0.0301	0.0105	0.0127	0.0154	0.0244
1	0.0375	0.0751	0.0151	0.0154	0.0341	0.0589
ϕ	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.3927	0.0313	0.0477	0.0152	0.0281	0.0345	0.059
0.7854	0.0274	0.0425	0.0138	0.0217	0.029	0.0475
4.7124	0.0067	0.0095	0.0224	0.1871	0.0163	0.1803
A & ϕ	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.1	0.0164	0.0232	0.0158	0.0342	0.0199	0.0404
0.3927						
0.5	0.0585	0.0914	0.0118	0.0155	0.0578	0.0907
0.7854						
1	0.0045	0.011	0.0551	0.6366	0.0426	0.5347
4.7124						

Tablo 4.11. Verilen yörünge değişikliği durumunda MMFAE fonksiyonuna göre maliyet değerleri

A	Eklem 1		Eklem 2		Eklem 1 ve 2	
	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.1	21.9532	22.7667	21.7993	22.6194	22.006	22.8431
0.5	20.8613	21.5558	21.5747	22.3917	20.6806	21.4049
1	17.8119	17.9512	21.0969	21.8584	17.1554	17.4827
ϕ	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.3927	21.9717	22.6542	21.698	22.5033	21.9437	22.6299
0.7854	21.9352	22.6326	21.7074	22.4983	21.9025	22.5944
4.7124	21.549	22.3137	21.8267	22.714	21.6177	22.5074
A & ϕ	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO	Bulanık-PSO	PID-PSO
0.1	22.0155	22.7905	21.7836	22.6095	22.054	22.8659
0.3927						
0.5	21.5242	21.885	21.4784	22.2566	21.2774	21.5963
0.7854						
1	16.5288	16.9958	21.5076	23.6514	16.4247	17.711
4.7124						

Bu bölümde, verilen robot yörüngesini kontrol etmek için Bulanık ve PID kontrolörlerin parametreleri, PSO algoritması kullanılarak optimize edilmiştir. Kontrolörlerin parametrelerini optimize edilmesi işleminde, maliyet fonksiyonların etkilerini incelemek için PSO algoritmasında 3 farklı maliyet fonksiyonu (MRSE, MAE ve MMFAE) kullanılmıştır. Ayrıca, kontrolörlerin sağlamlığını test etmek amacıyla kütle değişiklikleri yapılmış, farklı varyanslı gürültüler eklenmiş ve eğitilen yörüngede değişiklikler yapılmıştır.

4.3. İki Serbestlik Dereceli Robotun Kesir Dereceli PID Yörünge Kontrolü

Bu kısımda, robot yörünge kontrolü için evrimsel algoritmalar yardımı ile parametreleri ayarlanan kesir dereceli PID (FOPID) kontrolörlerin karşılaştırılması yapılacaktır. Bu karşılaştırmada, iki farklı optimizasyon tekniği olan PSO ve gerçek kodlu genetik algoritma (GA) kullanılacaktır. Bölüm 4.2.1’ de verilen iki eksenli düzlemsel robot ile Bölüm 4.2.3’te verilen maliyet fonksiyonları kullanılarak PSO ve GA ile FOPID kontrolörlerin parametreleri ayarlanacaktır. Ayrıca, farklı koşullar altında PSO ve GA’nın performansını karşılaştırmak ve bu algoritmalara göre ayarlanan FOPID kontrolörlerin sağlamlığını test etmek için sistem model parametreleri ve verilen yörünge değiştirilecek ve de sisteme gürültü eklenecektir. Simülasyon ortamında elde edilen sonuçlar detaylı şekilde verilecektir.

4.3.1. Kesir dereceli PID kontrolör

PID kontrolörün kalitesini ve sağlamlığını arttırmanın olanaklarından biri, türev ve integral derecesi tam sayı olmayan kesir dereceli kontrolörler kullanmaktır. Kesir dereceli bir PID kontrolör, λ dereceli integratör ve μ dereceli türevi içeren bir PID kontrolörün genelleştirilmiş şekli olan $PI^\lambda D^\mu$ kontrolörüdür.

$PI^\lambda D^\mu$ kontrolörün diferansiyel denklemi aşağıda verildiği gibidir.

$$u(t) = k_p e(t) + k_i D_t^{-\lambda} e(t) + k_d D_t^\mu e(t) \quad (4.22)$$

FOPID kontrolörün sürekli transfer fonksiyonu Laplace dönüşümü ile aşağıda verildiği gibi elde edilir.

$$G_c(s) = K_p + K_I s^{-\lambda} + K_D s^\mu \quad (4.23)$$

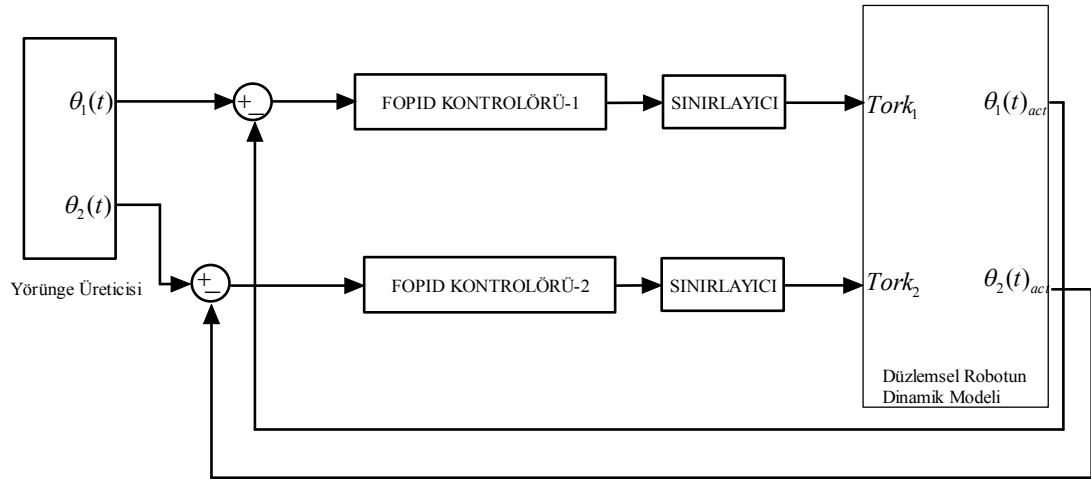
FOPID kontrolörün tasarımı için üç parametre (K_p , K_I , K_D) ve kesirli iki derece (λ , μ), verilen bir sistem için en iyi şekilde belirlenebilir.

4.3.2. PSO ile kesir dereceli PID kontrolörün ayarlanması

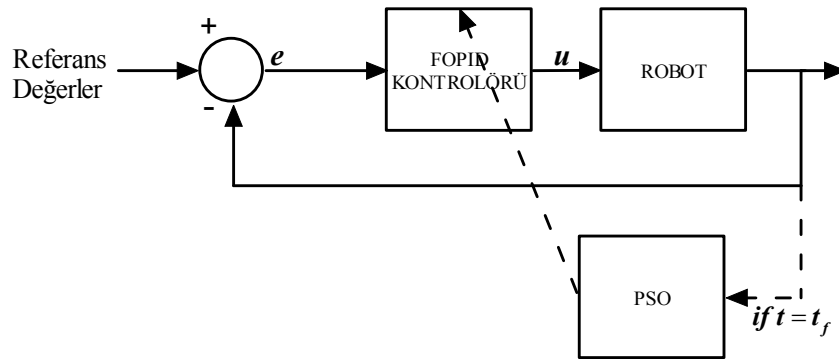
Her bir ekleme yer alan FOPID kontrolörleri 5 parametre içerdiğinden toplam 10 parametre PSO algoritması ile optimize edilecektir. Böylece, PSO algoritması 10 boyutlu uzayda kontrolör parametrelerin tümünü arar. Aşağıdaki denklemde verildiği gibi her bir parçacık, gerçel değerler atanmış 10 elemandan oluşur. Algoritmadaki bir parçacığın düzeni, aşağıda verildiği gibi gösterilir.

$$P_i = [K_{p_1}, K_{d_1}, \mu_1, K_{i_1}, \lambda_1, K_{p_2}, K_{d_2}, \mu_2, K_{i_2}, \lambda_2] \quad (4.24)$$

Yukardaki denklemde $K_{p_1}, \dots, \lambda_1$ ve $K_{p_2}, \dots, \lambda_2$ parametreleri, sırasıyla birinci ve ikinci kontrolörlerin elemanlarına karşılık gelmektedir.



Şekil 4.20. Düzlemsel robotun ve FOPID kontrolörlerin simülük modeli



Şekil 4.21. PSO ile FOPID kontrolör parametrelerin ayarlanma süreci

FOPID kontrolörleri ile kontrol edilen robot sistemin blok diyagramı Şekil 4.20’de gösterilmektedir. Şekil 4.21, PSO ile FOPID kontrolör parametrelerinin optimize edilme sürecinin blok yapısını göstermektedir. FOPID kontrolörlerin tüm parametreleri her simülasyon zamanı bitiminde (t_f) güncellenir.

Burada, hesapsal yükü göz önünde bulundurarak popülasyon büyüklüğü 10 parçacığa göre ayarlanmıştır. Dolayısıyla, her bir parçacık P_i , 10 eleman içerdiğinden sürünün büyüklüğü 10×10 ’dur. Ayarlama işlemi sırasında sürü içindeki bir parçacığa ait bir elemanın değerleri, makul aralığı aşabilir. Bu durumda, pratik gereksinimlerden ve farklı sistemlerin uygulamalarında FOPID kontrolörün parametrelerini ayarlama işlemi üzerinde odaklaşan makalelerden esinlenerek, FOPID kontrolörüne ait parametrelerin alt ve üst sınırları Tablo 4.12’de verilmektedir.

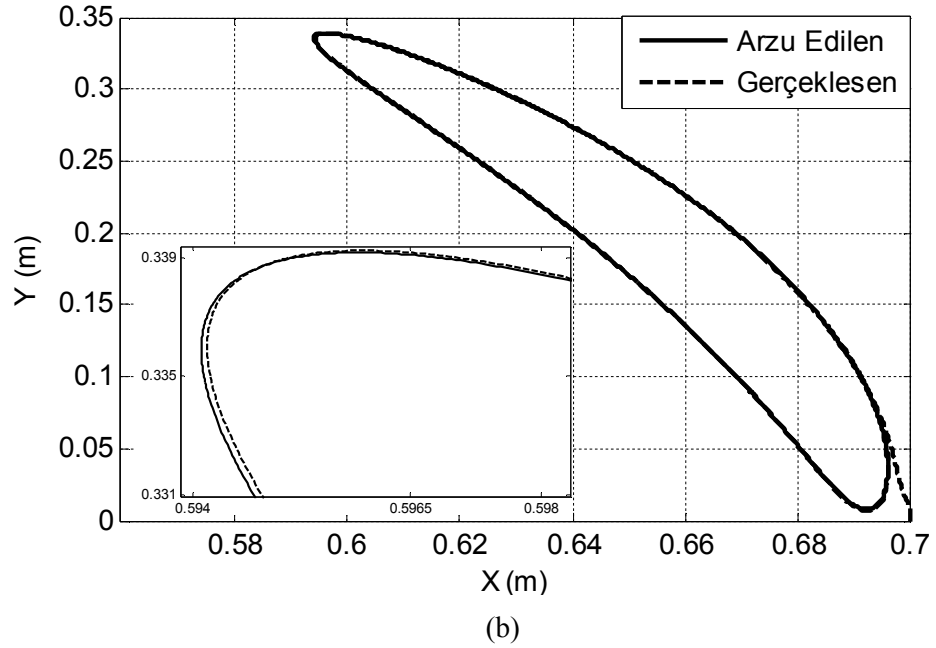
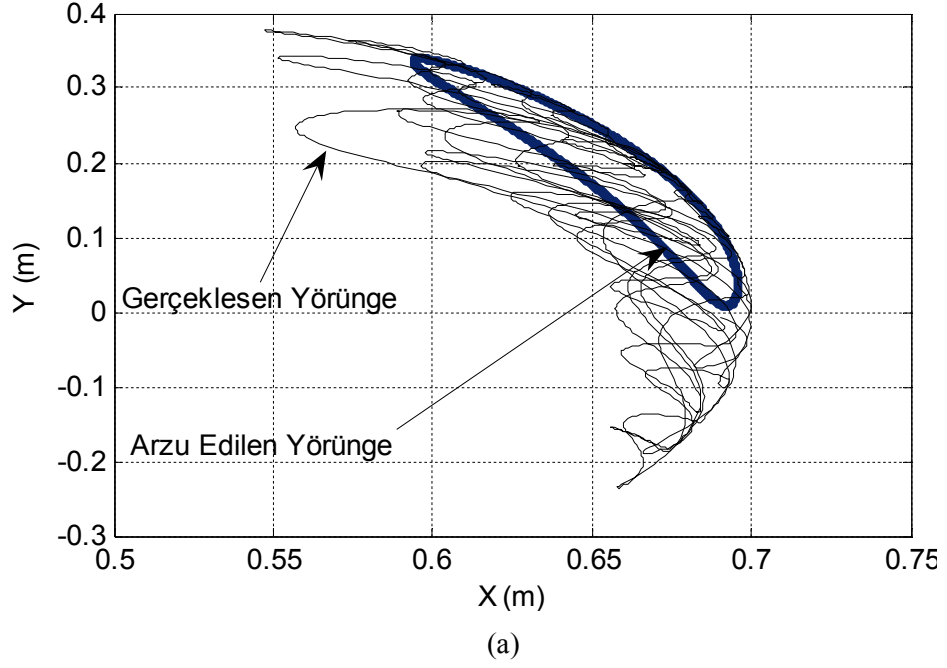
Tablo 4.12. FOPID kontrolör parametrelerinin sınır değerleri

Sınırlar	K_P	K_I	K_D	λ	μ
Üst	2000	2000	100	2	2
Alt	0	0	0	0	0

Şekil 4.21’de gösterilen PSO bloğunda, her bir parçacık için maliyet fonksiyonu hesaplanır ve de her simülasyon zamanı sonunda (t_f) pbest ve gbest değerleri elde edilir. Her bir parçacık için parçacığın hızı hesaplandıktan sonra parçacığın konumu güncellenir. Bu algoritma, verilen maksimum adım sayısına ya da minimum maliyet ölçüt değerine ulaşıncaya kadar çalıştırılır. Algoritmada yer alan χ , φ_1 ve φ_2 sırasıyla 0.76, 2.05 ve 2.05 olarak ayarlanmıştır.

Bölüm 4.2.3’te verilen yörünge denklemine göre, PSO algoritması, 3 farklı maliyet fonksiyonundan birisi ile her bir adımda 10 parçacık çalıştırmak suretiyle toplam 60 adım boyunca çalıştırılmıştır.

Şekil 4.22, verilen yörünge boyunca birinci adımdan son adıma kadar sürü içindeki bir parçacığın (P_1) hareketini gösterir.



Şekil 4.22. Ayarlama işlemi sırasında birinci adımdan (a) son adıma (b) kadar bir parçacığın hareketi

4.3.3. GA ile kesir dereceli PID kontrolörün ayarlanması

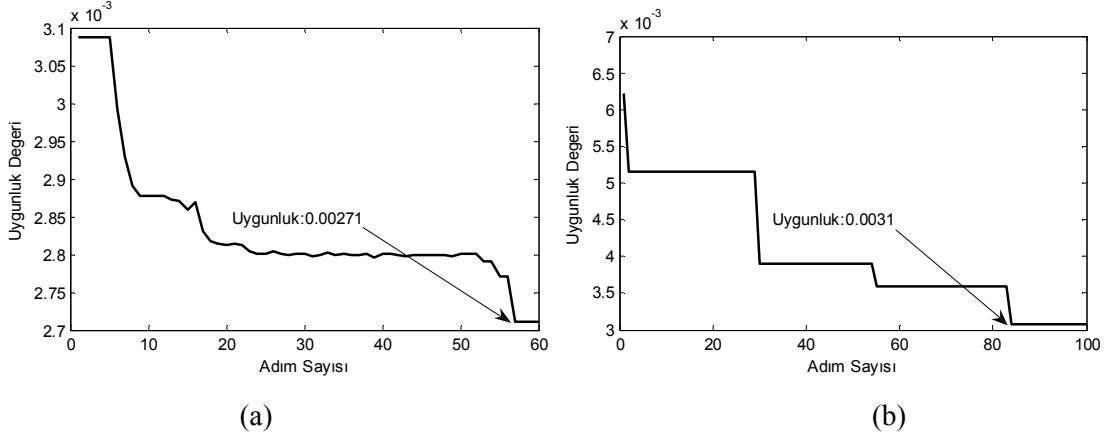
GA, rastlantısal arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan bir arama tekniğidir. Bir veri grubu içinde özel bir veriyi bulmak için kullanılır. Genetik algoritma bir evrimsel (gelişimsel) hesaplama tekniği olup, göz önüne alınan probleme ilişkin bir olası çözümler grubu (popülasyon) ile

çalışmaya başlar [110]. GA popülasyonundaki bireyler doğadaki gibi eşleşir ve çoğalırlar. Popülasyondaki farklı bireyler uygunlukları ile orantılı olan çoğalma hızları ile belirlenirler. Uygunluk, problemdeki amaç fonksiyonundan hareketle oluşturulur. GA bir biyolojik gene benzer bir yapıya sahiptir ve seçim, çoğalma, çarpazlama ve değişim işlemlerinin uygulanması ile optimal bir çözüm araştırır. En iyi birey optimizasyon süresince hayatta kalır.

Burada, önerilen tasarım yönteminin avantajlarını vurgulamak amacıyla gerçel değerli genetik algoritmaya dayanan bir FOPID (FOPID-GA) tasarlanacaktır. Bölüm 4.2.3'te verilen maliyet fonksiyonları (MRSE, MAE ve MMFAE) kullanılarak GA ile kontrolörün parametreleri ayarlanır. Tam bir karşılaştırma yapabilmek için PSO'ya dayanan FOPID (FOPID-PSO) ve FOPID-GA kontrolörlerinin optimizasyon şartları aynı olmalıdır. O nedenle, optimizasyon işleminde kontrol sinyallerin mümkün değer aralığı, tüm eklemlerde üst sınır için 30 N.m ve 20 N.m ve alt sınır için -14 N.m ve -3 N.m olarak sınırlandırılmıştır. Ayrıca, FOPID kontrolörü için başlangıç parametreleri, PSO'da verilen başlangıç parametreleri ile aynı seçilmiştir. GA'nın parametreleri aşağıda verildiği gibi belirlenmiştir.

- Seçim: Normalleştirilmiş geometrik seçim ('normGeomSelect')
- Amaç Fonksiyonu: MRSE, MAE ve MMFAE
- Çarpazlama: Aritmetik Çarpazlama ('arithXover')
- Mutasyon: Düzgün metot ('unifMutation')
- FOPID parametre sayısı: 5
- Popülasyon sayısı: 10
- Adım sayısı: 100

Farklı evrimsel tekniklerin arama performanslarını karşılaştırmak için PSO ve GA algoritmaları, 3 farklı maliyet fonksiyonu ile FOPID parametrelerin optimizasyon işleminde uygulanmıştır. Şekil 4.23'te, MRSE maliyet fonksiyonu kullanılarak her iki algoritmanın uygunluk değerleri gösterilmektedir.



Şekil 4.23. Optimizasyon süreci sırasında MRSE maliyet fonksiyonlu FOPID-PSO (a) ve FOPID-GA (b) kontrolörlerin yakınsama oranları

Şekil 4.23'e bakıldığında, FOPID-PSO kontrolörün uygunluk değeri, 55 adımdan sonra 0,0027 değerine ulaşmıştır. Diğer taraftan FOPID-GA kontrolörün uygunluk değeri, 82 adımdan sonra 0,0031 değerine ulaşmıştır. Şekilde görülebildiği gibi, PSO, başlangıçta hızlı bir yakınsama göstermiş fakat en iyi noktaya erişmek için daha fazla adıma gereksinim duymuştur. Yaklaşık 60 adım boyunca PSO algoritması daha iyi yakınsama sağlamıştır. Ayrıca, elde edilen sonuçlara bakıldığında, PSO algoritması GA ile karşılaştırıldığında en iyi FOPID kontrolör parametrelerini hızlı ve etkili bir şekilde arayabildiğini göstermiştir.

4.3.4. GA ve PSO tekniklerin karşılaştırılması

Üç farklı maliyet fonksiyonu kullanarak iki farklı algoritma ile ayarlanan FOPID kontrolörün parametreleri, Tablo 4.13 ve 4.14'te verilmiştir.

Tablo 4.13. Tasarlanan $FOPID^{\lambda} D^{\mu}$ kontrolörün PSO ile ayarlı parametreleri

FOPID Parametreleri	PSO-MRSE		PSO-MAE		PSO-MMFAE	
	Ekleml-1	Ekleml-2	Ekleml-1	Ekleml-2	Ekleml-1	Ekleml-2
K_p	994	1086	677	1650	241	273
K_i	603	25	582	117	1550	147
K_d	31	8	23	17	27	11
λ	1.014	1.044	0.988	1.018	1.0329	1.1350
μ	1.011	1.1711	0.985	1.009	0.9906	0.8991

Tablo 4.14. Tasarılan $FOPID^{\lambda}D^{\mu}$ kontrolörün GA ile ayarlı parametreleri

FOPID Parametreleri	GA-MRSE		GA-MAE		GA-MMFAE	
	Ekleml-1	Ekleml-2	Ekleml-1	Ekleml-2	Ekleml-1	Ekleml-2
K_p	1074	329	980	1339	157	317
K_i	628	984	660	75	1030	1157
K_d	21	18	30	19	31	21
λ	1.1423	0.9080	1.15	0.98	0.5427	0.4256
μ	1.0274	0.8333	0.946	0.89	1.0883	0.8739

Tablo 4.13 ve 4.14 incelendiğinde, birinci eklem için MRSE maliyet fonksiyonuna göre ayarlanan FOPID-PSO kontrolörün parametreleri, FOPID-GA kontrolörün parametrelerine hemen hemen yakındır. Ancak, ikinci eklem için her iki algoritmadan elde edilen tüm parametreler birbirinden farklıdır. Bunun sebebi, ikinci eklem, kontrolördeki parametre değişimlerine karşı duyarlı oluşudur. Beklenildiği gibi, her bir maliyet fonksiyonu, önceliğine dayalı olarak farklı en iyi kontrolör parametrelerini üretmiştir. Özellikle, MMFAE maliyet fonksiyonu, diğer maliyet fonksiyonların ürettiği parametrelerden çok farklı kontrolör parametreleri üretmiştir. PSO ile elde edilen FOPID kontrolöründeki K_p , K_i ve K_d parametreleri, farklı maliyet fonksiyonlarında farklı değerler alırken λ ve μ ise çok az değişiklik göstermiştir. MMFAE maliyet fonksiyonlu GA ile elde edilen FOPID kontrolöründeki λ parametresi, her iki eklem için diğer maliyet fonksiyonlarından elde edilen değerlerden çarpıcı bir şekilde farklılık göstermektedir.

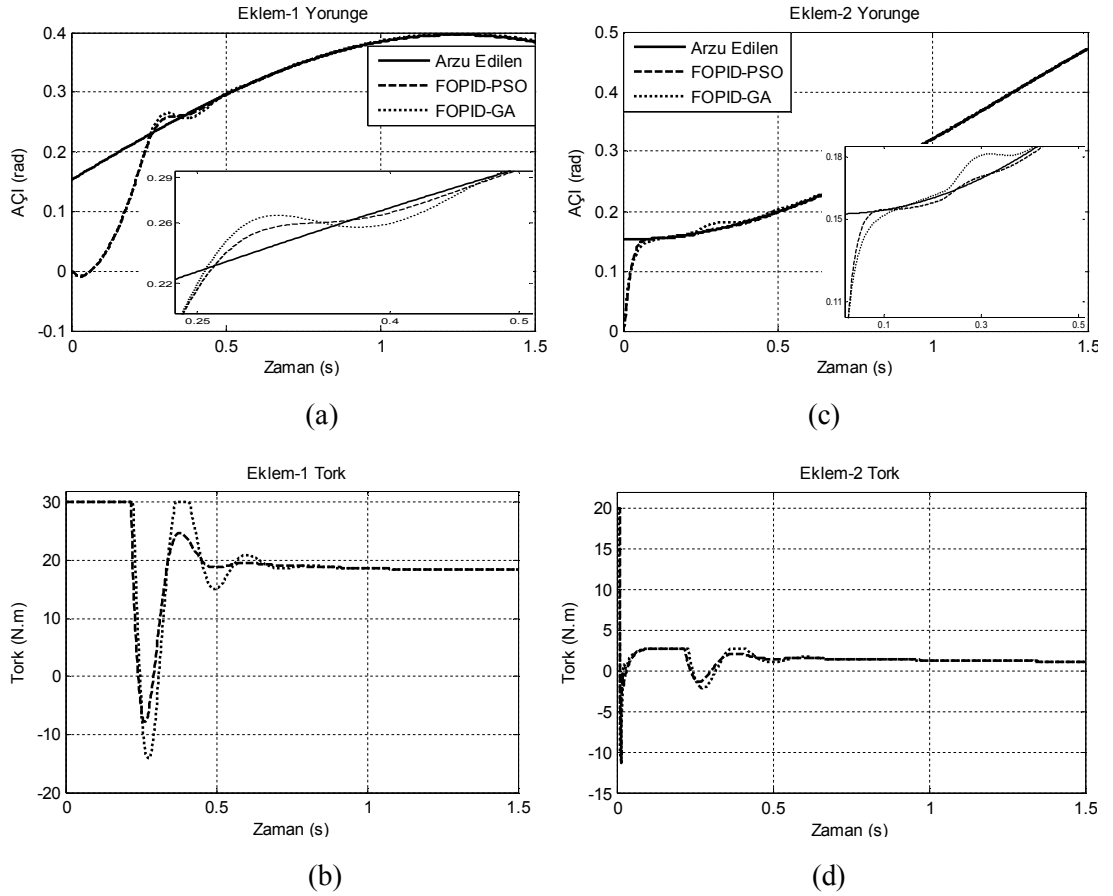
Bu maliyet fonksiyonları kullanılarak her iki algoritmaya göre ayarlanan kontrolörlerin herhangi bir bozucu etki altında olmadan göstermiş olduğu maliyet değerleri ve Denklem (4.19)'a göre hesaplanan ortalama RMS değerleri Tablo 4.15 'te verilmektedir. Tabloya bakıldığında PSO'nun, daha iyi uygunluk değerlerini elde edebildiği görülmektedir. Kontrolörlerin ortalama RMS değerleri incelendiğinde birbirine yakın kontrol sinyallerin üretildiği anlaşılmaktadır.

Tablo 4.15. Herhangi bir bozucu etki olmaksızın kontrolörlerin göstermiş olduğu maliyet ve ortalama RMS değerleri

Kontrolör	MRSE	U_{MRMS}	MAE	U_{MRMS}	MMFAE	U_{MRMS}
FOPID-PSO	0.0027	6.5389	0.0030	4.2507	7.2381	3.9283
FOPID-GA	0.0031	5.2384	0.0032	4.4696	7.2465	3.9424

FOPID-PSO ve FOPID-GA kontrolörlerin performanslarını göstermek ve bu kontrolörlerin sağlamlığını test etmek için sırasıyla her bir eklemin kütlesi 3 katına çıkarılacak, sisteme gürültü eklenecek ve farklı yörüngeler verilecektir.

İlk olarak her bir eklemdaki robot kolunun kütlesi 3 katına çıkarılır. Şekil 4.24, yörünge takibi boyunca MRSE maliyet fonksiyonlu kontrolörlerin kontrol sinyalini ve eklemlerin konumunu gösterir. Aynı zamanda, MAE ve MMFAE maliyet fonksiyonları kullanılarak da simülasyonlar gerçekleştirilmiştir. Bu maliyet fonksiyonlarına göre elde edilen sonuçlar Tablo 4.16’da verilmiştir. Tüm simülasyonlar göz önünde bulundurulduğunda FOPID-GA kontrolörü, FOPID-PSO kontrolöründen daha yavaş cevap zamanına sahip olduğu görülür. FOPID-PSO kontrolörü, her iki eklem için yörüngeyi başarılı bir şekilde takip etmektedir. Görülebildiği gibi, PSO ile elde edilen yörünge takip performansı, GA ile karşılaştırıldığında kütle değişimlerine karşı daha sağlamdır.



Şekil 4.24. Kütle değişikliği altında MRSE maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin kontrol sinyali (b,d) ve konumları (a,c)

Tablo 4.16’da, kütle değişikliği altında kontrolörlerde meydana gelen hata değerlerine göre hesaplanan maliyet fonksiyon değerleri ve ortalama RMS değerleri verilmektedir. Tablodan da anlaşılacağı üzere bu iç bozucu etki altında, FOPID-PSO kontrolörü tüm maliyet fonksiyonlarında, FOPID-GA kontrolöründen daha iyi sonuçlar vermiştir. Tabloda verilen kontrolörlerin ortalama RMS değerlerindeki yakınlık, hemen hemen aynı kontrol sinyallerinin üretildiğini göstermektedir. Bu deneysel sonuçlar, birbirine yakın üretilen kontrol sinyallerinde, FOPID-PSO kontrolörün daha doğru yörünge takip performansını başarabildiğini göstermiştir.

Tablo 4.16. Kütle değişikliği altında kontrolörlerin maliyet ve ortalama RMS değerleri

Kontrolör	MRSE	U_{MRMS}	MAE	U_{MRMS}	MMFAE	U_{MRMS}
FOPID-PSO	0.0080	11.1747	0.0088	11.2338	21.6646	11.0601
FOPID-GA	0.0081	11.2367	0.0091	11.3114	21.6945	11.1152

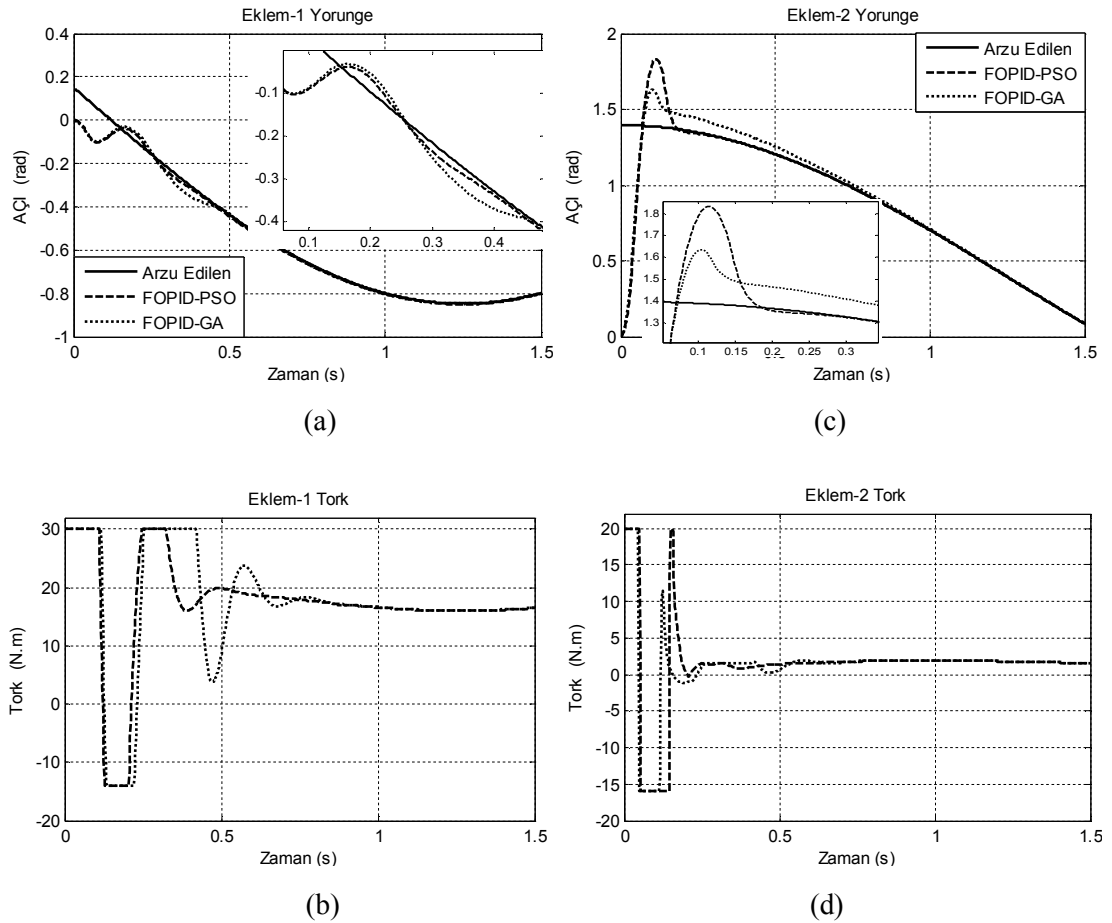
Kontrolörlerin sağlamlık değerlendirilmesinin ikinci aşamasında, robot sistemine değişik varyanslı gürültü eklenecektir. Bu gürültü, sırasıyla birinci ekleme sonra ikinci ekleme ve en son her iki ekleme birden uygulanacaktır. Farklı gürültü varyanslı yapılan deneylerden MRSE maliyet fonksiyonuna göre elde edilen değerler Tablo 4.17’de verilmektedir. Hata olarak MAE ve MMFAE maliyet fonksiyonları kullanıldığında aynı sonuçlar gözlemlendiği için, MAE ve MMFAE maliyet fonksiyonları ve farklı gürültü varyanslarında kontrolörler, hemen hemen aynı performansı üretmiştir.

Tablo 4.17. Farklı varyanslı gürültü eklendiğinde MRSE fonksiyonuna göre maliyet değerleri

Gürültü Varyansı	Eklem 1		Eklem 2		Eklem 1 ve 2	
	FOPID-PSO	FOPID-GA	FOPID-PSO	FOPID-GA	FOPID-PSO	FOPID-GA
0.1	0.0082	0.0088	0.0085	0.0099	0.0086	0.0103
0.5	0.0090	0.0110	0.0115	0.0140	0.0115	0.0156
0.7	0.0094	0.0122	0.0138	0.0165	0.0136	0.0187
0.8	0.0096	0.0129	0.0151	0.0179	0.0149	0.0203

Yukarıdaki tabloda verilen sonuçlarda görüldüğü gibi, FOPID-PSO kontrolörü, FOPID-GA kontrolöründen daha iyi sonuçlar üretmiş, daha sağlam kontrol performansı göstermiş ve iyi bir şekilde bozucu etkiyi gidermeyi başarmıştır. Gürültünün varyansı artırıldığında, verilen yörüngeden sapmalar meydana geldiği

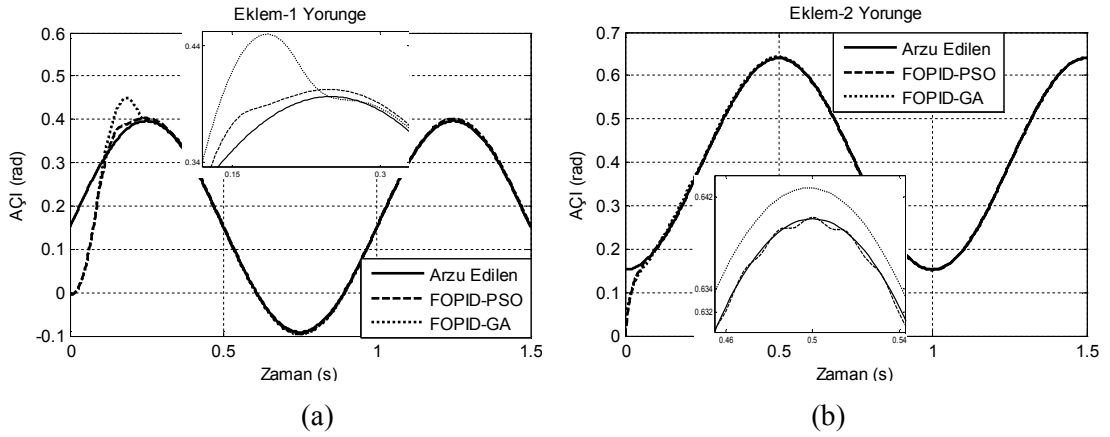
gözlemlenmiştir. Tablo 4.17’de verilen değerlerden açık bir şekilde görülüyor ki, FOPID-GA kontrolöründen elde edilen hata büyüklüğü (gürültünün gücü arttığında), PSO algoritmasına dayalı önerilen ayarlama yöntemi ile karşılaştırıldığında daha yüksektir. Sağlık testinin üçüncüsünde, verilen yörüngenin frekansının değiştirilmesi ve yörüngedeki genlik (A) ve faz (ϕ) değerlerinin sırasıyla birinci eklem sonra ikinci eklem ve son olarak her iki eklem için de değiştirilmesidir. Bu deneyler sonucu, FOPID-PSO kontrolörü verilen yörüngedeki bu değişimler karşısında daha sağlam olduğu gözlemlenmiştir. Örnek olarak göstermek amacıyla deneylerin içerisinde biri seçilmiştir. Seçilen bu deneyin sonucu Şekil 4.25’te, hem genlik hem de fazın değişikliği durumunda MRSE maliyet fonksiyonlu kontrolörlere göre eklemlerin gerçekleştirdiği konum, arzu edilen konum ve kontrol sinyali görülmektedir. Şekil incelendiğinde, FOPID-PSO kontrolörünün, her iki eklemden FOPID-GA kontrolöründen daha sağlam kararlılık ve performans karakteristiğine sahip olduğu anlaşılmaktadır.



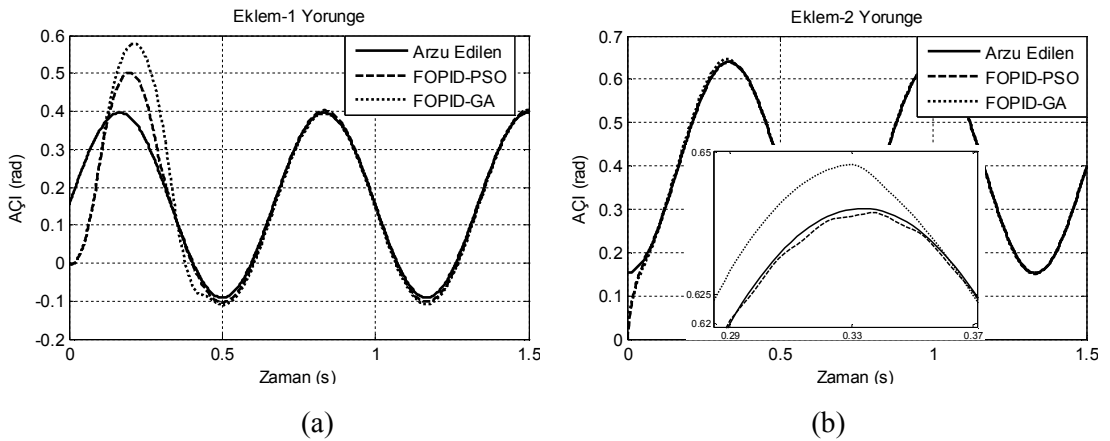
Şekil 4.25. Hem genlik hem de fazın değiştiği durumda MRSE maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin kontrol sinyali (b,d) ve konumları (a,c)

İzleme performansı üzerinde, verilen yörüngedeki frekans değişikliğinin etkilerini göstermek için kontrol edilen robot sistemine farklı frekans değerleri (2π and 3π) uygulanır. Bu durumda, FOPID-PSO ve FOPID-GA kontrolörleri ile kontrol edilen sistemin cevapları Şekil 4.26 ve 4.27’de gösterilmektedir. Şekil 4.26 ve 4.27 incelendiğinde, MRSE maliyet fonksiyonlu FOPID-PSO kontrolörü, FOPID-GA kontrolöründen daha sağlam olduğu ve daha iyi yörünge takibini yapabildiği görülmektedir.

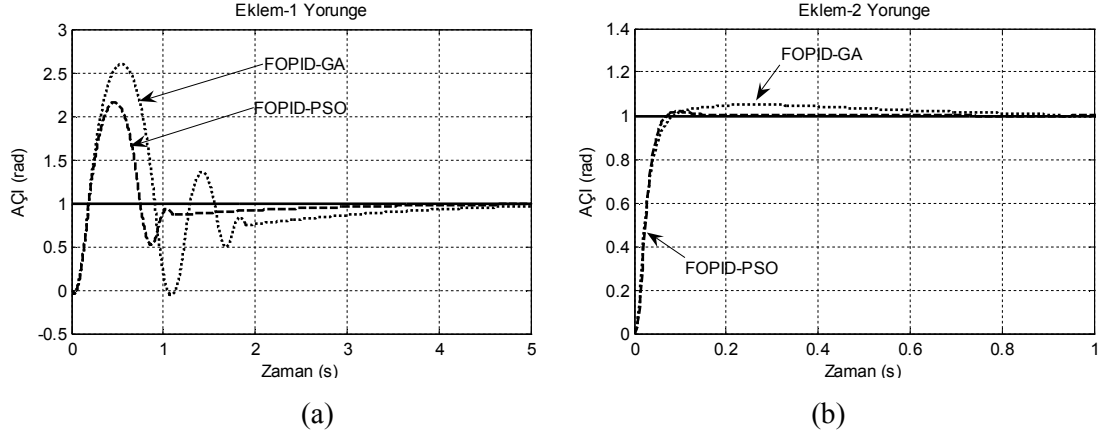
Kontrolörlerin konum kontrol performanslarını (Şekil 4.28) görmek için MRSE maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin birim basamak cevaplarına bakılır. Diğer maliyet fonksiyonlarını kullanarak simülasyonlar yapılmış, ancak MRSE maliyet fonksiyonunu kullanarak yapılan çalışmaların daha geçerli olduğu görülmüştür.



Şekil 4.26. 2π Hz frekansında MRSE (a,b) maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin frekans cevapları



Şekil 4.27. 3π Hz frekansında MRSE (a,b) maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin frekans cevapları



Şekil 4.28. MRSE maliyet fonksiyonlu FOPID-PSO ve FOPID-GA kontrolörlerin birim basamak cevapları (a,b)

Şekil 4.28’de görüldüğü gibi FOPID-PSO ile kontrol edilen sistemin cevabının dinamik özellikleri (taşma ve oturma zamanı), FOPID-GA kontrolöründen çok daha iyi olduğu görülmektedir.

4.4. Sonuç

Bu tez çalışmasında, dinamiği verilen iki eklemlili düzlemsel bir robot koluna sırasıyla PID kontrol, Bulanık Mantık Kontrol ve Kesir Dereceli PID (FOPID) kontrol yapıları uygulanmıştır. Bu kontrolörlere ait parametreler PSO algoritması kullanılarak optimize edilmiştir. İlk olarak PID-PSO ve Bulanık-PSO arasında uygulamalar yapıp sonuçlar gözlemlenmiştir. İkinci uygulama olarak; iki farklı optimizasyon tekniğinin (PSO ve GA) kullanılmasıyla FOPID parametrelerin ayarlanması gerçekleştirilmiştir. Ayrıca, tüm uygulamalarda hem PSO hem de GA için 3 farklı maliyet (MRSE, MAE ve MMFAE) fonksiyonu kullanılarak bu maliyet fonksiyonlarına göre en iyi kontrol parametreleri elde edilmiştir. Parametreleri en iyilenen kontrolörlerin performanslarını göstermek ve bu kontrolörlerin sağlamlığını test etmek için sırasıyla her bir eklemin kütlesi 3 katına çıkarılmış, sisteme değişik varyanslı gürültü eklenmiş ve farklı yörüngeler verilmiştir.

Birinci uygulama için yapılan deneysel sonuçlara göre, Bulanık-PSO kontrolörü, PID-PSO kontrolörü ile karşılaştırıldığında doğru sonuçları daha iyi başarabildiği ve de yörüngeden daha az sapma veya hiç sapma yapmadığı görülmüştür. Robot

yörünge kontrolü için yapılan simülasyon çalışmalarında Bulanık-PSO kontrolörü, daha iyi kontrol performansı sergilemiştir.

İkinci uygulama olarak, PSO ve GA algoritmalarına göre parametreleri optimize edilen FOPID kontrolörü için yapılan deneyler göz önünde bulundurulduğunda, FOPID-PSO kontrolörü, iyi bir performans ve sağlamlığı başarabilmiş ve FOPID-GA kontrolörü ile karşılaştırıldığında daha üstün olduğu görülmüştür. Ayrıca PSO, GA'ya göre daha hızlı arama yapabilmeyi ve daha iyi sonuçlar üretmeyi başarabilmiştir. Bunu yanında FOPID-PSO kontrolörü, PID kontrolörün kararlılığını ve esnekliğini genişletmiştir.

Sonuç olarak, PSO algoritması ile parametreleri ayarlanan Bulanık ve FOPID kontrolörleri kullanılarak hem farklı yörünge hem de değişik tipteki ve büyüklükteki bozuculara karşı iki eklemlili bir robot kolu için ideal bir kontrolör tasarlanabilmiştir.

5. STEWART PLATFORM'UN DİNAMİK ANALİZİ

5.1. Giriş

Stewart Platform'unun kapalı döngü yapısı ve kinematik kısıtlamalarından dolayı dinamik analizi ve dinamik denklemlerinin çıkarımı oldukça karmaşıktır. Ancak yapılan çalışmalarda Stewart Platform'unun dinamik modelinin analizi ve gelişimi, önemli yer teşkil etmektedir. Dinamik model, bilgisayar simülasyonu ve kontrol stratejilerinin geliştirilmesi için önemlidir. Ayrıca dinamik analiz sayesinde eklem kuvvet ve momentlerin tümünün ortaya konulması, yapıyı oluşturan katı bağların ve eyleyicilerin boyutlandırılmasında yarar sağlayabilir. İleri ve ters olmak üzere iki şekilde dinamik analiz yapılır. İleri dinamik analizde; bacaklara ait konum, hız, kuvvet ve sistemin kütle dağılım değerleri verildiğinde hareketli platformun ivmesi bulunur. Ters dinamik analizde ise hareketli platformun ağırlık merkezinin ivmesi, eklemlerin konumu, hızı ve sistemin kütle dağılım değerleri verildiğinde bu ivmeyi sağlayacak kuvvetler bulunur.

Stewart Platform'unun dinamik modelini elde etmek için farklı yöntemler kullanılmaktadır. Bu yöntemlerden bazıları; Newton-Euler Metodu, Lagrange formülasyonu, sanal çalışma ilkesi, Kane yöntemi ve Screw yöntemidir. Burada önemli olan, hangi yöntem kullanılırsa kullanılsın sonuçta aynı dinamik denklemlerin elde edilmesidir.

Newton-Euler yönteminde, manipülatörü oluşturan tüm parçaların dinamik denklemleri ayrı ayrı hesaplanır ve hareketin ani veya sonsuz küçüklikte yönleri incelenir. Dinamik denklemler çıkartılırken hız ve kuvvet vektörlerinden faydalanılır. Newton-Euler yönteminde, Lagrange yöntemindeki gibi karmaşık türev hesaplamalarına gerek duyulmamaktadır.

Yöntemdeki zorluk ise yapıyı oluşturan her bir bağ için serbest çizim diyagramı çizmek ve bu bağa ait denklemlerini yazmak gerekir. Eğer sistemin yapısı karmaşıklaştığında örneğin paralel manipülatörler için Newton-Euler denklemleri yazmak zorlaşabilir. Paralel manipülatörlerde her bir bağa ait serbest çizim diyagramları çizilir ve bu bağa ait Newton-Euler denklemleri çıkartılırsa tüm sistemin kısıt, kuvvet ve momentleri bulunabilir.

Dinamik denklemlerin çıkartılmasında yaygın bir şekilde kullanılan diğer bir yöntem Lagrange yöntemidir. Bu yöntemde dinamik denklemler elde edilirken tüm sistemin kinetik ve potansiyel enerjisinden faydalanılır. Newton-Euler yöntemine göre daha kolay olan ve karmaşık türev hesaplamaları içeren bu yöntemde, dinamik denklemler elde edilirken dikkatli hesaplama yapmak gerekir. Ancak günümüz bilgisayar programları sayesinde bu tip karmaşık türev hesaplamaları rahatlıkla çözülerek hesaplamada meydana gelebilecek hata oranı en aza indirilebilmektedir. Ayrıca yöntemde, dinamik denklemler oluşturulurken enerji bağıntıları kullanıldığından Newton-Euler yönteminin aksine skaler değerler kullanılmaktadır. Sonuç olarak, Lagrange yönteminde sistemin kinetik ve potansiyel enerjilerinden yola çıkılarak hız ve ivme karakteristikleri bulunmakta ve bu değişkenlerle de sistemin diğer dinamik değişkenleri hesaplanabilmektedir.

Paralel manipülatörlerde kullanılan diğer yöntem ise sanal iş prensibi yöntemidir. Bu yöntemde ilk önce doğrusal ve açısal ivmeler kullanılarak atalet kuvvet ve momentler hesaplanır. Sistemi oluşturan tüm mekanizmanın statik dengede olduğu kabul edilerek yöntemin uygulanmasıyla giriş kuvvet ve momentleri bulunur. Bu yöntemde kısıt kuvvet ve momentler hesaplanmadığından hesaplama zamanı daha kısadır. Ancak bu yöntemde başvurulan çalışmalar incelendiğinde yöntemin ters dinamik hesaplarında verimli olduğu fakat ileri dinamik denklemlerin elde edilmesinde karmaşık olduğu belirtilmiştir.

Paralel bir manipülatöre ait dinamik denklemler, Kane'in dinamik modeli kullanılarak da elde edilebilir. Kane yönteminde paralel manipülatörü oluşturan her bacak ve hareketli platform bağımsız olarak değerlendirilir ve dinamik denklemleri

bulunur. Çıkarılan bu dinamik denklemler birleştirilerek tüm sistemin dinamik denklemleri elde edilir.

Literatürde Stewart Platform dinamiği üzerine yapılan çalışmalar, seri bağlı manipülatörlerin dinamik modelleri üzerine yapılan çalışmalarla karşılaştırıldığında nispeten daha azdır. Yapılan en eski çalışma, atalet ve eklem sürtünmesinin ihmal edildiği Fitcher [111] ve Merlet [112] tarafından yapılan bir uygulamadır. Do ve Yong [113], Newton mekaniğini kullanarak Stewart Platform'u için ters dinamik problemini ele aldılar. Eklem sürtünmelerinin olmadığını varsayarak ve ince çubuklar gibi doğrusal eyleyicileri idealleştirerek 36 adet denklem seti çıkarıldılar. Ayrıca, eyleyici kuvvetleri ile yapının şekli arasındaki ilişkiyi analiz ettiler ve eyleyici kuvvetlerini minimum yapacak konfigürasyonlar önerdiler. Reboulet ve Berthomieu [114], 6 serbestlik dereceli bir paralel manipülatörün çalışma uzayı içerisinde tam bir dinamik modelini Newton-Euler metodunu kullanarak çıkardılar. Bir kontrol kuralında kullanılmak üzere çıkarılan bu denklemin, temel dinamik etkileşimlerini hesaba katan bir robot modeli olarak, basit bir matris formunda yazılabileceğini gösterdiler.

Khalil ve Ibrahim [115], paralel robotun ters dinamik modellerini hesaplamak için Newton-Euler yaklaşımıyla genel bir yöntem sundular. Bacak ve platform dinamiği ve de 3 Jacobian matrisinden oluşan bu modelleri tek bir denklem vasıtasıyla ifade ettiler. Bu metodu, farklı yapıdaki 3 serbestlik dereceli robotlar (Gough-Stewart, Space Robot ve C5 Robot) üzerinde uyguladılar. Riebe ve Ulbrich [116], 6 serbestlik dereceli paralel bir robotun ters dinamiğinin gerçek zamanlı hesaplanmasında kullanılmak üzere, her bir bacak tarafından sarf edilen kuvvetlerin gösterimini sağlayan Newton-Euler yaklaşımını kullanarak robotun lineer olmayan hareket denklemlerini çıkardılar. Wang ve arkadaşları [117], 6-UPS paralel bir manipülatörün dinamik modelini, Newton-Euler yaklaşımını kullanarak elde ettiler ve gerçek zamanlı kontrol uygulaması için bu dinamik modelin basitleştirme stratejisini sundular. Ayrıca, çıkarılan dinamik modelin ayrıntılı analizini, bacakların ve platformun atalet kuvvetleri ve momentlerini ayrıştırarak gerçekleştirdiler. Dasguta ve Mruthyunjaya [118], Newton-Euler yaklaşımını kullanarak 6-UPS ve 6-SPS'li Stewart Platform'unun atalet dağılımını ve genel bir yapı için kapalı formda

ters dinamiğini çıkardılar. Kullandıkları yöntemde, ilk önce bir bacağa ait kinematik ve dinamik denklemleri çözdüler. Daha sonra, bacağın hareketli platforma bağlantı noktasındaki kısıt kuvvetini hesapladılar ve son olarak da hareketli platformun kinematik ve dinamik denklemlerini çözdüler. Elde edilen bu denklemleri birleştirerek Stewart Platform dinamiğinin tam bir modelini çıkardılar.

Khalil ve Guegan [119], Gough-Stewart paralel robotun ileri ve ters dinamik modelleri için kapalı form çözümlerini sundular. Bu modellerde Newton-Euler yaklaşımını kullanarak bacakların ve hareketli platformun dinamik denklemlerini elde ettiler. Ayrıca, elde edilen bu modeller sayesinde açık bir şekilde, bacak bağlarının ve platformun atalet parametrelerin belirlenebileceğini gösterdiler. Harib ve Srinivasan [120], Stewart Platform tabanlı yapıların dinamik ve kinematikiğinin analitiksel bir çalışmasını sundular. Bu rijit yapıların dinamik denklemlerini Newton-Euler formülasyonunu kullanarak çıkardılar. Bunun yanında, eyleyici dinamik etkilerini modele dâhil etmek için kartezyen uzayında çıkarılan bu dinamik denklemlerini eklem uzayına dönüştürdüler. Ayrıca, eklemelerin sürtünme etkilerini, çıkarılan dinamik modele eklediler. Khalil ve Ibrahim [121], makalesinde Newton-Euler denklemlerini kullanarak paralel robotların ileri ve ters dinamik modelleri için basit ve genel bir kapalı form çözümü sunmuşlardır. Önerdikleri bu çözümde bacakların ve hareketli platformun dinamiğini ve de Jacobian matrislerini göstermişlerdir. Ayrıca, önerilen çözüm yöntemini farklı yapıları iki paralel robot üzerinde uygulamışlardır. Fried ve arkadaşları [122], Gough-Stewart Platform'un ters dinamik modeli için yeni bir formülasyon önermişlerdir. Newton-Euler denklemlerini basitleştirerek önerdikleri bu yaklaşım Khalil [121] tarafından geliştirilen yönteme dayanmaktadır. Bu yaklaşım, bacak ve platform uzayında ifade edilen atalet matrislerinin açık bir şekilde vurgulandığı formülasyondan oluşmaktadır. Bu çalışmalara ek olarak Newton-Euler yaklaşımını kullanarak paralel robotların dinamiği üzerine birçok çalışmalar yapılmıştır [123-126].

Stewart Platform manipülatörün dinamiğini çıkarmanın diğeri bir yöntemi ise Lagrange formülasyonudur. Bu konuda literatürde Abdellatif ve Heimann [127], 6 serbestlik dereceli paralel kinematik manipülatörlerin ters dinamiğini tanımlayan 6 boyutlu diferansiyel denklemlerini açık ve detaylı bir şekilde sundular.

Basitleştirmeden ve hesapsal olarak verimli bir şekilde Lagrange formülasyonunu kullanarak tam bir modelin çıkarabileceğini gösterdiler. Ayrıca, hesapsal verimlilik için koordinatların seçimi ve formülasyonun önemini ele aldılar. Bunun yanında, elde edilen modelin kullanışlı ve yararlı özelliklerini, farklı simülasyon ve deneysel sonuçlarla gösterdiler. Gou ve Li [128], Stewart Platform manipülatörün bacak dinamiğini içeren Newton-Euler ve Lagrangian formülasyonunu birleştirerek görev uzayı içerisinde kapalı form dinamik denklemlerini çıkardılar. Ayrıca, bacaklar üzerinde rol oynayan eyleyici kuvvetlerini çıkarılan dinamik formülasyona göre belirlediler. Bu formülasyon, özel bir Stewart Platform manipülatörün birkaç ters dinamik problemleri üzerine yapılan çalışmalarda kullanıldığını gösterdiler. Harib [129], Stewart Platform'unun katı gövde dinamik denklemlerini elde etmek için Newton-Euler ve Lagrange formülasyonlarını kullandı. Çıkarılan dinamik denklemleri geçerli kılmak için elde edilen dinamik denklemlerin sayısal sonuçlarını her iki yöneme göre karşılaştırdı. Coriolis, merkezkaç ve yerçekimi kuvvet/tork kısımlarını içeren vektörü Newton-Euler yöntemini kullanarak daha etkin bir şekilde hesaplarken diğer taraftan sistemin atalet matrisini, Lagrange formülasyonunu kullanarak sistemin kinetik enerji ifadesinden kapalı formda direk olarak elde edebildiğini gösterdi. Ayrıca, eyleyicilere ait elektriksel ve mekaniksel dinamiği elde ederek sistemin katı gövde dinamiği ile birleştirdi. Eyleyiciler dinamiği eklem uzayında elde edildiğinden bunun kartezyen uzayına ya da katı gövde dinamiğinin eklem uzayına dönüşmesinin gerekliliğinden bahsetmiştir. Bunun için gerekli olan Jacobian matrisini ve onun zamana göre birinci türevinin elde edilmesini göstermiştir.

Bhattacharya ve arkadaşları [130], Stewart Platform manipülatörünün çevirim içi parametre kestirimi için hesapsal olarak hızlı ve verimli çalışan algoritmalar sundular. Lagrangian formülasyonunu kullanarak dinamik denklemlerini çıkardılar ve çalışma uzayı koordinatlarında ifade edildiğinde, bu dinamik denklemlerin, hesaplama bakımından çok yararlı olan paralel bir yapıyı ortaya koyduğunu gözlemlediler. Lin ve Chen [131], Stewart Platform manipülatörü için dinamik denklemlerin bilgisayar ortamında üretimini sağlayan etkili bir algoritma sundular. Platforma ait dinamik denklemleri Lagrange formülasyonu kullanarak çıkardılar. MATLAB ortamında yazdıkları bir program sayesinde sembolik olarak önerdikleri

algoritmayı uyguladılar. Ayrıca, Lagrange formülasyonuna göre elde ettikleri dinamik denklemlerin verimliliğini doğrulamak için hesaplamalı tork kontrol yöntemini sundular. Hajimirzaalian ve arkadaşları [132], Stewart Platform'un ters dinamik denklemlerini Lagrange formülasyonunu kullanarak çıkardılar. Bunun yanında, Stewart Platform'un ileri dinamiğini simüle etmek ve çözmek için ADAMS mühendislik yazılımını kullandılar. Ayrıca, sayısal bir örnek ile robotun ters ve ileri dinamiği için elde edilen sonuçları ADAMS yazılımını kullanarak karşılaştırdılar. Sonuçta, denklemlerin ve analizin doğruluğunu gerçekleştirdiler. Bu çalışmaların dışında, Stewart-Gough tipi paralel manipülatörler için kinetik ve potansiyel enerjiler ile dinamik denklemlerin elde edilebildiği Lagrange formülasyonun kullanıldığı birçok yayın literatürde yer almaktadır [133-136].

Diğer bir metot olan sanal çalışma ilkesi yöntemi Wang ve Gosselin [137] tarafından Stewart Platform manipülatörünün dinamik denklemleri çıkarılmasında kullanılmıştır. Çalışmada ilk önce, manipülatörün katı gövdelerin her birinin doğrusal ve açısız ivmelerini kullanarak atalet kuvvet ve momentlerini hesapladılar. Sonra, tüm manipülatörün statik dengede olduğunu varsayarak giriş kuvvet ya da torkları çıkarmak için sanal çalışma ilkesi yöntemini kullandılar. Ayrıca, klasik Newton-Euler yaklaşımını kullanarak bir örnek ile sonuçları doğruladılar. Her iki yaklaşımda da benzer sonuçların elde edildiğini ve de Newton-Euler yaklaşımdan %30 daha hızlı bir algoritma olduğunu gösterdiler. Li ve Xu [138], öteleme hareketi yapan 3 serbestlik dereceli PRC (Prismatic-Revolut-Cylindrical) paralel kinematik yapı için sağlam kontrol ve dinamik model üzerine bir çalışma yaptılar. Bu çalışmada, kütle dağılım etkisini ortaya koyarak, sanal çalışma ilkesi yardımıyla basitleştirilmiş dinamik denklemleri çıkardılar ve ADAMS yazılım paketi ile sanal bir prototip üzerinde bu denklemlerin doğruluğunu gösterdiler. Staicu [139], 6-6 Stewart-Gough paralel manipülatörün ters dinamik problemini sanal çalışma ilkesine dayanan bir yaklaşımı kullanarak çözmüştür. Ayrıca, çalışmada Lagrange denklemleri kullanılarak elde edilen sonuçlar doğrulanmıştır.

Tsai [140], Stewart-Gough manipülatörün dinamik denklemlerini çıkarmak için sanal çalışma ilkesine ve Jacobian matrisi kavramına dayanan bir yöntem sundu. Ayrıca, manipülatörün ters dinamiğini çözmek için hesapsal bir algoritma geliştirerek

hareketli platform için verilen birkaç yörüngeye göre simülasyonlar yaptı. Sanal çalışma ilkesi yöntemini kullanarak paralel manipülatörlerin dinamiği üzerine literatürde birçok çalışma mevcuttur [141-144].

Paralel manipülatörlerin dinamik denklemlerin çıkarılmasında kullanılan diğer bir yöntem ise Kane denklemleri yöntemidir. Liu ve arkadaşları [145], Kane denklemine dayalı Gough-Stewart Platform manipülatörü için ileri dinamik denklemlerin yeni bir çıkarımını önerdiler. MATLAB ortamında bu formülasyonu uygulayarak simülasyon sonuçları ile bu yeni yaklaşımın geçerliliğini gösterdiler. Ayrıca, Lagrange formülasyonu ve Newton-Euler yöntemi ile karşılaştırma yaparak çalışmada önerilen modelleme işleminin daha apaçık ve sistematik olduğunu ve de elde edilen dinamik denklemlerin çok az ve öz olduğunu gösterdiler. Meng ve arkadaşları [146], sürekli mknatıslı senkron motorla sürülen bir 6 serbestlik dereceli Stewart Platform'un dinamiğini Kane denklemi yöntemini kullanarak çıkardılar. Bu yöntem ile hem zamanla değişen eşdeğer ataletin hesaplanmasının önlendiğini hem de motor sisteminin mekaniksel hareketlerinin etkilerini gösterdiler. Ayrıca, tipik yörüngeler verilerek simülasyonlar ile Stewart Platform'un performansını elde ettiler. Jianming [147], hidrolik olarak sürülen Stewart Platform'un dinamik modelini Kane yöntemi ile çıkardılar. Hidrolik servo sistemin dinamik modelini çıkararak platform dinamiği ile birleştirdiler. Son olarak, deneysel bir uygulama yaparak çalışmada çıkarılan dinamik model ile gerçek robot arasındaki eşleşme analizinin doğruluğunu gösterdiler. Yang ve arkadaşları [148], 6 serbestlik dereceli paralel bir robotun dinamik denklemlerinin hesapsal verimliliğini geliştirmek ve modeli basitleştirmek amacıyla paralel robotu 7 katı gövdeli olarak ele aldılar ve robotun dinamik modelini Kane yöntemini kullanarak çıkardılar. Ayrıca, çıkarılan dinamik denklemleri doğrulamak amacıyla Simulink ortamında çalışan SimMechanics kullanılarak paralel robotun tam bir bilgisayar modelini oluşturdular. Simülasyon sonuçları ile robotun dinamik modelinin doğru, etkili ve gerçek zamanlı kontrol için daha verimli olduğunu gösterdiler.

Genelleştirilmiş bir koordinat çerçevesinin benimsenmesiyle tüm sistemin enerji hesabına dayalı olan Lagrange ve sanal iş prensibi formülasyonları ile sistemin dinamik denklemleri elde edilebilir. Paralel manipülatörlerin dinamik analizindeki

böyle bir enerji yaklaşımı, screw teorisi aracılığı ile daha da basitleştirilebilir ve standartlaştırılabilir. Bunun için, Gallardo ve arkadaşları [149], sanal çalışma ilkesi ile screw teorisini birleştirerek dinamik bir formülasyon elde ettiler. Sanal çalışma ilkesi yaklaşımını basitleştirmek için Screw teorisini kullandılar. Screw teorisi, üç boyutlu uzayda hızları ve kuvvetleri ifade eden ve hem dönele hem de prizmatik kısımları birleştiren bir yoldur. İlk önce 2 serbestlik dereceli paralel mekanizmayı sonra da Gough-Stewart Platform mekanizmasına uygulanan yöntemin verimliliğini gösterdiler.

Paralel manipülatörlerin dinamik modellenmesi, sistemin kapalı döngü yapısı ve kinematik kısıtlardan dolayı özünde karmaşıktır. Bu sebeple, Lopes makalesinde [150], Stewart Platform'un dinamik modeli için geliştirilmiş momentuma dayalı bir yaklaşım sundu. Geliştirilmiş momentum, manipülatörün her katı gövdesi üzerinde rol oynayan geliştirilmiş kuvvetin kinetik parçasını hesaplamak için kullanılır. Bu yöntemde, aynı çerçevede ifade edilen katı gövdelerin ataleti ve de Coriolis ve merkezci terimli matrisler için analitik ifadeler elde edilir. Yerçekimi kuvvet vektörü ise manipülatörün potansiyel enerjisi kullanılarak elde edilir. Lopes makalesinde önerdiği yaklaşımı kullanarak düşük hesapsal yük ile dinamik modelin elde edilebildiğini göstermiştir.

Genel olarak literatüre bakıldığında paralel mekanizmaların dinamik analizinde Newton-Euler ve Lagrange yöntemleri yaygın olarak kullanılmaktadır. Bu yöntemlere ek olarak sanal iş prensibi, Kane denklemleri, screw teorisi ve geliştirilmiş momentuma dayalı yaklaşım denklemleri kullanarak çözüm yapan araştırmacılar da literatürde yer almaktadır. Böylece, bu yaklaşımlar aynı fiziksel sistemi tanımladığında bunların hepsi eşdeğerdir. Yöntemlere göre elde edilen dinamik denklemlerin karmaşıklık ve hesapsal yük seviyeleri farklı olsa da tüm yöntemler benzer dinamik denklemleri verir. Burada dikkat edilmesi gereken nokta, hangi yöntem kullanılırsa kullanılsın, aynı dinamik denklem sisteminin elde edilmesidir. Önerilen tekniklerin başlıca hedefi ise manipülatörün dinamik modelinin hesaplanmasındaki işlem sayısının küçültülmesidir.

Bu bölümde yapılacak çalışmada, 6 serbestlik dereceli Stewart Platform manipülatörün ters kinematik analizi ve manipülatöre ait kapalı formdaki katı gövde dinamik denklemlerin Lagrange yöntemine göre nasıl elde edileceği anlatılacaktır. İlk önce, verilen hareketli platformun yörüngesine göre referans bacak uzunluklarını hesaplayan manipülatörün ters kinematik denklemleri çıkarılacaktır. Daha sonra, önerilen yöntemle göre hareketli platformun ve bacakların sırasıyla kinetik ve potansiyel enerjileri elde edilecektir. Hareketli platformun hızı ile bacakların hızı arasındaki ilişkiyi veren Jacobian matrisinin iki farklı şekilde türetilmesinden bahsedilecektir. Ayrıca, bacaklarda yer alan eyleyicilerin mekanik ve elektriksel dinamiği çıkarılarak manipülatörün katı gövde dinamiğine eklenecektir. Son olarak, iki farklı Jacobian matrisine göre çıkarılan tüm sistemin dinamiği MATLAB-SIMULINK ortamında modellenerek fiziksel (gerçek) robot üzerinde model doğruluğu için deneyler yapıp sonuçlar gözlemlenecektir.

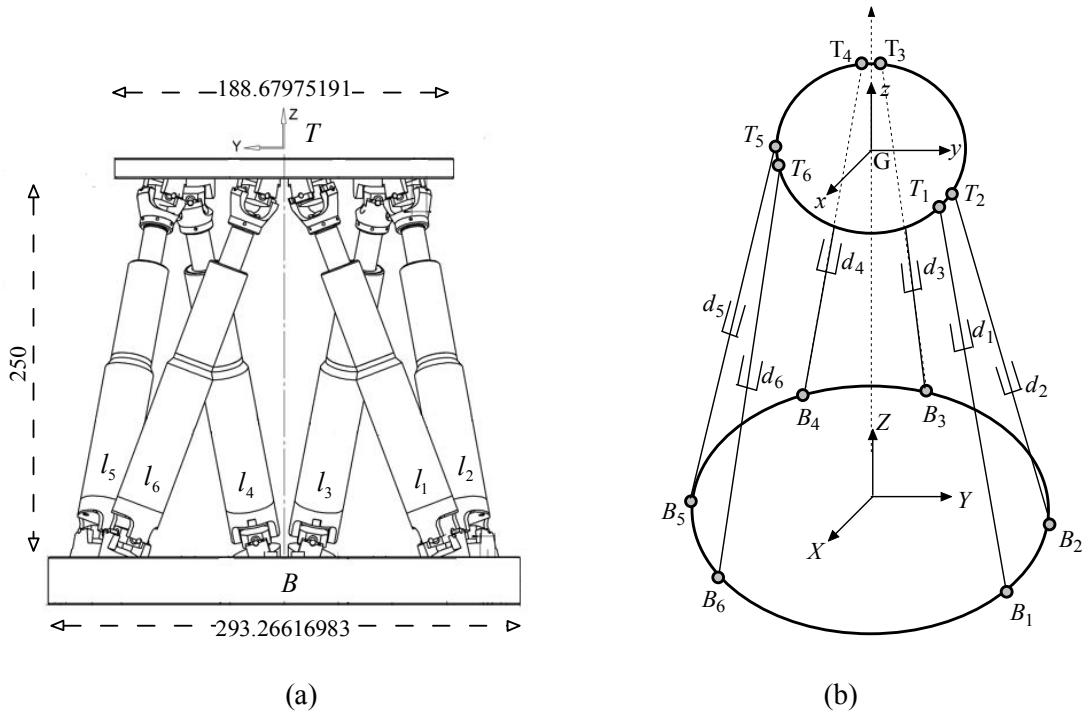
5.2. Ters Kinematik Analizi

Bu bölümde, katı gövde dinamik denklemlerin geliştirilmesi, kontrolör tasarımı ve değerlendirilmesi için gerekli olan Stewart Platform mekanizmasının ters kinematik analizi yapılacaktır. İlk önce, çalışma boyunca kullanılacak olan koordinat sistemleri ve notasyondan bahsedilecek daha sonra Stewart Platform mekanizmasının genel bir sınıfı için ters kinematik çözümü verilecektir.

Genel olarak tüm robotik manipülatörlerde olduğu gibi Stewart Platform'un kinematik analiz incelenirken temel iki kinematik problem ile ilgilenilir. Bunlardan birincisi ileri kinematik problem olup, bacak boyları verildiğinde hareketli platformun sabit alt platforma göre ötelenme ve yönlenmesini bulmaktır. Burada hareketli platformun köşelerini veren vektörün ve vektörleri hareketli platformun koordinat sisteminden alt platformun koordinat sistemine geçiren dönme matrisinin bulunması gerekmektedir. İleri kinematik probleminde, belirli bacak boylarında platformun alabileceği tüm konfigürasyonların bilinmesi gerekir. Ters kinematik problemi, hareketli platformun ağırlık merkezinin konum ve yönlenmesi verildiğinde bacak uzunluklarının bulunmasıdır. Paralel manipülatörlerin ileri kinematik analizi seri manipülatörlere göre daha karmaşıktır. Ters kinematik analizi ise seri

manipülörlere göre daha kolaydır. Yapılan çalışmalara bakıldığında, paralel manipülörlerde ileri kinematik problemin çözümünün tek olmadığı söylenebilir. Bu demektir ki bir giriş, yani bacak boyuna karşılık birden fazla konfigürasyon mevcuttur. Ters kinematik problemin ise çözümü tektir, yani bir hareketli platformun bir konfigürasyonu sadece bir giriş ile oluşabilir.

Şekil 5.1’de Stewart Platform’unun yapısı gösterilmiştir. Şekil 5.1’de görüldüğü gibi alttaki sabit platformun merkezine B - XYZ , üstteki hareketli platformun merkezine ise T - xyz koordinat sistemleri yerleştirilmiştir.



Şekil 5.1. Stewart Platform’u (a) ve kinematik düzenleşimi (b)

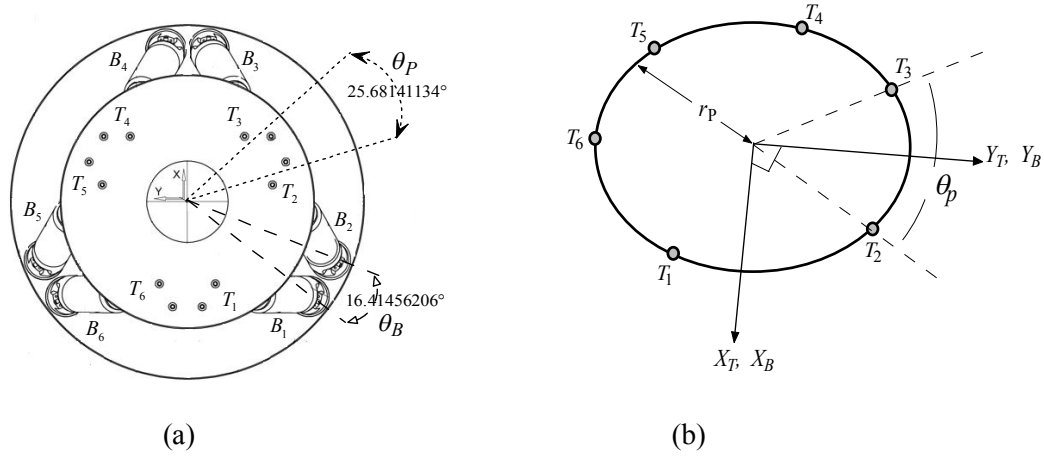
Sabit platformdaki eklemlere B_i ($i=1,2, \dots, 6$) ve hareketli platformdaki eklemlere T_i ($i=1,2, \dots, 6$) koordinat sistemleri yerleştirilmiştir. Her $B_i T_i$ pasif eklemlerini birleştiren altı adet bacak vardır. Üst platformda konum ve yönelim değişimi meydana getirecek olan aktif eyleyiciler bu bacaklar üzerine yerleştirilmiştir. Bu eyleyicilerin tek boyutlu hareketi d_i ($i=1,2, \dots, 6$) şeklinde tanımlanmıştır.

Altı serbestlik dereceli Stewart Platform’u 3 adet yönelim ve 3 adet konum değişkeninden oluşmaktadır. Üst hareketli çerçevenin temel çerçeveye göre konumu $[p_x \ p_y \ p_z]^T$ ile yönelimi ise $[\alpha \ \beta \ \gamma]^T$ ile gösterilir. Üst platformdaki her bir ekleme

yerleştirilen T_i koordinat sisteminin orijininin, T koordinat sisteminin orijinine göre konumlarını vektörel olarak aşağıdaki gibi gösterilir.

$$GT_i = \begin{bmatrix} GT_{xi} & GT_{yi} & GT_{zi} \end{bmatrix}^T \quad (5.1)$$

Denklem (5.1)'deki ifadeyi daha açık bir şekilde yazmak için Şekil 5.1'de gösterilen üst platformun üst görünümü aşağıdaki gibi yeniden çizilir.



Şekil 5.2. Tüm platformun (a) ve üst platformun eklem noktaları (b)

Bu durumda Denklem (5.1);

$$GT_i = \begin{bmatrix} GT_{xi} \\ GT_{yi} \\ GT_{zi} \end{bmatrix} = \begin{bmatrix} r_{up} \cos(\lambda_i) \\ r_{up} \sin(\lambda_i) \\ 0 \end{bmatrix} \quad (5.2)$$

şeklinde yazılır. Denklemde r_{up} üst platformun yarı çapıdır ve λ_i ise aşağıdaki gibi elde edilir.

$$\lambda_i = \frac{i\pi}{3} - \frac{\theta_p}{2} \quad i = 1, 3, 5 \quad (5.3)$$

$$\lambda_i = \lambda_{i-1} + \theta_p \quad i = 2, 4, 6$$

Denklemde θ_p her bir üst eklem noktasının diğer üst eklem noktasıyla yaptığı açığı ifade eder. Ayrıca, alt sabit platformun eklem noktalarındaki B_i koordinat sisteminin orijininin B - XYZ koordinat sistemine göre konumları Denklem (5.4)'deki gibi;

$$B_i = \begin{bmatrix} B_{xi} \\ B_{yi} \\ B_{zi} \end{bmatrix} = \begin{bmatrix} r_{base} \cos(v_i) \\ r_{base} \sin(v_i) \\ 0 \end{bmatrix} \quad (5.4)$$

şeklinde yazılır. Denklemde r_{base} alt platformun yarı çapını göstermektedir. Ayrıca v_i

$$v_i = \frac{i\pi}{3} - \frac{\theta_b}{2} \quad i = 1, 3, 5 \quad (5.5)$$

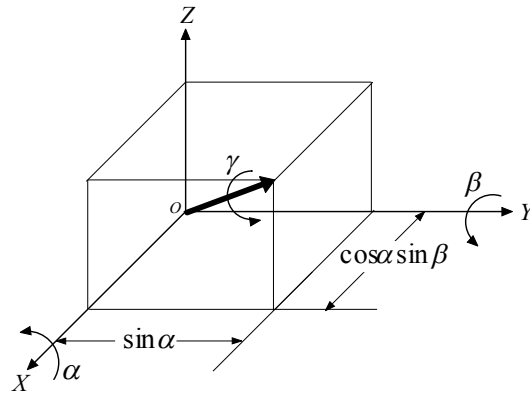
$$v_i = v_{i-1} + \theta_b \quad i = 2, 4, 6$$

şeklinde yazılır. Denklemde θ_b her bir alt eklem noktasının diğer alt eklem noktasıyla yaptığı açığı ifade eder.

Hareketli platformun sabit platforma göre yönelim matrisi R ise Şekil 5.3'te görüldüğü gibi sabit açı setiyle önce X eksenini boyunca α , daha sonra Y eksenini boyunca β ve son olarak Z eksenini boyunca γ açısıyla döndürülerek Denklem (5.6)'daki gibi elde edilir.

$$R_{XYZ}(\alpha, \beta, \gamma) = R_Z(\gamma)R_Y(\beta)R_X(\alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (5.6)$$

$$= \begin{bmatrix} \cos \beta \cos \gamma & \cos \gamma \sin \alpha \sin \beta - \cos \alpha \sin \gamma & \sin \alpha \sin \gamma + \cos \alpha \cos \gamma \sin \beta \\ \cos \beta \sin \gamma & \cos \alpha \cos \gamma + \sin \alpha \sin \beta \sin \gamma & \cos \alpha \sin \beta \sin \gamma - \cos \gamma \sin \alpha \\ -\sin \beta & \cos \beta \sin \alpha & \cos \alpha \cos \beta \end{bmatrix}$$



Şekil 5.3. Sabit açı setiyle dönme işlemi

Denklemden sırayla X , Y ve Z eksenlerindeki dönmeyi temsil eden $R_X(\alpha)$, $R_Y(\beta)$ ve $R_Z(\gamma)$ matrisleri aşağıdaki gibi verilmiştir.

$$R_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (5.7)$$

$$R_Y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (5.8)$$

$$R_Z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

Buradaki amaç, hareketli platformun sabit platforma göre yönelim matrisi (R) ve konum vektörü (P) verildiğinde bacakların uzunluklarının (l_i) bulunmasıdır. Bu işleme ters kinematik problem denir ve bacak vektörleri aşağıdaki denklemden yararlanılarak elde edilir.

$$L_i = R_{XYZ} GT_i + P - B_i \quad i=1,2, \dots, 6 \quad (5.10)$$

Sonuç olarak bacak uzunluğu bacak vektörünün normudur ($l_i = \|L_i\|$). Denklemden $P=[P_X \ P_Y \ P_Z]$ alt sabit platformun merkezine yerleştirilen B -XYZ koordinat sistemine göre üç boyutlu uzaydaki gidilmesi istenen konum vektörünü göstermektedir. Denklem (5.10)'un açık şekli aşağıda verildiği gibidir.

$$L_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} GT_{xi} \\ GT_{yi} \\ 0 \end{bmatrix} + \begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix} - \begin{bmatrix} B_{xi} \\ B_{yi} \\ 0 \end{bmatrix} \quad i=1,2, \dots, 6 \quad (5.11)$$

Denklemden GT_{xi} , GT_{yi} , B_{xi} ve B_{yi} platformun yapısından elde edilen sabit değişkenlerdir. Denklem (5.11)'deki işlemler yapıldığında;

$$L_i = \begin{bmatrix} P_X - B_{xi} + GT_{xi}r_{11} + GT_{yi}r_{12} \\ P_Y - B_{yi} + GT_{xi}r_{21} + GT_{yi}r_{22} \\ P_Z + GT_{xi}r_{31} + GT_{yi}r_{32} \end{bmatrix} \quad (5.12)$$

şeklinde olacaktır. Elde edilen vektörün uzunluğunu bulmak için, vektör bileşenlerinin kareleri alınıp toplanırsa,

$$l_i^2 = (P_X - B_{xi} + GT_{xi}r_{11} + GT_{yi}r_{12})^2 + (P_Y - B_{yi} + GT_{xi}r_{21} + GT_{yi}r_{22})^2 + (P_Z + GT_{xi}r_{31} + GT_{yi}r_{32})^2 \quad (5.13)$$

şeklinde elde edilir. Bu denklemden görüldüğü üzere konum vektörü ve yönelim matrisi elemanları kullanılarak kolaylıkla gerekli kol uzunlukları bulunur. Kartezyen uzayda hareketli platformun konumu ve yönelimi, konum vektörü ve yönelimini ifade eden açılar cinsinden aşağıdaki gibi ifade edilir.

$$X_{P-O} = [P_X \ P_Y \ P_Z \ \alpha \ \beta \ \gamma]^T \quad (5.14)$$

Genellikle Stewart Platform'larındaki arayüzlerde yukarıdaki vektör verilir. Bu vektörün açı elemanları Denklem (5.6)'da yerine konularak yönelim matrisi elemanları bulunur. Bu matris elemanları ve konum vektörü elemanları Denklem (5.13)'te yerine konularak kol uzunlukları elde edilir.

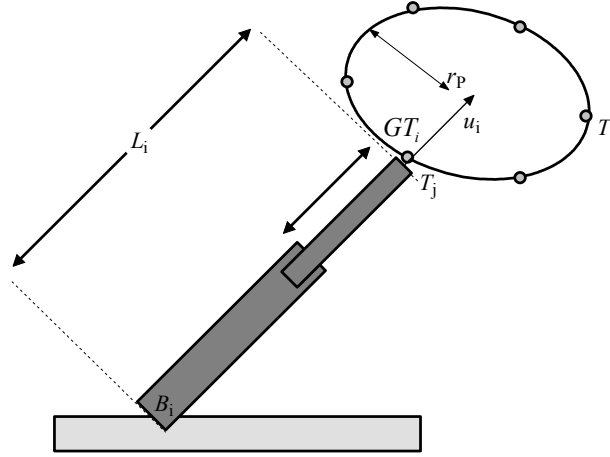
5.3. Jacobian Matrisinin Elde Edilmesi

5.3.1. Yöntem A

Şekil 5.4'te bir bacağa ait kinematik düzenleşim verilmiştir. GT_i , T_i koordinat sisteminin orijininin T koordinat sisteminin orijinine göre konumlarını gösteren vektördür. u_i ise sabit ve hareketli parçalardan oluşan bacağın doğrultusunu gösteren birim vektördür ve

$$\bar{u}_i = \frac{B_i T_j}{B_i T_j} \begin{cases} j = \frac{i+1}{2} & i, \text{ tek ise} \\ j = \frac{i}{2} & i, \text{ çift ise} \end{cases} \quad (5.15)$$

şeklinde ifade edilmektedir. Ayrıca $|L_i|$ ($i=1,2,\dots,6$) bacağın toplam uzunluğunu göstermektedir. Dikkat edileceği gibi L_i sabit ve hareketli kısımlardan oluştuğu için sürekli değişebilmektedir.



Şekil 5.4. Stewart Platform'unun bir bacağına ait kinematik düzenleşim

Bacaklarda oluşan hızlar J_{IA} ve J_{IIA} Jacobian matrisleri cinsinden ifade edilebilir. Bacakların hızları ile üst platformun hızları arasındaki ilişki Jacobian J_A matrisi cinsinden yazılabilir.

$$\dot{L} = J_A \dot{X}_{p-o} = J_{IA} J_{IIA} \dot{X}_{p-o} \quad (5.16)$$

Denklem (5.16)'da bulunan Jacobian matrisleri aşağıdaki denklemlerde gösterildiği gibi ifade edilebilir.

$$\dot{L} = J_{IA} \vec{V}_{T_j} \quad (5.17)$$

$$\vec{V}_{T_j} = J_{IIA} \dot{X}_{p-o} \quad (5.18)$$

\vec{V}_{T_j} bacakların uç hızlarını gösteren vektör, \dot{X}_{p-o} ise kartezyen uzayda üst platformun lineer ve açısal hızlarını gösteren vektördür. Denklemlerde yer alan J_{IA} ve J_{IIA} Jacobian matrisleri sırasıyla Denklem (5.19) ve (5.21)'de verilmiştir.

$$J_{IA} = \begin{bmatrix} \vec{u}_1^T & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \vec{u}_2^T & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \vec{u}_3^T & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \vec{u}_4^T & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & \vec{u}_5^T & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \vec{u}_6^T \end{bmatrix}_{6 \times 18} \quad (5.19)$$

Matriste verilen \vec{u}_i , bacakların doğrultusunu gösteren $\vec{u}_i = [u_{x_i} \quad u_{y_i} \quad u_{z_i}]^T$ vektörüdür ve Denklem (5.20)'deki gibi;

$$\vec{u}_i = \frac{R_{XYZ} GT_i + P - B_i}{|R_{XYZ} GT_i + P - B_i|} = \frac{L_i}{l_i} \quad (5.20)$$

şeklinde elde edilir. Denklem (5.16)'da yer alan J_{IIA} ifadesi aşağıda verildiği gibi hesaplanır.

$$J_{IIA} = \begin{bmatrix} I_{3 \times 3} & J_{IIA_{14}} & J_{IIA_{15}} & J_{IIA_{16}} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ I_{3 \times 3} & J_{IIA_{184}} & J_{IIA_{185}} & J_{IIA_{186}} \end{bmatrix}_{18 \times 6} \quad (5.21)$$

Matriste yer alan $I_{3 \times 3}$ birim matrisi göstermektedir. Diğer elemanların elde edilişi Bölüm 5.4.3'te verilen Denklem (5.83)'te gerekli düzeltmeler yapıp Denklem (5.86)'dan hesaplanır. Buna göre, Denklem (5.21)'in dördüncü sütunu aşağıdaki gibi elde edilir.

$$\begin{bmatrix} J_{IIA_{14}} & J_{IIA_{24}} & J_{IIA_{34}} \end{bmatrix}^T = R_Y(\beta)S(X)R_X(\alpha)R_Z(\gamma)GT_1 \quad (5.22)$$

$$\begin{bmatrix} J_{IIA_{44}} & J_{IIA_{54}} & J_{IIA_{64}} \end{bmatrix}^T = R_Y(\beta)S(X)R_X(\alpha)R_Z(\gamma)GT_2 \quad (5.23)$$

$$\begin{bmatrix} J_{IIA_{74}} & J_{IIA_{84}} & J_{IIA_{94}} \end{bmatrix}^T = R_Y(\beta)S(X)R_X(\alpha)R_Z(\gamma)GT_3 \quad (5.24)$$

$$\begin{bmatrix} J_{IIA_{104}} & J_{IIA_{114}} & J_{IIA_{124}} \end{bmatrix}^T = R_Y(\beta)S(X)R_X(\alpha)R_Z(\gamma)GT_4 \quad (5.25)$$

$$\begin{bmatrix} J_{IIA_{134}} & J_{IIA_{144}} & J_{IIA_{154}} \end{bmatrix}^T = R_Y(\beta)S(X)R_X(\alpha)R_Z(\gamma)GT_5 \quad (5.26)$$

$$\begin{bmatrix} J_{IIA_{164}} & J_{IIA_{174}} & J_{IIA_{184}} \end{bmatrix}^T = R_Y(\beta)S(X)R_X(\alpha)R_Z(\gamma)GT_6 \quad (5.27)$$

Denklem (5.21)'in beşinci sütunu aşağıdaki gibi elde edilir.

$$\begin{bmatrix} J_{IIA_{15}} & J_{IIA_{25}} & J_{IIA_{35}} \end{bmatrix}^T = S(Y)R_Y(\beta)R_X(\alpha)R_Z(\gamma)GT_1 \quad (5.28)$$

$$\begin{bmatrix} J_{IIA_{45}} & J_{IIA_{55}} & J_{IIA_{65}} \end{bmatrix}^T = S(Y)R_Y(\beta)R_X(\alpha)R_Z(\gamma)GT_2 \quad (5.29)$$

$$\begin{bmatrix} J_{IIA_{75}} & J_{IIA_{85}} & J_{IIA_{95}} \end{bmatrix}^T = S(Y)R_Y(\beta)R_X(\alpha)R_Z(\gamma)GT_3 \quad (5.30)$$

$$\begin{bmatrix} J_{IIA_{105}} & J_{IIA_{115}} & J_{IIA_{125}} \end{bmatrix}^T = S(Y)R_Y(\beta)R_X(\alpha)R_Z(\gamma)GT_4 \quad (5.31)$$

$$\begin{bmatrix} J_{IIA_{135}} & J_{IIA_{145}} & J_{IIA_{155}} \end{bmatrix}^T = S(Y)R_Y(\beta)R_X(\alpha)R_Z(\gamma)GT_5 \quad (5.32)$$

$$\begin{bmatrix} J_{IIA_{165}} & J_{IIA_{175}} & J_{IIA_{185}} \end{bmatrix}^T = S(Y)R_Y(\beta)R_X(\alpha)R_Z(\gamma)GT_6 \quad (5.33)$$

Denklem (5.21)'in altıncı sütunu aşağıdaki gibi elde edilir.

$$\begin{bmatrix} J_{IIA_{16}} & J_{IIA_{26}} & J_{IIA_{36}} \end{bmatrix}^T = R_Y(\beta)R_X(\alpha)S(Z)R_Z(\gamma)GT_1 \quad (5.34)$$

$$\begin{bmatrix} J_{IIA_{46}} & J_{IIA_{56}} & J_{IIA_{66}} \end{bmatrix}^T = R_Y(\beta)R_X(\alpha)S(Z)R_Z(\gamma)GT_2 \quad (5.35)$$

$$\begin{bmatrix} J_{IIA_{76}} & J_{IIA_{86}} & J_{IIA_{96}} \end{bmatrix}^T = R_Y(\beta)R_X(\alpha)S(Z)R_Z(\gamma)GT_3 \quad (5.36)$$

$$\begin{bmatrix} J_{IIA_{106}} & J_{IIA_{116}} & J_{IIA_{126}} \end{bmatrix}^T = R_Y(\beta)R_X(\alpha)S(Z)R_Z(\gamma)GT_4 \quad (5.37)$$

$$\begin{bmatrix} J_{IIA_{136}} & J_{IIA_{146}} & J_{IIA_{156}} \end{bmatrix}^T = R_Y(\beta)R_X(\alpha)S(Z)R_Z(\gamma)GT_5 \quad (5.38)$$

$$\begin{bmatrix} J_{IIA_{166}} & J_{IIA_{176}} & J_{IIA_{186}} \end{bmatrix}^T = R_Y(\beta)R_X(\alpha)S(Z)R_Z(\gamma)GT_6 \quad (5.39)$$

Denklemlerdeki S eksi bakışlıklı matristir (skew symmetric matrix). Eğer $a = [a_x, a_y, a_z]^T$ şeklinde üç vektörden oluşursa eksi bakışlıklı matris S(a) aşağıdaki gibi gösterilir.

$$S(a) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (5.40)$$

Eğer a , birim normal vektörler X , Y ve Z ifadelerinden oluşursa $S(X)$, $S(Y)$ ve $S(Z)$ eksi bakışimli matrisleri aşağıdaki gibi elde edilir.

$$S(X) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (5.41)$$

$$S(Y) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (5.42)$$

$$S(Z) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.43)$$

5.3.2. Yöntem B

Şekil 5.1’de gösterildiği gibi üst platformun orijinine yerleştirilen T -xyz koordinat sistemi ile sabit platformun orijinine yerleştirilen B -XYZ koordinat sistemi arasındaki eşleme, hareketli platformun yönelimini belirleyen (α, β, γ) sabit açı setini (x-y-z) kullanarak 3×3 ’lük bir rotasyon matrisi ile elde edilmiştir.

Şekil 5.4’e bakıldığında (T) koordinat sisteminde $GT_i = [GT_{xi} \quad GT_{yi} \quad GT_{zi}]^T$ olarak verilen hareketli platform üzerindeki $j.ci$ bağlantı noktasının koordinatları T_j , sabit platform (B) koordinat sistemine göre aşağıdaki gibi elde edilir.

$$T_j = [P_x \quad P_y \quad P_z]^T + {}^B R_T GT_i = x + {}^B R_T GT_i \quad (5.44)$$

T_j bağlantı noktasının konumu elde edildiğine göre $i.ci$ bağ vektörü L_i aşağıdaki gibi elde edilir.

$$L_i = \vec{T}_j - \vec{B}_i = B_i \vec{T}_j \quad (5.45)$$

Böylece Denklem (5.20)'de elde edilen prizmatik eklem doğrultusundaki birim vektörü (u_i) genel olarak aşağıda verildiği gibi i . bacak vektörünün bacak uzunluğuna bölünmesiyle de elde edilebilir.

$$\vec{u}_i = \frac{B_i \vec{T}_j}{|L_i|} = \frac{L_i}{l_i} \quad (5.46)$$

T_j bağlantı noktasının hızı, Denklem (5.44)'ün zamana göre türevi alınarak elde edilir.

$$\dot{T}_j = \vec{V}_{T_j} = \begin{bmatrix} \dot{P}_x & \dot{P}_y & \dot{P}_z \end{bmatrix}^T + \omega \times {}^B R_T G T_i = \dot{x} + \omega \times {}^B R_T G T_i \quad (5.47)$$

Yukarıdaki denklemde, alt platform koordinat sistemine göre hareketli platform üzerinde bir noktayı gösteren konum vektörünün zaman göre türevi, o konum vektörü ile üst platform açısız hızının (ω) vektörel çarpımı yapılarak elde edilir. Bölüm 5.4.1'de detaylı olarak çıkarımı verilen açısız hız aşağıda verilmiştir.

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & 0 \\ 0 & 1 & -\sin \alpha \\ -\sin \beta & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \quad (5.48)$$

$G T_i$ sabit olduğundan dolayı yönelim matrisinin zamana göre türevi Denklem (5.49)'daki gibi;

$${}^B \dot{R}_T = \omega \times {}^B R_T \quad (5.49)$$

şeklinde ifade edilir. $i.ci$ prizmatik eklem eksenindeki bu hız vektörünün (\dot{T}_j) izdüşümü, o bağı uzama hızını diğer bir deyişle bacak hızını verir.

$$\dot{L}_i = \dot{T}_j \cdot \vec{u}_i = \begin{bmatrix} \dot{P}_x & \dot{P}_y & \dot{P}_z \end{bmatrix}^T \cdot \vec{u}_i + \omega \times ({}^B R_T G T_i) \cdot \vec{u}_i = \dot{x} \cdot \vec{u}_i + \omega \times ({}^B R_T G T_i) \cdot \vec{u}_i \quad (5.50)$$

Yukarıdaki denklemde üçlü içsel çarpım ($a \times b \cdot c$) için vektörlerin yerlerini

değiştirmeden $a \cdot bxc$ şeklinde yazılabilir. Bu denklemi kullanarak Denklem (5.50) tekrar düzenlenirse,

$$\dot{L}_i = \dot{x} \cdot \vec{u}_i + \omega \cdot \left({}^B R_T GT_i \right) \times \vec{u}_i \quad (5.51)$$

şeklinde elde edilir. Yukarıdaki denklem, 6 eklem için matris formunda yazılırsa aşağıdaki gibi elde edilir.

$$\dot{L} = J_B \begin{bmatrix} \dot{x} \\ \omega \end{bmatrix} = J_B \dot{X}_{P-O} = J_{IB} J_{IIB} \dot{X}_{P-O} \quad (5.52)$$

Yukarıdaki denklemde yer alan J_{IB} ve J_{IIB} sırasıyla Denklem (5.53) ve (5.54)'deki gibi;

$$J_{IB} = \begin{bmatrix} u_{x1} & u_{y1} & u_{z1} & \left({}^B R_T GT_1 x \vec{u}_1 \right)^T \\ \vdots & \vdots & \vdots & \vdots \\ u_{x6} & u_{y6} & u_{z6} & \left({}^B R_T GT_6 x \vec{u}_6 \right)^T \end{bmatrix}_{6 \times 6} \quad (5.53)$$

$$J_{IIB} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \beta & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -\sin \alpha \\ 0 & 0 & 0 & -\sin \beta & 0 & \cos \alpha \end{bmatrix}_{6 \times 6} \quad (5.54)$$

şeklinde verilmiştir. Bu bölümde iki farklı yöntemle göre çıkarılan J_A ve J_B Jacobian matrislerin MATLAB ortamında yazılan kodları, Ek-A'da Jacobian_A ve Jacobian_B isimli başlıklar altında verilmiştir.

5.4. Dinamik Denklemlerin Çıkarımı

Bu bölümde, Lagrange formülasyonu kullanılarak Stewart Platform'unun genel bir sınıfı için katı gövde dinamik denklemleri elde edilecektir. Burada, hareketli platformun ve bacakların dinamik modelleri ayrı ayrı hesaplandıktan sonra elde edilen bu katı gövde dinamik modeline eyleyici dinamiği eklenecektir. Dinamik

denklemleri çıkarabilmek için Stewart Platform'u, üst platform ve altı adet bacadan oluşan iki ana kısma ayrılacaktır. Bu iki ana kısmın kinetik ve potansiyel enerjileri hesaplanıp bunlardan yararlanılarak dinamik denklemler çıkarılacaktır. Ayrıca, eyleyicilerin elektriksel ve mekaniksel dinamik denklemleri hesaplanarak sistemin tüm dinamiği elde edilecektir.

5.4.1. Üst platformun kinetik enerjisi

Üst platformun kinetik enerjisi öteleme ve dönme hareketlerinden kaynaklanan iki farklı kinetik enerji ifadesinden oluşmaktadır. Üst platformun kütle merkezinin öteleme hareketinden dolayı meydana getirdiği kinetik enerji,

$$K_{up(trans)} = \frac{1}{2} m_{up} (\dot{P}_X^2 + \dot{P}_Y^2 + \dot{P}_Z^2) \quad (5.55)$$

Denklemden m_{up} üst platformun kütle merkezini göstermektedir. Üst platformun kütle merkezinin dönme hareketinden dolayı meydana getirdiği kinetik enerji,

$$K_{up(rot)} = \frac{1}{2} \vec{\Omega}_{up(mf)}^T I_{(mf)} \vec{\Omega}_{up(mf)} \quad (5.56)$$

denklemden $I_{(mf)}$ ve $\vec{\Omega}_{up(mf)}$ sırasıyla üst hareketli platformun atalet tensörünü ve açısal hızını göstermektedir ve basit olarak aşağıdaki gibi ifade edilebilirler.

$$I_{(mf)} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (5.57)$$

$$\vec{\Omega}_{up(mf)} = R_Z(\gamma)^T R_X(\alpha)^T R_Y(\beta)^T \vec{\Omega}_{up(ff)} \quad (5.58)$$

Denklemden $\vec{\Omega}_{up(ff)}$ üst platformun sabit alt platform cinsinden açısal hızını göstermektedir ve sabit açı setini oluşturan α , β ve γ açılarından faydalanarak aşağıdaki gibi yazılır.

$$\vec{\Omega}_{up(ff)} = \dot{\alpha} R_Y(\beta) \vec{X} + \dot{\beta} \vec{Y} + \dot{\gamma} R_X(\alpha) R_Z(\gamma) \vec{Z} \quad (5.59)$$

Denklem (5.59)'u açık bir şekilde yazılırsa aşağıdaki gibi ifade edilir.

$$\begin{aligned}
\bar{\Omega}_{up(ff)} &= \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \\
& \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \\
&= \begin{bmatrix} \cos \beta & 0 & 0 \\ 0 & 1 & -\sin \alpha \\ -\sin \beta & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}
\end{aligned} \tag{5.60}$$

Bu durumda üst platformun açısal hızı aşağıdaki gibi elde edilir.

$$\begin{aligned}
\bar{\Omega}_{up(mf)} &= R_Z(\gamma)^T R_X(\alpha)^T R_Y(\beta)^T \bar{\Omega}_{up(ff)} \\
&= \begin{bmatrix} c\gamma & s\gamma & 0 \\ -s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & s\alpha \\ 0 & -s\alpha & c\alpha \end{bmatrix} \begin{bmatrix} c\beta & 0 & -s\beta \\ 0 & 1 & 0 \\ s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} c\beta & 0 & 0 \\ 0 & 1 & -s\alpha \\ -s\beta & 0 & c\alpha \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \\
&= \begin{bmatrix} c\gamma & cas\gamma & -cac\gamma s\beta - cas\alpha s\gamma + cac\beta s\alpha s\gamma \\ -s\gamma & cac\gamma & -cac\gamma s\alpha + cas\beta s\gamma + cac\beta s\alpha c\gamma \\ 0 & -s\alpha & s^2\alpha + c^2\alpha c\beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}
\end{aligned} \tag{5.61}$$

Denklemden yer alan $s(\cdot) = \sin(\cdot)$ ve $c(\cdot) = \cos(\cdot)$ şeklindeki ifadelerdir. Yukarıdaki ifadeden faydalanarak üst platformun kütle merkezinin dönme hareketinden dolayı meydana getirdiği kinetik enerji aşağıdaki gibi bulunur.

$$\begin{aligned}
K_{up(rot)} &= \frac{1}{2} \bar{\Omega}_{up(mf)}^T I_{(mf)} \bar{\Omega}_{up(mf)} \\
&= \frac{1}{2} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}^T \begin{bmatrix} I_x c^2\gamma + I_y s^2\gamma & (I_x - I_y) cac\gamma s\gamma & k_{13} \\ (I_x - I_y) cac\gamma s\gamma & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}
\end{aligned} \tag{5.62}$$

Denklemden yer alan k_{13}, k_{22}, k_{23} ve k_{33} değişkenlerine ait ifadeler aşağıdaki denklemlerde verildiği gibidir. $k_{31} = k_{13}, k_{32} = k_{23}$

$$\begin{aligned}
k_{13} &= c\gamma I_x (-s\gamma cas\alpha - c\gamma s\beta c\alpha + s\gamma cas\alpha c\beta) - s\gamma I_y (-cac\gamma s\alpha + s\gamma s\beta c\alpha \\
& \quad + cac\gamma s\alpha c\beta)
\end{aligned} \tag{5.63}$$

$$k_{22} = c^2\alpha(I_x s^2\gamma + I_y c^2\gamma) + I_z s^2\alpha \quad (5.64)$$

$$k_{23} = s\gamma c\alpha I_x (-s\gamma c\alpha s\alpha - c\gamma s\beta c\alpha + s\gamma c\alpha s\alpha c\beta) + c\gamma c\alpha I_y (-c\alpha c\gamma s\alpha + s\gamma s\beta c\alpha + c\alpha c\gamma s\alpha c\beta) - s\alpha I_z (s^2\alpha + c^2\alpha c\beta) \quad (5.65)$$

$$k_{33} = I_x (-s\gamma c\alpha s\alpha - c\gamma s\beta c\alpha + s\gamma c\alpha s\alpha c\beta)^2 + I_y (-c\alpha c\gamma s\alpha + s\gamma s\beta c\alpha + c\alpha c\gamma s\alpha c\beta)^2 + I_z (s^2\alpha + c^2\alpha c\beta) \quad (5.66)$$

Sonuç olarak üst platformun toplam kinetik enerjisi aşağıdaki gibi bulunur.

$$K_{up} = K_{up(trans)} + K_{up(rot)} = \frac{1}{2} m_{up} (\dot{P}_X^2 + \dot{P}_Y^2 + \dot{P}_Z^2) + \frac{1}{2} \bar{\Omega}_{up(mf)}^T I_{(mf)} \bar{\Omega}_{up(mf)} \quad (5.67)$$

$$= \frac{1}{2} \dot{X}_{P-O}^T \cdot M_{up}(X_{P-O}) \cdot \dot{X}_{P-O} = \frac{1}{2} \begin{bmatrix} \dot{P}_X & \dot{P}_Y & \dot{P}_Z & \dot{\alpha} & \dot{\beta} & \dot{\gamma} \end{bmatrix} M_{up} \begin{bmatrix} \dot{P}_X \\ \dot{P}_Y \\ \dot{P}_Z \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

Denklemden M_{up} üst platformun kütle matrisini göstermektedir. Aşağıda verildiği gibi elde edilir.

$$M_{up} = \begin{bmatrix} m_{up} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{up} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{up} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x c^2\gamma + I_y s^2\gamma & (I_x - I_y) c\alpha c\gamma s\gamma & k_{13} \\ 0 & 0 & 0 & (I_x - I_y) c\alpha c\gamma s\gamma & k_{22} & k_{23} \\ 0 & 0 & 0 & k_{31} & k_{32} & k_{33} \end{bmatrix} \quad (5.68)$$

Yukarıda elde edilen üst platformun kütle matrisinin MATLAB ortamında yazılan kodları, Ek-A'da UstPlatformKutleMatrisi isimli başlık altında verilmiştir.

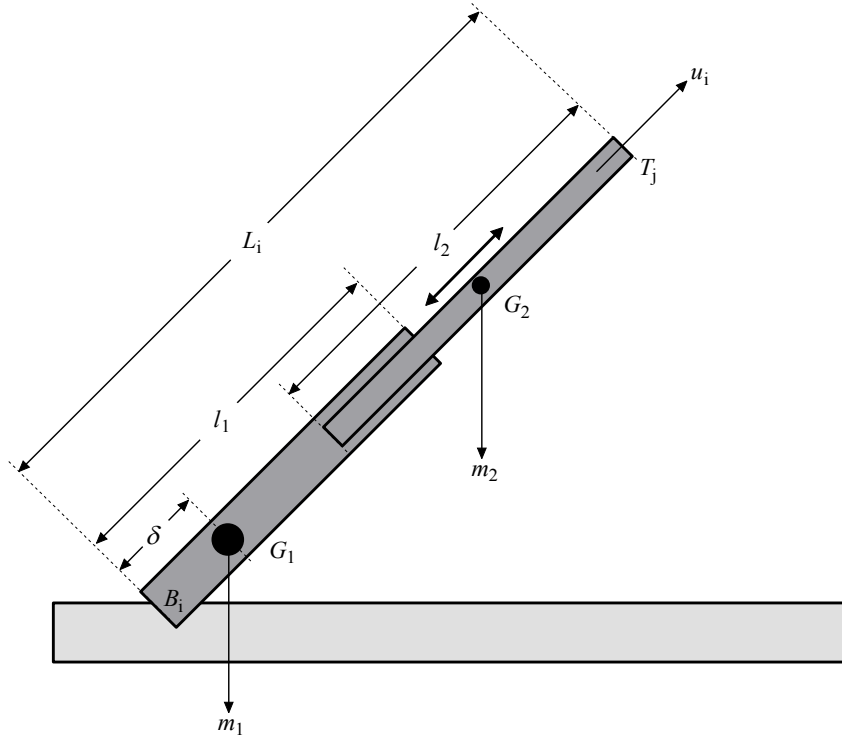
5.4.2. Üst platformun potansiyel enerjisi

Üst platformun toplam potansiyel enerjisi ise aşağıdaki gibi bulunur.

$$P_{up} = m_{up} g P_z = \begin{bmatrix} 0 & 0 & m_{up} g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_X & P_Y & P_Z & \alpha & \beta & \gamma \end{bmatrix}^T \quad (5.69)$$

5.4.3. Bacakların kinetik enerjisi

Her bir bacağın kinetik ve potansiyel enerjilerinin hesaplanması için bacaklar sabit ve hareketli olmak üzere iki ayrı parçaya bölünür. Şekil 5.5'te bir bacağına ait kinematik ve dinamik düzenleşim verilmiştir. Şekil 5.5'te G_{1i} sabit l_1 uzunluğundaki parçanın kütle merkezini göstermektedir. Dikkat edileceği gibi ağır DC motorlar bu parçaya monte edildiğinden kütle merkezi, kütle dağılımı homojen olmayan l_1 uzunluğunun tam ortasında değildir. Sabit l_1 parçasının üzerinde B_1 ile G_{1i} arasındaki mesafe δ olarak gösterilmiştir. Sabit parçanın aksine hareketli parçanın kütle dağılımı homojendir. Dolayısıyla kütle merkezi G_{2i} , l_2 uzunluğunun tam ortasındadır. Şekil 5.5'te ayrıca m_1 ve m_2 sabit ve hareketli parçaların kütlelerini, u_i ise bu iki parçadan oluşan bacağın doğrultusu gösteren bir vektördür



Şekil 5.5. Stewart Platform'unda bir bacağın kinematik ve dinamik düzenleşimi

Bir bacağın toplam potansiyel ve kinetik enerjini bulmak için öncelikle bu bacağın kütle merkezinin (G_i) konum ve hızının bulunması gereklidir.

$$\overrightarrow{B_i G_i} = \frac{1}{m_1 + m_2} \left[\left(\delta m_1 l_1 - \frac{1}{2} m_2 l_2 \right) + m_2 L_i \right] \vec{u}_i \quad (5.70)$$

Yukarıdaki denklem aşağıdaki gibi yazılabilir.

$$\overrightarrow{B_i G_i} = \left[\hat{I} + \frac{m_2}{m_1 + m_2} L_i \right] \vec{u}_i \quad (5.71)$$

Denklemdaki \hat{I} aşağıdaki gibi ifade edilir.

$$\hat{I} = \frac{1}{m_1 + m_2} \left(\delta m_1 l_1 - \frac{1}{2} m_2 l_2 \right) \quad (5.72)$$

(G_i) kütle merkezinin hızı aşağıdaki gibi yazılabilir.

$$\vec{V}_{G_i} = \frac{d\overrightarrow{OG}}{dt} = \frac{d\overrightarrow{B_i G_i}}{dt} = \hat{I} \frac{d\vec{u}_i}{dt} + \frac{m_2}{m_1 + m_2} \frac{d\overrightarrow{O_i T_j}}{dt} \quad (5.73)$$

denklemden $L_i \vec{u}_i = \frac{L_i \overrightarrow{O_i T_j}}{O_i T_j (=L_i)}$ ise $L_i \vec{u}_i = \overrightarrow{O_i T_j}$ olarak yazılır. Burada \vec{u}_i ifadesinin

türevi ise

$$\begin{aligned} \frac{d\vec{u}_i}{dt} &= \frac{d(\overrightarrow{B_i T_j} / L_i)}{dt} = -\frac{1}{L_i^2} \frac{dL_i}{dt} \overrightarrow{B_i T_j} + \frac{1}{L_i} \vec{V}_{T_j} = -\frac{1}{L_i} \frac{dL_i}{dt} \vec{u}_i + \frac{1}{L_i} \vec{V}_{T_j} \\ &= -\frac{1}{L_i} (\vec{V}_{T_j} \cdot \vec{u}_i) \vec{u}_i + \frac{1}{L_i} \vec{V}_{T_j} \end{aligned} \quad (5.74)$$

Yukarıdaki denklemde bölümün türevi alınmıştır ve ayrıca $\frac{d(\overrightarrow{O_i T_j})}{dt} = \vec{V}_{T_j}$ ve

$\vec{u}_i = \frac{\overrightarrow{B_i T_j}}{L_i} = \frac{\overrightarrow{O_i T_j}}{L_i}$ dir. Sonuç olarak Denklem (5.74)'teki hız ifadesi,

$$\vec{V}_{G_i} = \frac{\hat{I}}{L_i} \left[\vec{V}_{T_j} - (\vec{V}_{T_j} \cdot \vec{u}_i) \vec{u}_i \right] + \frac{m_2}{m_1 + m_2} \vec{V}_{T_j} \quad (5.75)$$

şeklinde elde edilir. Yukarıdaki denklemde \vec{V}_{G_i} hız ifadesini kullanarak bacakların kinetik enerjisi aşağıdaki gibi yazılabilir.

$$\begin{aligned}
K_{Li} &= \frac{1}{2}(m_1 + m_2)\vec{V}_{Gi}^T \vec{V}_{Gi} \\
&= \frac{1}{2}(m_1 + m_2) \left[h_i (\vec{V}_{Tj}^T \vec{V}_{Tj} - \vec{V}_{Tj}^T \vec{u}_i \vec{u}_i^T \vec{V}_{Tj}) + \left(\frac{m_2}{m_1 + m_2} \right)^2 \vec{V}_{Tj}^T \vec{u}_i \vec{u}_i^T \vec{V}_{Tj} \right]
\end{aligned} \tag{5.76}$$

Denkleminde yer alan h_i ifadesi aşağıdaki gibidir.

$$h_i = \left(\frac{\hat{I}}{L_i} + \frac{m_2}{m_1 + m_2} \right)^2 \tag{5.77}$$

Denklem (5.76)'daki ifadenin bir kısmı dönel bir kısmı da ötelemeden kaynaklanan kinetik enerjiyi göstermektedir. Bunlar, Denklem (5.78) ve (5.79)'daki gibi;

$$K_{Li(rot)} = \frac{1}{2}(m_1 + m_2) \left[h_i (\vec{V}_{Tj}^T \vec{V}_{Tj} - \vec{V}_{Tj}^T \vec{u}_i \vec{u}_i^T \vec{V}_{Tj}) \right] \tag{5.78}$$

$$K_{Li(trans)} = \frac{1}{2}(m_1 + m_2) \left[\left(\frac{m_2}{m_1 + m_2} \right)^2 \vec{V}_{Tj}^T \vec{u}_i \vec{u}_i^T \vec{V}_{Tj} \right] \tag{5.79}$$

şeklinde yazılabilir. Hatırlanacağı gibi \vec{u}_i bacak yönünü gösteren birim vektör ve \vec{V}_{Tj} 'de bacakların uç hızlarını gösteren vektör olduğundan

$$\vec{V}_{Tj}^T \vec{u}_i = \frac{dL_i}{dt} = \dot{L}_i \tag{5.80}$$

şeklinde yazılabilir. Bu durumda Denklem (5.76);

$$\begin{aligned}
K_{Li} &= \frac{1}{2}(m_1 + m_2) \left[\vec{V}_{Tj}^T h_i \vec{V}_{Tj} - h_i \dot{L}_i \dot{L}_i + \left(\frac{m_2}{m_1 + m_2} \right)^2 \dot{L}_i \dot{L}_i \right] \\
&= \frac{1}{2}(m_1 + m_2) \left[\vec{V}_{Tj}^T h_i \vec{V}_{Tj} - \dot{L}_i \dot{L}_i \left(h_i - \left(\frac{m_2}{m_1 + m_2} \right)^2 \right) \right] \\
&= \frac{1}{2}(m_1 + m_2) \left[\vec{V}_{Tj}^T h_i \vec{V}_{Tj} - \dot{L}_i k_i \dot{L}_i \right]
\end{aligned} \tag{5.81}$$

gibi daha sade bir şekilde yazılabilir. Denkleminde yer alan k_i aşağıda verildiği gibidir.

$$k_i = \frac{\hat{I}}{L_i} \left(\frac{\hat{I}}{L_i} + \frac{m_2}{m_1 + m_2} \right) = h_i - \left(\frac{m_2}{m_1 + m_2} \right)^2 \quad (5.82)$$

Sonuç olarak Denklem (5.81) altı bacağında kinetik enerjisini verecek şekilde aşağıdaki gibi yazılır.

$$K_{Legs} = \sum_1^6 K_{Li} = \frac{1}{2} (m_1 + m_2) \left[\begin{array}{c} \vec{V}_{T1} \\ \vec{V}_{T2} \\ \vec{V}_{T3} \\ \vec{V}_{T4} \\ \vec{V}_{T5} \\ \vec{V}_{T6} \end{array} \right]^T H \left[\begin{array}{c} \vec{V}_{T1} \\ \vec{V}_{T2} \\ \vec{V}_{T3} \\ \vec{V}_{T4} \\ \vec{V}_{T5} \\ \vec{V}_{T6} \end{array} \right] - \left[\begin{array}{c} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \\ \dot{L}_4 \\ \dot{L}_5 \\ \dot{L}_6 \end{array} \right]^T K \left[\begin{array}{c} \dot{L}_1 \\ \dot{L}_2 \\ \dot{L}_3 \\ \dot{L}_4 \\ \dot{L}_5 \\ \dot{L}_6 \end{array} \right] \quad (5.83)$$

Denklemden H ve K matrisleri aşağıda verilmiştir.

$$H = \left[\begin{array}{cccccccc} h_1 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & h_1 & 0 & 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & h_1 & 0 & 0 & 0 & \dots & \dots & 0 \\ \vdots & \dots & \dots & \ddots & \dots & \dots & \dots & \dots & 0 \\ \vdots & \dots & \dots & \dots & \ddots & \dots & \dots & \dots & 0 \\ \vdots & \dots & \dots & \dots & \dots & \ddots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & h_6 & 0 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & h_6 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 0 & h_6 \end{array} \right]_{18 \times 18} \quad (5.84)$$

$$K = \left[\begin{array}{cccccc} k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_6 \end{array} \right]_{6 \times 6} \quad (5.85)$$

Bölüm 5.3'te iki farklı yöntem (Yöntem A ve Yöntem B) göre çıkarılan Jacobian matrisleri, bacakların hızları ile üst platformun hızları arasındaki ilişkiyi vermektedir.

Buna göre Yöntem A için elde edilen $\dot{L}_i = J_{iA} \vec{V}_{Ti}$ ve $\vec{V}_{Ti} = J_{iA} \dot{X}_{P-O}$ hız ifadeleri

kullanılarak Denklem (5.83)'te yazılıp gerekli düzenlemeler yapılırsa bacakların kinetik enerjisi, Denklem (5.86)'daki gibi;

$$K_{Legs} = \frac{1}{2} \dot{X}_{P-O}^T M_{Legs} (X_{P-O}) \dot{X}_{P-O} \quad (5.86)$$

şeklinde olacaktır. Yukarıdaki denklemde yer alan M_{Legs} ifadesi Yöntem A'ya göre aşağıdaki gibi elde edilir.

$${}^A M_{Legs} = (m_1 + m_2) \left[(J_{IIA})^T \left[H - (J_{IA})^T K J_{IA} \right] J_{IIA} \right] \quad (5.87)$$

Benzer şekilde Yöntem B için yine Bölüm 5.3'te elde edilen J_{IB} ve J_{IIB} Jacobian matrisleri kullanılarak Denklem (5.83) ifadesinde yerine konulduğunda bacaklara ait kütle matrisi aşağıdaki gibi bulunur.

$${}^B M_{Legs} = (m_1 + m_2) \left[(J_{IIB})^T \left[H - (J_{IB})^T K J_{IB} \right] J_{IIB} \right] \quad (5.88)$$

Yukarıda hem Yöntem A'ya göre hem de Yöntem B'ye göre elde edilen bacaklara ait kütle matrisinin MATLAB ortamında yazılan kodları, EK-A'da BacakKutleMatrisi_A ve BacakKutleMatrisi_B isimler altında verilmiştir.

5.4.4. Bacakların potansiyel enerjisi

Bacakların potansiyel enerjisi aşağıdaki gibi bulunur.

$$P_{Legs} = (m_1 + m_2) g \sum_1^6 (\overrightarrow{B_i G_i} \cdot \vec{Z}) = (m_1 + m_2) g \sum_1^6 \left[\hat{I} + \frac{m_2}{m_1 + m_2} L_i \right] (\vec{u}_i \cdot \vec{Z}) \quad (5.89)$$

Daha öncede belirtildiği gibi $\vec{u}_i = \frac{\overrightarrow{B_i T_j}}{L_i}$ olduğundan

$$\vec{u}_i \cdot \vec{Z} = \frac{1}{L_i} Z_{T_j} \quad (5.90)$$

şeklinde yazılabilir. Denklemde Z_{T_j} aşağıdaki gibi hesaplanır.

$$Z_{T_j} = p_z + \vec{Z} \cdot \vec{GT}_{j(ff)} \quad \begin{cases} j = \frac{i+1}{2} & i \text{ tek} \\ j = \frac{i}{2} & i \text{ çift} \end{cases} \quad (5.91)$$

Sonuç olarak bacakların potansiyel enerjisini veren denklem aşağıdaki gibi elde edilir.

$$P_{Legs} = (m_1 + m_2)g \sum_{i=1}^3 \left[\hat{I} \left(\frac{1}{L_{2i}} + \frac{1}{L_{2i-1}} \right) + \frac{2m_2}{m_1 + m_2} \right] (p_z + Z_{T_j}) \quad (5.92)$$

5.4.5. Dinamik denklemlerin yazılımı

Genelleştirilmiş koordinatları q ve genelleştirilmiş kuvvetleri de τ olarak düşünüldüğünde Lagrange formülasyonuna göre hareket denklemleri aşağıda verildiği gibi elde edilir.

$$\frac{d}{dt} \frac{dL}{dq} - \frac{\partial L}{\partial q} = \frac{d}{dt} \left(\frac{\partial K(q, \dot{q})}{\partial \dot{q}} \right) - \frac{\partial K(q, \dot{q})}{\partial q} + \frac{\partial P(q)}{\partial q} = \tau \quad (5.93)$$

Denklemden yer alan $K(q, \dot{q})$ sistemin toplam kinetik enerjisini $P(q)$ ise sistemin toplam potansiyel enerjisini göstermektedir. Stewart Platform'un hareketli platform ve bacaklarına ait kinetik ve potansiyel enerji denklemleri elde edildikten sonra Lagrange denkleminde yerine yazılarak Stewart Platform'una ait dinamik denklemler elde edilir. Buna göre, q ile X_{P-O} yer değiştirildiğinde dinamik denklemler Kartezyen uzayda aşağıdaki gibi ifade edilebilir.

$$J^T(X_{P-O})F = M(X_{P-O})\ddot{X}_{P-O} + V_m(X_{P-O}, \dot{X}_{P-O})\dot{X}_{P-O} + G(X_{P-O}) \quad (5.94)$$

Denklemden $M(X_{P-O})$, üst platformun ve bacakların kütle matrisini, $V_m(X_{P-O}, \dot{X}_{P-O})$, Coriolis ve merkezkaç matrisini, $G(X_{P-O})$, yerçekimi kuvvet vektörünü ve son olarak $F = [f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6]$ 'dir. f_i ise i . bacağın eyleyicisi tarafından uygulanan kuvveti göstermektedir. Jacobian matrisleri farklı yöntemlerle çıkarıldığından aşağıda hem Yöntem A hem de Yöntem B'ye göre

toplam kütle matrisi, toplam Coriolis ve merkezkaç matrisi ve toplam yer çekimi kuvvet vektörü aşağıda gösterildiği gibi elde edilir.

$${}^A M(X_{p-o}) = M_{up} + {}^A M_{Legs}, \quad {}^B M(X_{p-o}) = M_{up} + {}^B M_{Legs} \quad (5.95)$$

$${}^A V_m(X_{p-o}, \dot{X}_{p-o}) = V_{mup} + {}^A V_{mLegs}, \quad {}^B V_m(X_{p-o}, \dot{X}_{p-o}) = V_{mup} + {}^B V_{mLegs} \quad (5.96)$$

$${}^A G(X_{p-o}) = G_{up} + {}^A G_{Legs}, \quad {}^B G(X_{p-o}) = G_{up} + {}^B G_{Legs} \quad (5.97)$$

Denklem (5.96)'da gösterilen $V_{mup} + {}^A V_{mLegs}$ ve $V_{mup} + {}^B V_{mLegs}$ ifadeleri sırasıyla aşağıdaki gibi bulunur.

$$V_{mup(i,j)} = \frac{1}{2} \sum_{k=1}^6 \left(\frac{\partial M_{up}(k,j)}{\partial X_{p-o}(i)} + \frac{\partial M_{up}(k,i)}{\partial X_{p-o}(j)} - \frac{\partial M_{up}(i,j)}{\partial X_{p-o}(k)} \right) \dot{X}_{p-o}(k) \quad (5.98)$$

$${}^A V_{Legs(i,j)} = \frac{1}{2} \sum_{k=1}^6 \left(\frac{\partial {}^A M_{Legs}(k,j)}{\partial X_{p-o}(i)} + \frac{\partial {}^A M_{Legs}(k,i)}{\partial X_{p-o}(j)} - \frac{\partial {}^A M_{Legs}(i,j)}{\partial X_{p-o}(k)} \right) \dot{X}_{p-o}(k) \quad (5.99)$$

$${}^B V_{Legs(i,j)} = \frac{1}{2} \sum_{k=1}^6 \left(\frac{\partial {}^B M_{Legs}(k,j)}{\partial X_{p-o}(i)} + \frac{\partial {}^B M_{Legs}(k,i)}{\partial X_{p-o}(j)} - \frac{\partial {}^B M_{Legs}(i,j)}{\partial X_{p-o}(k)} \right) \dot{X}_{p-o}(k) \quad (5.100)$$

Son olarak, Denklem (5.97)'de gösterilen $G_{up} + {}^A G_{Legs}$ ve $G_{up} + {}^B G_{Legs}$ ifadeleri de genel olarak sırasıyla aşağıdaki gibi bulunur.

$$G_{up}(k) = \frac{\partial P_{up}(X_{p-o})}{\partial X_{p-o}(k)} \quad (5.101)$$

$$\begin{aligned} {}^A G_{Legs}(k) &= \frac{\partial {}^A P_{Legs}(X_{p-o})}{\partial X_{p-o}(k)} \\ &= (m_1 + m_2) g \sum_{i=1}^3 \left[\hat{I} \left[\frac{1}{{}^A L_{2i}^2} \left(\frac{\partial {}^A L_{2i}}{\partial X_{p-o}(k)} \right) + \frac{1}{{}^A L_{2i-1}^2} \left(\frac{\partial {}^A L_{2i-1}}{\partial X_{p-o}(k)} \right) \right] \right] (p_z + Z_{T_i}) \\ &\quad + (m_1 + m_2) g \sum_{i=1}^3 \left[\hat{I} \left(\frac{1}{{}^A L_{2i}} + \frac{1}{{}^A L_{2i-1}} \right) + \frac{2m_2}{(m_1 + m_2)} \right] (p_z + Z_{T_i}) \end{aligned} \quad (5.102)$$

$$\begin{aligned}
{}^B G_{Legs}(k) &= \frac{\partial^B P_{Legs}(X_{p-o})}{\partial X_{p-o}(k)} \\
&= (m_1 + m_2)g \sum_{i=1}^3 \left[\hat{I} \left[\frac{1}{{}^B L_{2i}^2} \left(\frac{\partial^B L_{2i}}{\partial X_{p-o}(k)} \right) + \frac{1}{{}^B L_{2i-1}^2} \left(\frac{\partial^B L_{2i-1}}{\partial X_{p-o}(k)} \right) \right] \right] (p_z + Z_{T_i}) \quad (5.103) \\
&\quad + (m_1 + m_2)g \sum_{i=1}^3 \left[\hat{I} \left(\frac{1}{{}^B L_{2i}} + \frac{1}{{}^B L_{2i-1}} \right) + \frac{2m_2}{(m_1 + m_2)} \right] (p_z + Z_{T_i})
\end{aligned}$$

Yukarıdaki denklemlerde yer alan $\left(\frac{\partial L_n}{\partial X_{p-o}(k)} \right)$ ifadesi, Jacobian matrislerinden

faydalanılarak her iki metot için aşağıda verildiği gibi elde edilir.

$$\frac{\partial^A L_n}{\partial X_{p-o}(k)} = \sum_{m=1}^9 J_{IA_{nm}} J_{IIA_{mk}}, \quad \frac{\partial^B L_n}{\partial X_{p-o}(k)} = \sum_{m=1}^6 J_{IB_{nm}} J_{IIB_{mk}} \quad (5.104)$$

Yukarıda hem Yöntem A'ya göre hem de Yöntem B'ye göre elde edilen bacaklara ait Corilois ve merkezkaç matrislerinin ve yer çekimi kuvvet vektörlerinin MATLAB ortamında yazılan kodları, EK-A'da BacakCoriloisMatrisi_A, BacakCoriloisMatrisi_B, BacakYerCekimiVektor_A ve BacakYerCekimiVektor_B isimler altında verilmiştir.

Denklem (5.94)'te, her bir bacaktaki kuvveti elde edebilmek için üst platformun konum ve açısı, doğrusal ve açısal hızları, doğrusal ve açısal ivmeleri ve de Jacobian matrisinin tersi verilmesi gerekmektedir. Aşağıda, eklemlerde oluşan kuvvetlerin genel bir denklemi verilmiştir.

$$F = J^{-T} M(X_{p-o}) \ddot{X}_{p-o} + J^{-T} V_m(X_{p-o}, \dot{X}_{p-o}) \dot{X}_{p-o} + J^{-T} G(X_{p-o}) \quad (5.105)$$

Üst platforma ait yörünge bilgisi verildiği zaman, konum ve açı (X_{p-o}), doğrusal ve açısal hızları (\dot{X}_{p-o}), doğrusal ve açısal ivmeleri (\ddot{X}_{p-o}) bilinmektedir. \dot{X}_{p-o} ve X_{p-o} bilindiğinde M , V_m , G ve J hesaplanabilir. Bu denklemle bacaklardaki kuvveti elde edebilmek için tek koşul, verilen yörünge boyunca Jacobian matrisinin tersinin var olmasıdır. Başka bir deyişle bu yörünge boyunca Jacobian matrisinin

determinantı sıfırdan farklı olmalıdır. Bu, yörünge boyunca tekil noktaların olmaması anlamına gelir.

Denklem (5.94)'te ivme ifadesi yalnız bırakılacak şekilde yeniden yazılırsa aşağıdaki gibi elde edilir.

$$\ddot{X}_{P-O} = M^{-1}[J^T(X_{P-O})F - V_m(X_{P-O}, \dot{X}_{P-O})\dot{X}_{P-O} - G(X_{P-O})] \quad (5.106)$$

Bu denklemde ivmenin bulunabilmesi için kütle matrisinin tersinin alınabiliyor olması gerekir. Kütle matrisi pozitif tanımlı bir matris olduğundan bu matrisin tersi her koşulda alınabilir. Bu denklemden eklem uzayında kullanılabilecek denklem elde edilebilir. Denklem (5.107), Jacobian matrisi kullanılarak kartezyen uzayla eklem uzayı arasındaki ilişkiyi ifade eder.

$$\dot{L}_i = J_1 J_2 \dot{X}_{P-O} = J \dot{X}_{P-O} \quad (5.107)$$

Yukarıdaki ifadenin bir daha türevi alındığında aşağıdaki denklem elde edilir.

$$\ddot{L}_i = \dot{J} \dot{X}_{P-O} + J \ddot{X}_{P-O} \quad (5.108)$$

Burada kartezyen uzaydaki hız yerine eklem uzayındaki hız ifadesini yazıp kartezyen uzaydaki ivme ifadesi çekilirse Denklem (5.109) elde edilir.

$$\ddot{X}_{P-O} = J^{-1}\ddot{L}_i - J^{-1}\dot{J} \dot{X}_{P-O} \quad (5.109)$$

Denklem (5.106)'da, Denklem (5.107) ve (5.109)'u yerine koyup yeniden yazılırsa,

$$J^{-1}\ddot{L}_i - J^{-1}\dot{J} \dot{X}_{P-O} = M^{-1}[J^T F - V_m J^{-1}\dot{L}_i - G] \quad (5.110)$$

elde edilir. Bu denklem, eklem uzayındaki ivme yalnız kalacak şekilde yeniden düzenlendiğinde Denklem (5.111)'deki gibi;

$$\begin{aligned} \ddot{L}_i &= JM^{-1}[J^T F - V_m J^{-1}\dot{L}_i - G] + \dot{J} J^{-1}\dot{L}_i \\ &= JM^{-1}J^T F + (\dot{J} - JM^{-1}V_m)J^{-1}\dot{L}_i - JM^{-1}G \end{aligned} \quad (5.111)$$

şeklinde elde edilir. Bu denklem, eklem uzayındaki kuvvet yalnız kalacak şekilde düzenlendiğinde aşağıdaki gibi olur.

$$F = J^{-T} M J^{-1} \ddot{L}_i + J^{-T} (V_m - M J^{-1} \dot{J}) J^{-1} \dot{L}_i + J^{-T} G \quad (5.112)$$

Bu denklemi daha basit yazmak için yeni ifadeler kullanıldığında eklem uzayındaki kuvvet ifadesi aşağıdaki gibi olur.

$$F = M_e \ddot{L}_i + V_{me} \dot{L}_i + G_e \quad (5.113)$$

Burada kullanılan eklem uzayındaki kütle matrisi, Coriolis-merkezkaç matrisi ve yerçekimi vektörü aşağıda verilmiştir.

$$M_e = J^{-T} M J^{-1} \quad (5.114)$$

$$V_{me} = J^{-T} (V_m - M J^{-1} \dot{J}) J^{-1} \quad (5.115)$$

$$G_e = J^{-T} G \quad (5.116)$$

Görüldüğü üzere bu denklemde oldukça işlem yükü fazladır. Bu denklemde kuvvetin elde edilmesi, verilen yörünge boyunca Jacobian matrisinin tersinin alınabilmesini gerektirmektedir.

5.4.6. Eyleyiciler dinamiği

Burada, eyleyici dinamiği çıkarılırken hem motorun hem de vidalı bilyenin (ballscrew) kendi dönme eksenlerine göre dönmesi ile oluşan mekaniksel kısmın dinamiği ve de elektromanyetik alan içerisinde depolanan enerjiden kaynaklanan elektriksel kısmın dinamiği dikkate alınacaktır. Ayrıca, eyleyici dinamiğinin içeriğini göstermek amacıyla DC-motor kullanılacaktır.

Bir önceki bölümde, aşağıdaki vektör denklemi ile temsil edilen ikinci dereceden doğrusal olmayan diferansiyel denklemlerle modellenen Stewart Platform mekaniğinin katı gövde dinamiği elde edilmişti. Aşağıda Stewart Platform'una ait genel tork ifadesi verilmektedir.

$$\tau = M(X_{P-O}) \ddot{X}_{P-O} + N(X_{P-O}, \dot{X}_{P-O}) \dot{X}_{P-O} + G(X_{P-O}) \quad (5.117)$$

Yukarıdaki denklem, Denklem (5.118), (5.119) ve (5.120)'yi kullanarak tekrar düzenlendiğinde Denklem (5.121)'deki gibi;

$$\dot{L} = J \dot{X} \Rightarrow \dot{X} = J^{-1} \dot{L} \quad (5.118)$$

$$\ddot{L} = J \ddot{X} + \dot{J} \dot{X} \Rightarrow \ddot{X} = J^{-1} \ddot{L} - J^{-1} \dot{J} \dot{X} \quad (5.119)$$

$$\tau = J^T F \quad (5.120)$$

$$J^T F = M(X) [J^{-1} \ddot{L} - J^{-1} \dot{J} \dot{X}] + N(X, \dot{X}) J^{-1} \dot{L} + G(X) \quad (5.121)$$

elde edilir. Yukarıdaki denklem tekrar düzenlenip sadece eklem uzayı değişkeni türünden yazıldığında bacaklardaki giriş kuvvet vektörü aşağıdaki gibi elde edilir.

$$F = M_j(X) \ddot{L} + N_j(X) \dot{L} + G_j(X) \quad (5.122)$$

Denklemden yer alan $M_j(X)$, $N_j(X)$ ve $G_j(X)$ ifadeleri aşağıdaki denklemlerde sırasıyla verilmiştir.

$$M_j(X) = J^{-T} M(X_{P-O}) J^{-1} \quad (5.123)$$

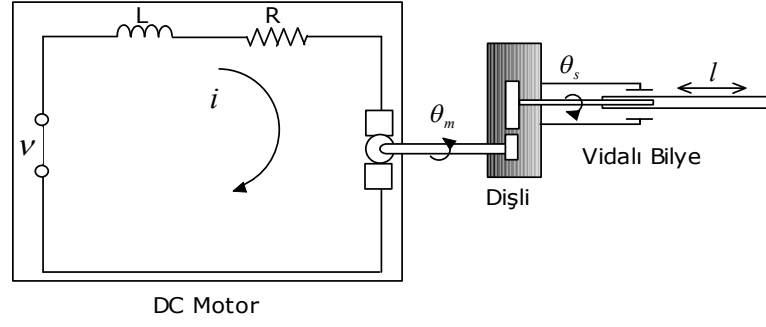
$$N_j(X) = (J^{-T} N(X_{P-O}, \dot{X}_{P-O}) - J^{-T} M(X_{P-O}) J^{-1} \dot{J}) J^{-1} \quad (5.124)$$

$$G_j(X) = J^{-T} G(X_{P-O}) \quad (5.125)$$

Denklemlerdeki \dot{J} , Jacobian matrisinin zamana göre türevidir. Eklem uzayında dinamik denklemler ifade edildiğinden eyleyici dinamiği kolayca bu dinamik modele eklenebilir. Eyleyicinin dinamik davranışını tarif eden denklemler, sırasıyla aşağıda tanımlandığı gibidir.

Şekil 5.6, tipik bir DC motor-vida bilyeli eyleyicinin başlıca elektrik ve mekanik parçalarını göstermektedir. Bu DC motor tarafından üretilen tork τ_{mi} , aşağıda verildiği gibi modellenir. Bu üretilen tork, elektrik kısmında yer alan armatür akımı i ile doğru orantılıdır.

$$\tau_{mi} = k_t i \quad (5.126)$$



Şekil 5.6. DC Motor ve motorun dişli mekaniksel kısmı

k_t , motor tork sabitidir. Motorun armatürüne ait elektriksel devrenin diferansiyel denklemi aşağıda verildiği gibidir.

$$v_i = L \frac{di}{dt} + Ri + k_b \dot{\theta}_{mi} \quad (5.127)$$

k_b , ters elektro motor kuvvet sabitini, θ_{mi} , motor milinin dönmesiyle oluşan açısıl konumu göstermektedir. Bunun yanında, mekaniksel olarak motor milinin modellenmesini ifade eden denklem aşağıda verildiği gibidir.

$$\tau_{mi} = J_m \ddot{\theta}_{mi} + b_m \dot{\theta}_{mi} + \tau_{si} \quad (5.128)$$

Denklemden yer alan J_m ve b_m sırasıyla, dönme eksenine ilişkin atalet momenti ve motor milinin sürtünme katsayısını gösterir. τ_{si} , vidalı bilye tarafından motor miline uygulanan torku verir. Vidalı bilye kısmındaki tork ifadesi Denklem (5.129)'daki gibi;

$$n \tau_{si} = J_s \ddot{\theta}_{si} + b_s \dot{\theta}_{si} + \tau_{ri} \quad (5.129)$$

modellenir. Yukarıda yer alan n dişli oranıdır. J_s ve b_s sırasıyla, vida-bilyesinin atalet momenti ve mekanik sürtünmesidir. τ_{ri} , vidalı bilyenin yuvası boyunca iç bağlama borusu (strut tube) tarafından vidalı bilyeye uygulanan torktur. θ_{si} , vidalı bilyenin açısıl konumudur ve aşağıdaki denklemde verildiği gibi motor milinin konumu ile ilişkilidir.

$$\theta_{mi} = n \theta_{si} \quad (5.130)$$

Denklem (5.128), (5.129) ve (5.130)'u kullanarak rotordaki eyleyici yük sisteminin tork denklemi aşağıdaki gibi elde edilir.

$$n\tau_{mi} = (J_s + n^2 J_m) \ddot{\theta}_{si} + (b_s + n^2 b_m) \dot{\theta}_{si} + \tau_{ri} \quad (5.131)$$

Yukarıdaki tork ifadesi, Stewart Platform'unun her bir bacağına hareketlendiren doğrusal kuvvete (f_i), vidalı-bilye mekanizması ile dönüştürülür. Buna göre τ_{ri} ile f_i arasındaki ilişki aşağıda verilmiştir.

$$f_i = \frac{2\pi}{\rho} \tau_{ri} \quad (5.132)$$

Yukarıda yer alan ρ vidalı bilyenin açılal konumunu ilişkilendiren vidalı-bilye dönüş adımıdır (ball-screw pitch). Buna göre bacağın lineer hareketini (l_i) veren ifade aşağıdaki gibi bulunur.

$$l_i = \frac{\rho}{2\pi} \theta_{si} \quad (5.133)$$

Denklem (5.132) ve (5.133)'ü kullanarak Denklem (5.131) tekrar düzenlendiğinde aşağıdaki gibi elde edilir.

$$\tau_{mi} = \frac{2\pi}{n\rho} (J_s + n^2 J_m) \ddot{l}_i + \frac{2\pi}{n\rho} (b_s + n^2 b_m) \dot{l}_i + \frac{\rho}{n2\pi} f_i \quad (5.134)$$

Yukarıdaki denklemi kısa olarak ifade edersek aşağıdaki gibi yazılabilir.

$$\tau_{mi} = m_a \ddot{l}_i + n_a \dot{l}_i + k_a f_i \quad (5.135)$$

Bu denklem *i.ci* bağına eyleyici dinamiğini tanımlar. Stewart Platform'unda 6 eyleyici ve bunların özdeş olduğu kabul edilirse yukarıdaki ifade matris formunda yazılabilir.

$$\tau_m = M_a \ddot{L} + N_a \dot{L} + K_a F \quad (5.136)$$

Yukarıda yer alan ifadeler aşağıdaki denklemlerde sırasıyla verilmiştir.

$$M_a = m_a I_{6 \times 6} = \frac{2\pi}{n\rho} (J_s + n^2 J_m) I_{6 \times 6} \quad (5.137)$$

$$N_a = n_a I_{6 \times 6} = \frac{2\pi}{n\rho} (b_s + n^2 b_m) I_{6 \times 6} \quad (5.138)$$

$$K_a = k_a I_{6 \times 6} = \frac{\rho}{n2\pi} I_{6 \times 6} \quad (5.139)$$

5.4.7. Eyleyici dinamiği eklenmiş sistem dinamiği

Burada, Jacobian matrislerinin farklı yöntemlerle elde edilışinden dolayı hem Yöntem A hem de Yöntem B'ye göre çıkarılan platform dinamiğine, elde edilen eyleyici dinamiği eklenerek tüm sistemin dinamik denklemleri çıkarılacaktır.

İlk olarak Yöntem A için Denklem (5.136) ifadesi, eklem uzayı deęişkeni türünden elde edilen bacaklardaki giriş kuvvet vektörü (Denklem (5.122)) ile birleřtirildiğinde ařağıdaki gibi ifade edilir.

$$\tau_m = M_a \ddot{L} + N_a \dot{L} + K_a F = M_a \ddot{L} + N_a \dot{L} + K_a \left({}^A M_j(X) \ddot{L} + {}^A N_j(X) \dot{L} + {}^A G_j(X) \right) \quad (5.140)$$

Yukarıdaki denklem daha anlaşılır ve kısa olarak ifade edildiğinde ařağıdaki gibi yazılabilir.

$$\tau_m = {}^A \bar{M}_j(X) \ddot{L} + {}^A \bar{N}_j(X) \dot{L} + {}^A \bar{G}_j(X) \quad (5.141)$$

Denklem (5.141)'de yer alan ifadeler ařağıda sırasıyla verilmiştir. Bu denklem, eklem uzayında ifade edilmiş katı gövde ve eyleyici dinamiklerin birleřimini veren ifadedir.

$${}^A \bar{M}_j(X) = K_a (J_A)^{-T} {}^A M(X_{P-O}) (J_A)^{-1} + M_a \quad (5.142)$$

$${}^A \bar{N}_j(X) = K_a \left((J_A)^{-T} {}^A N(X_{P-O}, \dot{X}_{P-O}) - (J_A)^{-T} {}^A M(X_{P-O}) (J_A)^{-1} \dot{J}_A \right) (J_A)^{-1} + N_a \quad (5.143)$$

$${}^A \bar{G}_j(X) = K_a (J_A)^{-T} {}^A G(X_{P-O}) \quad (5.144)$$

Stewart Platform'un dinamik denklemleri, kartezyen uzayında çıkarıldığından ve simülasyonlar buna göre yapılacağından yukarıda elde edilen bu birleşik denklem kartezyen uzayında aşağıdaki gibi yazılabilir.

$$\tau_m = {}^A\bar{M}_c(X)\ddot{X} + {}^A\bar{N}_c(X)\dot{X} + {}^A\bar{G}_c(X) \quad (5.145)$$

$${}^A\bar{M}_c(X) = K_a(J_A)^{-T} {}^A M(X_{p-o}) + M_a J_A \quad (5.146)$$

$${}^A\bar{N}_c(X) = K_a(J_A)^{-T} {}^A N(X_{p-o}, \dot{X}_{p-o}) + N_a J_A + M_a \dot{J}_A \quad (5.147)$$

$${}^A\bar{G}_c(X) = K_a(J_A)^{-T} {}^A G(X_{p-o}) \quad (5.148)$$

Yöntem A'dakine benzer şekilde aynı adımlar izlenerek Yöntem B için kartezyen uzayında Stewart Platform'un dinamik denklemleri aşağıdaki gibi elde edilir.

$$\tau_m = {}^B\bar{M}_c(X)\ddot{X} + {}^B\bar{N}_c(X)\dot{X} + {}^B\bar{G}_c(X) \quad (5.149)$$

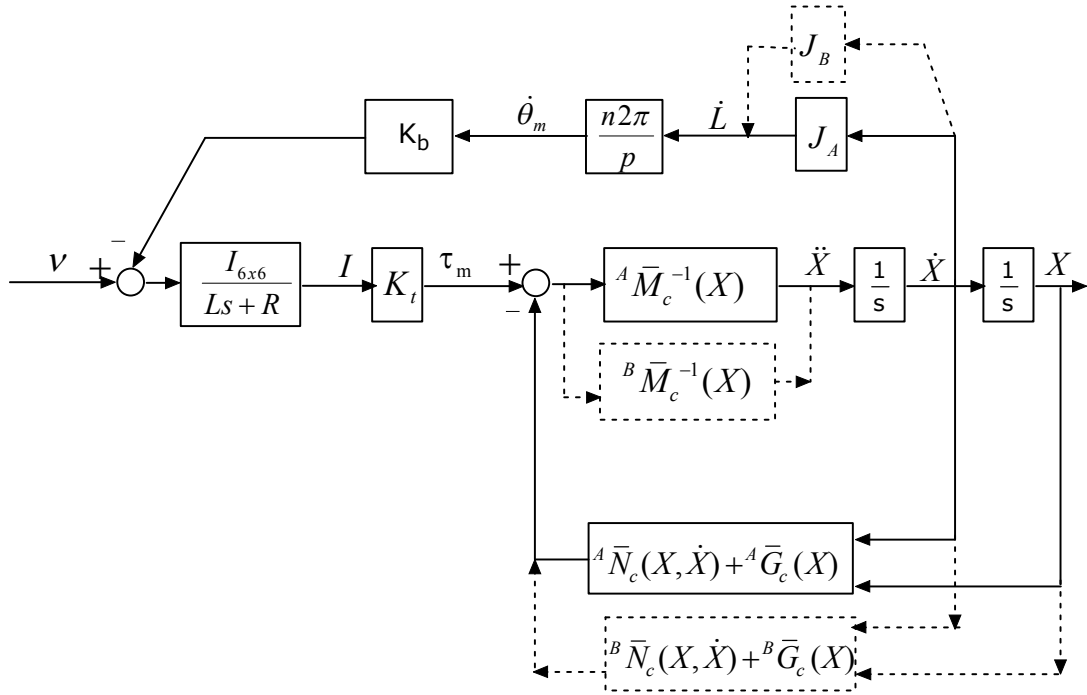
$${}^B\bar{M}_c(X) = K_a(J_B)^{-T} {}^B M(X_{p-o}) + M_a J_B \quad (5.150)$$

$${}^B\bar{N}_c(X) = K_a(J_B)^{-T} {}^B N(X_{p-o}, \dot{X}_{p-o}) + N_a J_B + M_a \dot{J}_B \quad (5.151)$$

$${}^B\bar{G}_c(X) = K_a(J_B)^{-T} {}^B G(X_{p-o}) \quad (5.152)$$

Hem Yöntem A hem de Yöntem B'ye göre elde edilen tüm sistemin dinamiği için oluşturulan Stewart Platform'un simülasyon blok diyagramı, Şekil 5.7'de gösterilmektedir.

Şekil 5.7'den de anlaşılacağı üzere, yapılacak deneysel çalışmalarda, eklem uzayında sistem dinamiğini tanımlayan denklemlerden ziyade kartezyen uzayında sistem dinamiğini tanımlayan denklemler kullanılacaktır. Bunu sebebi, eklem değişkeni olan L 'den üst platformun konumu ve yönelimini veren X_{p-o} 'yu hesaplayan ileri kinematiği kullanmaktan kaçınmaktır ve de katı gövde dinamiğinin hesaplanabilmesi ve simüle edilebilmesi için X_{p-o} 'nun bilinmesi yeterlidir.

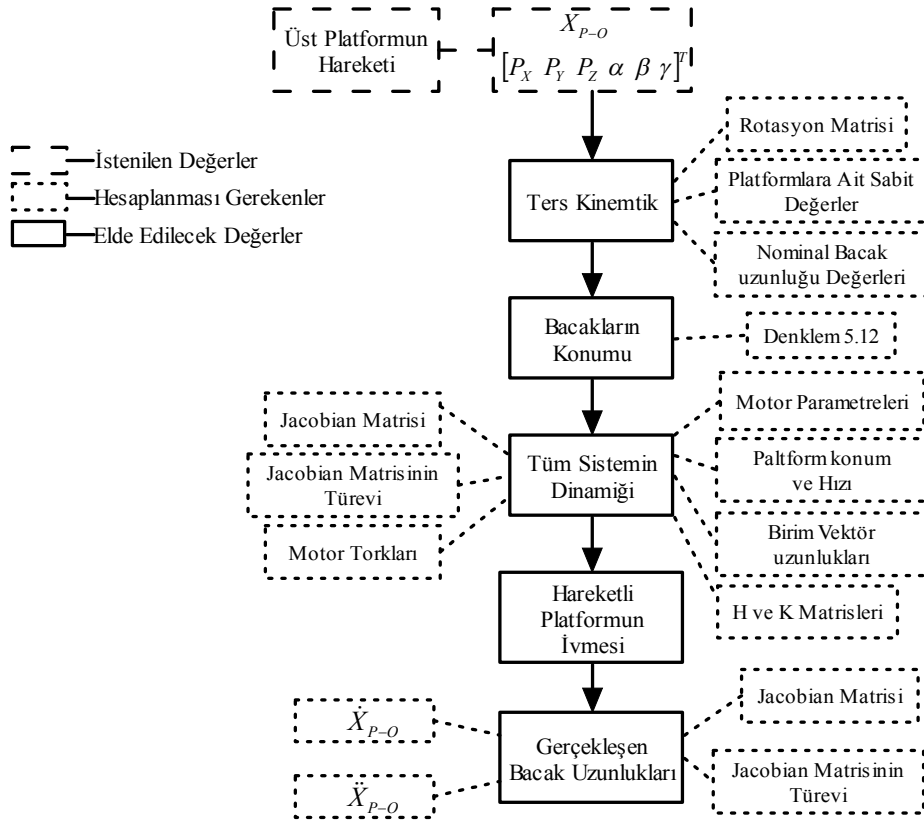


Şekil 5.7. Tüm Stewart Platform dinamiğinin simülasyon blok diyagramı

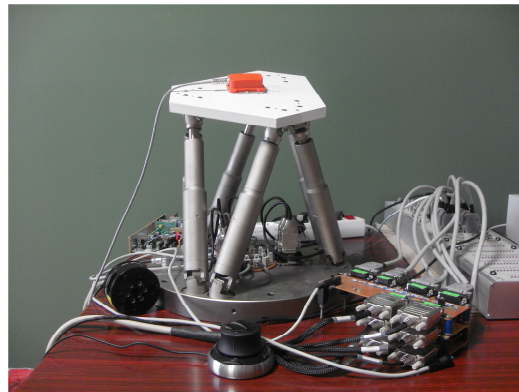
5.5. Deneysel Kurulum ve Simülasyon Blokları

Buraya kadar olan bölümlerde, Stewart Platform manipülatörün hem eyleyici hem de katı gövde dinamik denklemlerinin elde edilmesi verilmiştir. Bu bölümde ise, elde edilen denklemlerden yararlanılarak Stewart Platform'un dinamik davranışını gözlemlemek amacıyla tüm sistemin dinamik modeli, Matlab-Simulink ortamında bloklar halinde gerçekleştirilmesi yapılacaktır. Böylece, gerçekleştirilen bu Simulink model ile gerçek robot üzerinde yapılan deneylerden elde edilen veriler karşılaştırılarak tüm sistem için elde edilen dinamik modelin geçerliliği doğrulanacaktır.

Verilen üst platform yörüngesine göre Stewart Platform'u için dinamik denklemlerin simülasyon algoritmasına ait akış diyagramı, Şekil 5.8'de gösterilmektedir. Şekil 5.9'da gösterilen Stewart Platform manipülatörü, dinamik modelleme sürecinde kullanılmıştır. Şekilde gösterildiği gibi bu deneysel manipülatör, iki ana gövde (üst ve alt platform), altı doğrusal motor, kontrolör, güç kaynağı, acil durdurma devresi ve kontrolör-robot ara geçiş kartından oluşmaktadır.



Şekil 5.8. Dinamik simülasyon algoritması

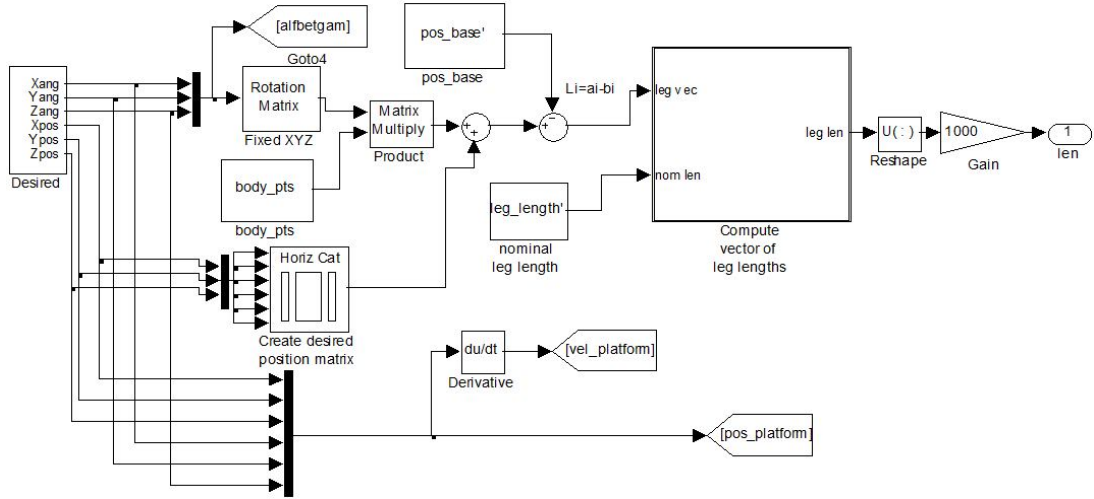


Şekil 5.9. Stewart Platform düzeneği

Model doğrulama deneylerinin uygulanabilmesi için MATLAB-Simulink ortamında tamamen programlanabilen ve gerçek zamanlı kodların yüklenebildiği dSPACE DS1103 gerçek zamanlı kontrolör kullanılmıştır. Böylece, gerçekleşen deneyin gözlemlenebilmesi ve deney gerçekleştirilirken her bacakta yer alan enkoderlerden verilerin toplanabilmesi sağlanmıştır.

5.5.1. Simülasyon blokları

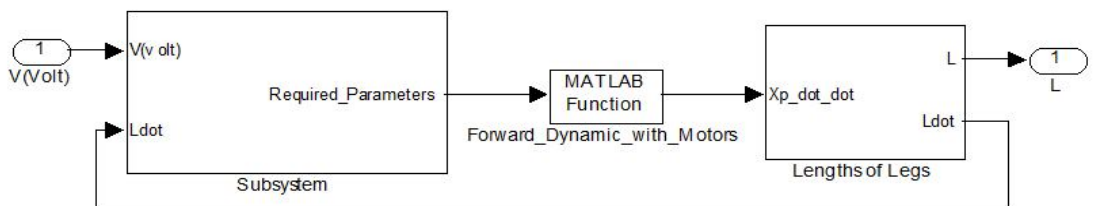
İlk olarak, 6 serbestlik dereceli Stewart Platform manipülâtörün referans bacak uzunluklarını hesaplayan ters kinematik çözümü, Şekil 5.10'da gösterilen Leg_Trajectory isimli simülasyon bloğunda verilmiştir.



Şekil 5.10. Referans bacak uzunluğunu hesaplayan “Leg_Trajectory” bloğu

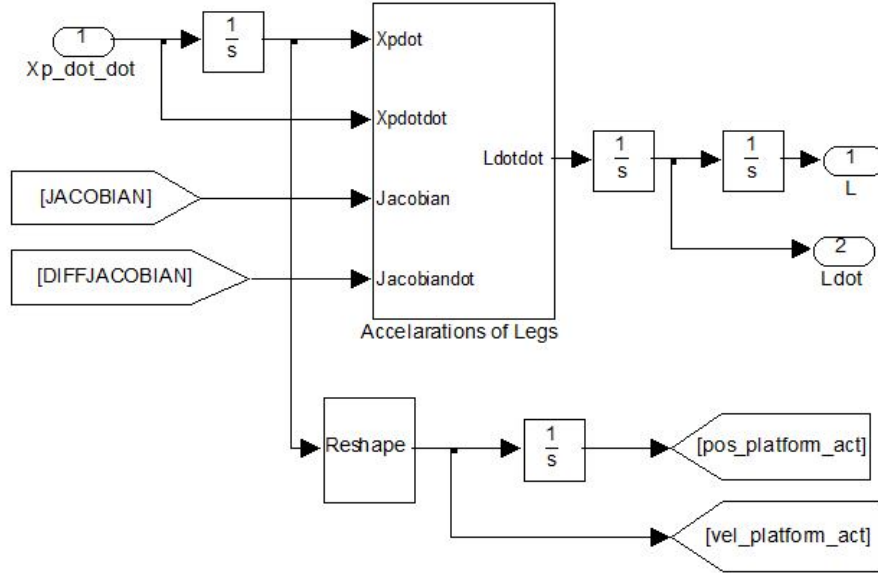
Bu blok, her bir bacağa ait uzunluğu zamana bağlı olarak hesaplayıp “len” adıyla referans yörünge şeklinde dışarı vermektedir. Şekil 5.10’da gösterilen “Desired” bloğunda Xang, Yang ve Zang sırasıyla hareketli platformun α , β ve γ dönme açılarını Xpos, Ypos ve Zpos ise Px, Py ve Pz konumları temsil etmektedir.

Şekil 5.11, Stewart_Platform_Dynamics bloğunun içerdiği, ileri dinamik denklemleri için gerekli olan ifadelerin (Jacobian matrisi, H ve K matrisleri, Motor torkları, Jacobian matrisinin türevi) hesaplandığı Subsystem bloğunu, dinamik denklemlerin hesaplandığı Forward_Dynamic_with_Motors fonksiyonu ve gerçekleşen bacak uzunluklarının elde edildiği Length of Legs bloklarını göstermektedir.



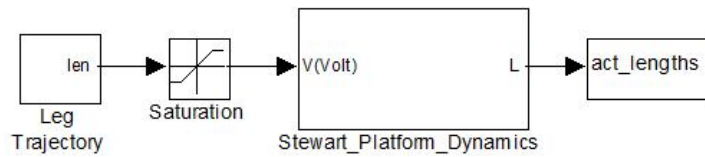
Şekil 5.11. Gerçekleşen bacak uzunluğu üreten “Stewart_Platform_Dynamics” bloğu

Şekil 5.11’de yer alan Lengths of Legs bloğu, Denklem (5.108)’in hesaplanması sonucu bacak uzunluklarını veren bloktur. Şekil 5.12, denklemin hesaplandığı simülink bloğunu göstermektedir.



Şekil 5.12. Bacak uzunlukların hesaplandığı “Lengths of Legs” bloğu

Şekil 5.13, herhangi bir kontrolör olmadan Yöntem A’ya göre çıkarılan dinamik ifadelerden (Denklem (5.145)) ya da Yöntem B’ye göre çıkarılan dinamik ifadelerden (Denklem (5.149)) faydalanarak 6 serbestlik dereceli Stewart Platform’unun simülink ortamında gerçekleştirilmesini gösterir.

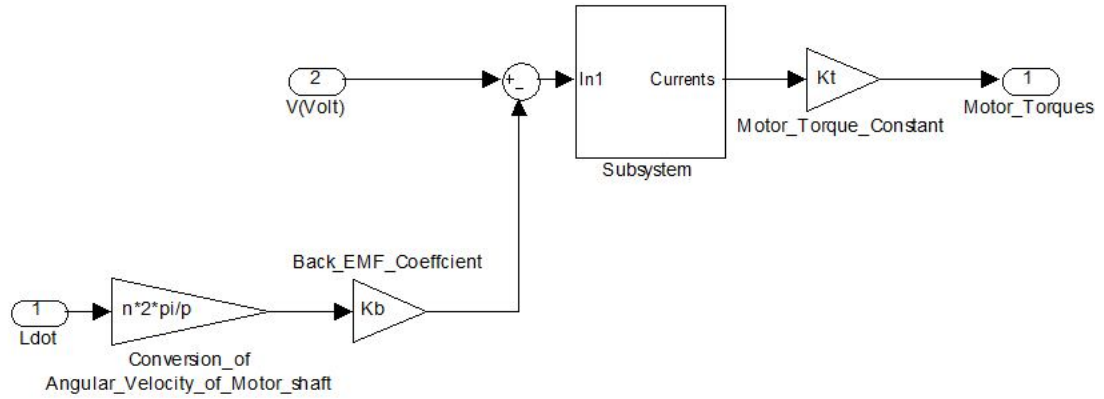


Şekil 5.13. Stewart Platform’unun simülink modeli

Şekil 5.13’te gösterildiği gibi hareketli platformun konum ve yönelimi verildiğinde $X_{P-O} = [P_x \ P_y \ P_z \ \alpha \ \beta \ \gamma]^T$ buna karşılık gelen bacak uzunlukları hesaplanır. Bu bacak uzunlukları limit sınırları içerisinde tutulduktan sonra Stewart Platform’un hem katı gövde dinamiği (hareketli platform ve bacak dinamikleri) hem de eyleyici dinamiklerini içeren tüm sistem dinamiği hesaplanır. Böylece gerçekleşen bacak uzunlukları elde edilir.

5.5.2. Eyleyici ve platform parametreleri

Şekil 5.14, Stewart Platform'unda kullanılan eyleyicilere ait motor torkların hesaplandığı simülink bloğunu göstermektedir. Tablo 5.1'de ise simülasyonda kullanılan motorlara ait parametre değerleri verilmiştir.



Şekil 5.14. Eyleyicilerde üretilen torkların simülink modeli

Tablo 5.1. Eyleyici parametreleri

Parametre	Birimi	Değeri
Armatür Direnci R	ohm	7.10
Armatür Endüktansı L	H	265e-6
Elektromotor Kuvvet Sabiti K_b	V/rpm	2.730e-3
Tork Sabiti K_t	Nm/A	26.10e-3
Dişli Oranı n	Rad/rad	1
Motor Mili Atalet Momenti J_m	Kg.m ²	0.58e-6
Vidalı Bilye Atalet Momenti J_s	Kg.m ²	0.0002857e-3
Motor Milinin Viskoz Sürtünme Katsayısı B_m	N.s/rad	0.00196e-3
Vidalı Bilye Viskoz Sürtünme Katsayısı B_s	N.s/rad	0.123e-3
Vidalı Bilye Dönüş Adımı ρ	m	0.001



(a)



(b)

Şekil 5.15. Vidalı Bilye (Ball-Screw)

Eyleyicilerde kullanılan vidalı bilye Şekil 5.15'te gösterilmektedir. Bunun sayesinde motorda oluşan dönme hareketi lineer harekete dönüştürülür. Böylece, hem açıdan mm'ye hem de torktan kuvvete geçilir.

Deneyde kullanılan platforma ait temel kinematik parametreler Tablo 5.2'de verilmiştir.

Tablo 5.2. Stewart Platform'un yapısal parametreleri

Parametre	Birimi	Değeri
Üst platform kütlesi m_{up}	Kg	1.1324
Bacağın sabit kısmın kütlesi m_1	Kg	0.4279
Bacağın hareketli kısmın kütlesi m_2	Kg	0.1228
Bacağın sabit kısmın uzunluğu l_1	m	0.22
Bacağın hareketli kısmın uzunluğu l_2	m	0.05
Üst platformun yarıçapı r_p	m	0.18867975191
Alt platformun yarıçapı r_{base}	m	0.29326616983
Üst platformun atalet tensörü $I_{(mf)}$	Kg.m ²	$\begin{bmatrix} 0.0025 & 0 & 0 \\ 0 & 0.0025 & 0 \\ 0 & 0 & 0.005 \end{bmatrix}$

Çalışmada kullanılan Stewart Platform'u için bacakların üst ve alt platforma bağlantı noktalarının hesaplandığı, eyleyicilerde yer alan motor parametre değerlerinin bulunduğu ve de hareketli platforma ait parametre değerlerinin hesaplandığı MATLAB ortamında yazılan kodlar, EK-A'da RobotParametersSetup ve InitialParameters_with_Motor isimler altında verilmiştir.

5.6. Deneysel ve Simülasyon Sonuçları

İki farklı yöntemle göre MATLAB-Simulink ortamında geliştirilen Stewart Platform'unun dinamik denklemlerinin doğruluklarını karşılaştırmak amacıyla gerçek robot üzerinde deneyler yapılacaktır. Robot üzerinde yapılan deneylerden elde edilen veriler ile yöntemlere göre oluşturulan dinamik model sonuçları, hem grafiksel hem de tablolar şeklinde karşılaştırılacaktır. Her bir deney için hareketli platformun yörünge denklemi verilecek ve referans bacak konumları ile robotun ve yöntemlerin gerçekleştirdiği konumlar grafiksel olarak gösterilecektir.

Tüm deney grafiklerinde, deneysel Stewart Platform'un gerçekleştirdiği bacak konumları Robot olarak, Yöntem A ve Yöntem B'ye göre dinamik modellerin gerçekleştirilen bacak konumları ise sırasıyla SIM-A ve SIM-B olarak gösterilmiştir. Ayrıca, her bir deney sırasında robotun gerçekleştirdiği bacak uzunlukları ile yöntemlere göre elde edilen dinamik modellerin gerçekleştirdiği bacak uzunlukları arasındaki hataya göre maliyet fonksiyon değerleri hesaplanacaktır. Aşağıda verilen maliyet fonksiyon denklemine göre tüm bacaklar için elde edilen maliyet değerleri tablo halinde verilecektir.

$$E(k) = \frac{1}{N} \sum_{i=1}^N |e_k(i)| \quad (5.153)$$

Denkleminde yer alan e_k , deneysel veri ile dinamik modellerden elde edilen simülasyon sonuçları arasındaki k . bacağın hatasını, N ise toplam örnek sayısını göstermektedir.

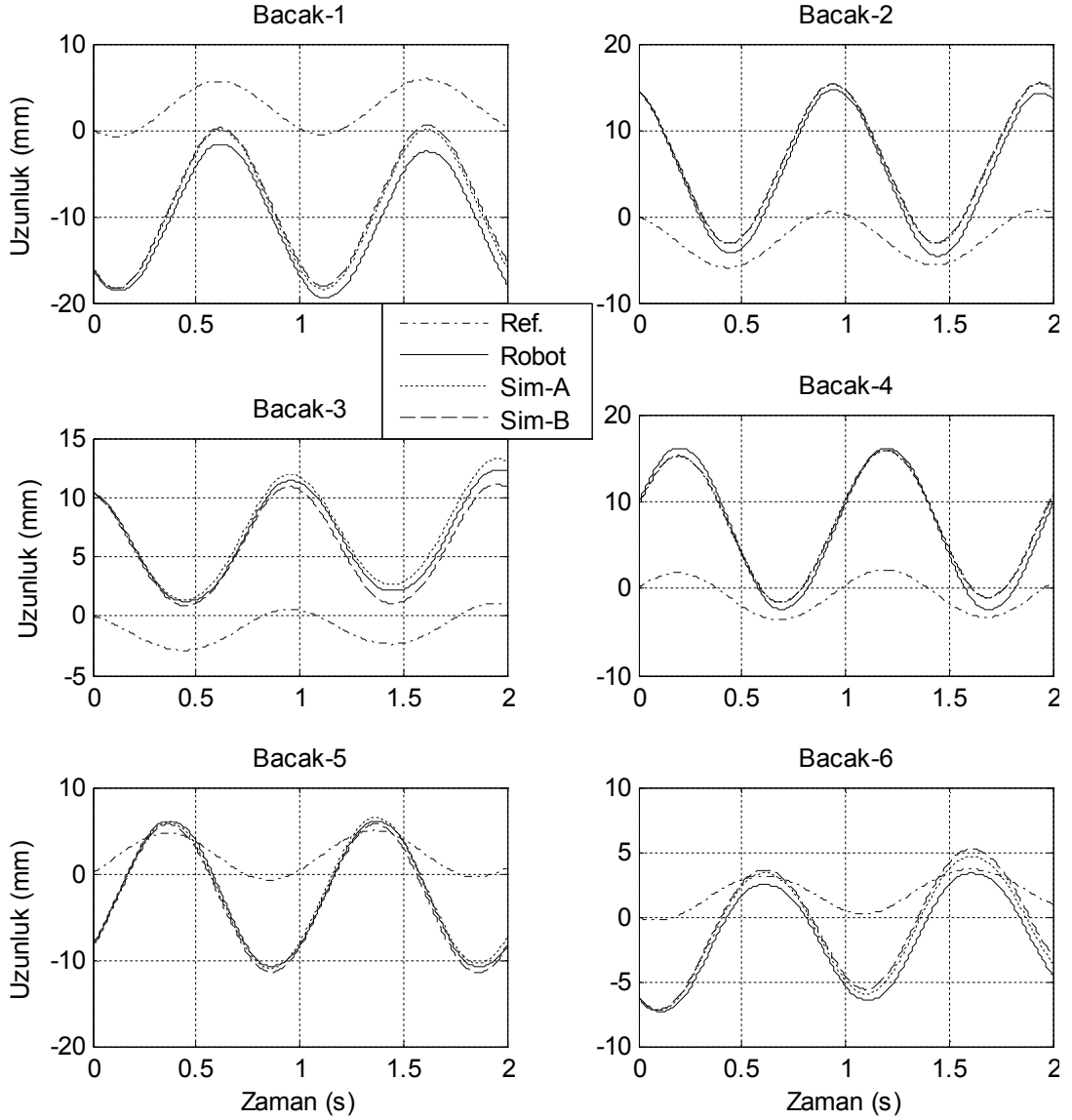
5.6.1. Deney 1

J_A ve J_B matrislerine göre elde edilen dinamik modelleri doğrulamak ve deneysel veri ile simülasyon sonuçlarını karşılaştırmak için ilk olarak, aşağıda verilen hareketli platform yörüngesi kullanılmıştır.

$$\begin{aligned} x(t) &= 0 & \alpha(t) &= 10 \cos(2\pi t) \\ y(t) &= 0 & \beta(t) &= 10 \sin(2\pi t) \\ z(t) &= 0.25 & \gamma(t) &= 10 \cos(2\pi t) \end{aligned} \quad (5.154)$$

Verilen yörüngeden anlaşılacağı üzere hareketli platform yalnızca x-y-z eksenlerinde bir dönme hareketi yapacaktır. Bu platform yörüngesine göre bacakların referans uzunlukları, manipülâtörün ters kinematiği ile hesaplanıp hem gerçek robot üzerinde hem de dinamik modellerin oluşturulduğu simülasyonda uygulanmıştır.

Şekil 5.16, verilen bacak referans yörüngeleri ile deneysel Stewart Platform manipülâtörün ve iki farklı yöntemle elde edilen dinamik modellerin gerçekleştirdikleri bacak yörüngelerini göstermektedir.



Şekil 5.16. Deney 1 için verilen yörünge boyunca simülasyon sonuçlarının karşılaştırılması

Şekil 5.16’da görüldüğü gibi her iki yöntem de robotun gerçekleştirdiği bacak konum yörüngelerini takip edebilmektedir. Grafik gözlemlendiğinde 1. ve 6. bacak konumlarındaki takip hatası diğer bacaklarda meydana gelen hata değerlerine göre daha fazladır. Ayrıca, referans bacak konumlarına bakıldığında kontrolör olmadığından bacak yörüngelerinde hem yöntemler hem de robotta gerçekleştirilen değerler için faz kayması olmadığı ancak tepe değerleri arasında fark olduğu görülmektedir. Her iki yöntemde çıkarılan dinamik modeller için her bir bacakta meydana gelen maliyet değerleri, Denklem (5.153) kullanılarak Tablo 5.3’te verilmektedir.

Tablo 5.3. Deney 1'e göre dinamik modellerin maliyet fonksiyon deęerleri

Model	Bacak 1	Bacak 2	Bacak 3	Bacak 4	Bacak 5	Bacak 6
SIM-A	1.4706	0.9826	0.4790	0.7349	0.3170	0.7526
SIM-B	1.8036	1.0301	0.6172	0.7260	0.5240	1.0895

Tablo incelendięinde, her iki dinamik modelin (SIM-A ve SIM-B) cevaplarında az farklılıklar olduęu grlmektedir. Ancak, sistemin tm dinamięini ieren SIM-A'nın dięer tm dinamikli SIM-B'den daha iyi performans gsterdięi verilen tablodan anlaşılmaktadır.

5.6.2. Deney 2

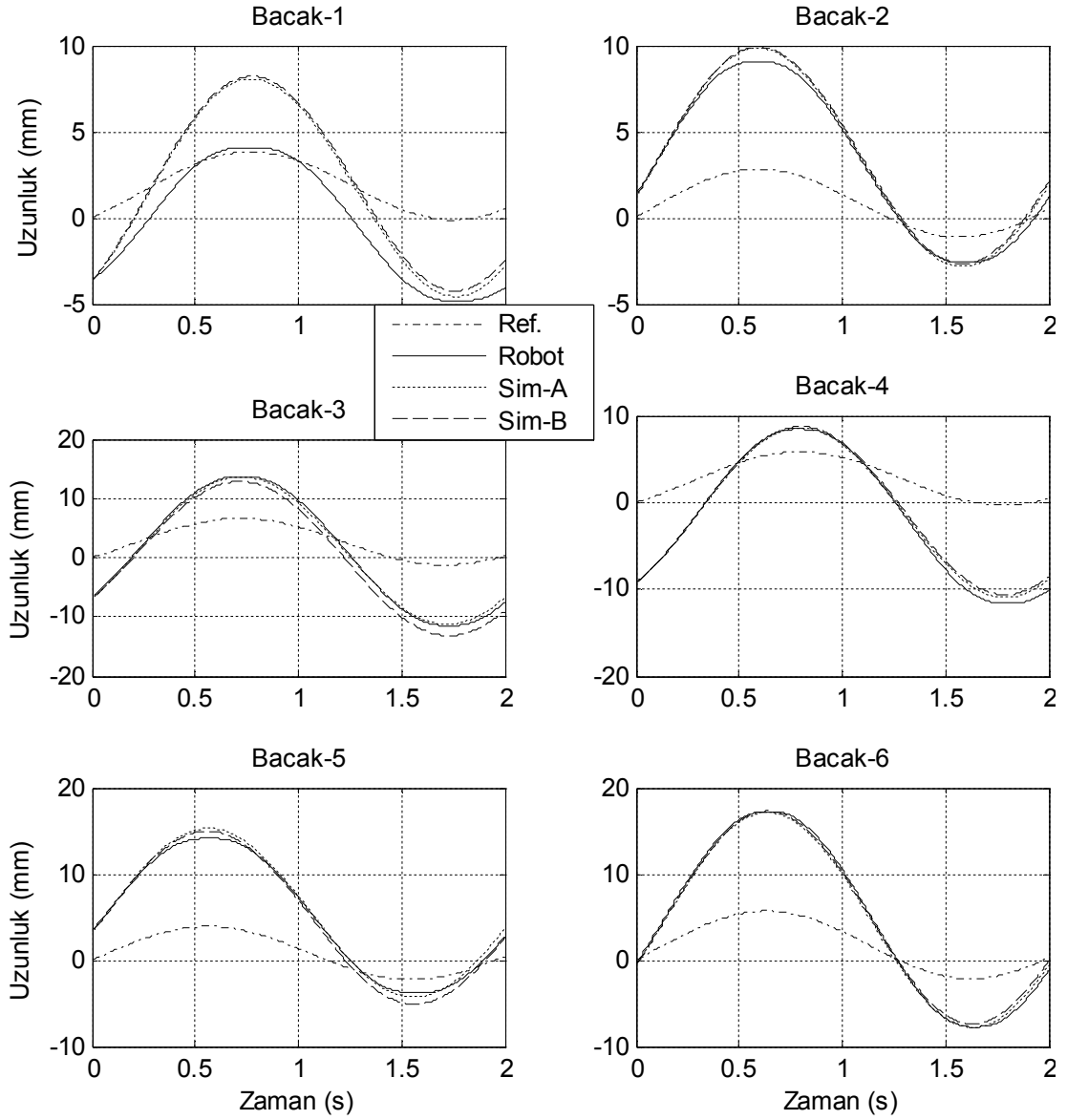
Bu doęrulama deneyinde, hareketli platform yalnızca x-y-z eksenlerinde teleme hareketi yaparken gerekleřen bacak konumları gzlemlenecektir. Bunun iin aŐaęıda verilen hareketli platformun yrnge denklemini kullanılmıŐtır.

$$\begin{aligned}x(t) &= 10 \sin(\pi t) & \alpha(t) &= 0 \\y(t) &= 10 \cos(\pi t) & \beta(t) &= 0 \\z(t) &= 0.25 + 10 \sin(\pi t) & \gamma(t) &= 0\end{aligned}\tag{5.155}$$

Verilen hareketli platform yrnge denklemine gre ters kinematik sonucu elde edilen referans bacak konumları ile robotun ve dinamik modellerin gerekleŐtirdięi bacak konumları, Őekil 5.17' de gsterilmektedir.

Grafięe bakıldıęında, hem Yntem A (SIM-A) hem de Yntem B' ye (SIM-B) gre elde edilen dinamik modellerin, robotun gerekleŐtirdięi bacak yrngelerini takip edebildikleri grlmektedir. Őekil incelendięinde, her iki yntem iin 1. bacakta meydana gelen takip hatasının dięer bacaklara gre daha fazla olduęu anlaşılmaktadır. Yntemler karŐılaŐtırıldıęında, SIM-A modelinin daha yakın yrnge takibi yapabildięi grlmektedir.

Yapılan model doęrulama deneyi sonucu, SIM-A ve SIM-B dinamik modellerin gerekleŐtirdięi bacak uzunlukları hatalarına gre hesaplanan maliyet deęerleri her bir bacak iin Tablo 5.4'te verilmiŐtir.



Şekil 5.17. Deney 2 için verilen yörünge boyunca simülasyon sonuçlarının karşılaştırılması

Tablo incelendiğinde, SIM-A dinamik modeli, her bir bacakta daha düşük maliyet değerlerine sahiptir. Sonuç olarak, tabloda verilen değerler ve grafik göz önünde bulundurulduğunda, SIM-A dinamik modelin yörünge takip performansının, SIM-B dinamik modeline göre daha iyi olduğu görülmektedir.

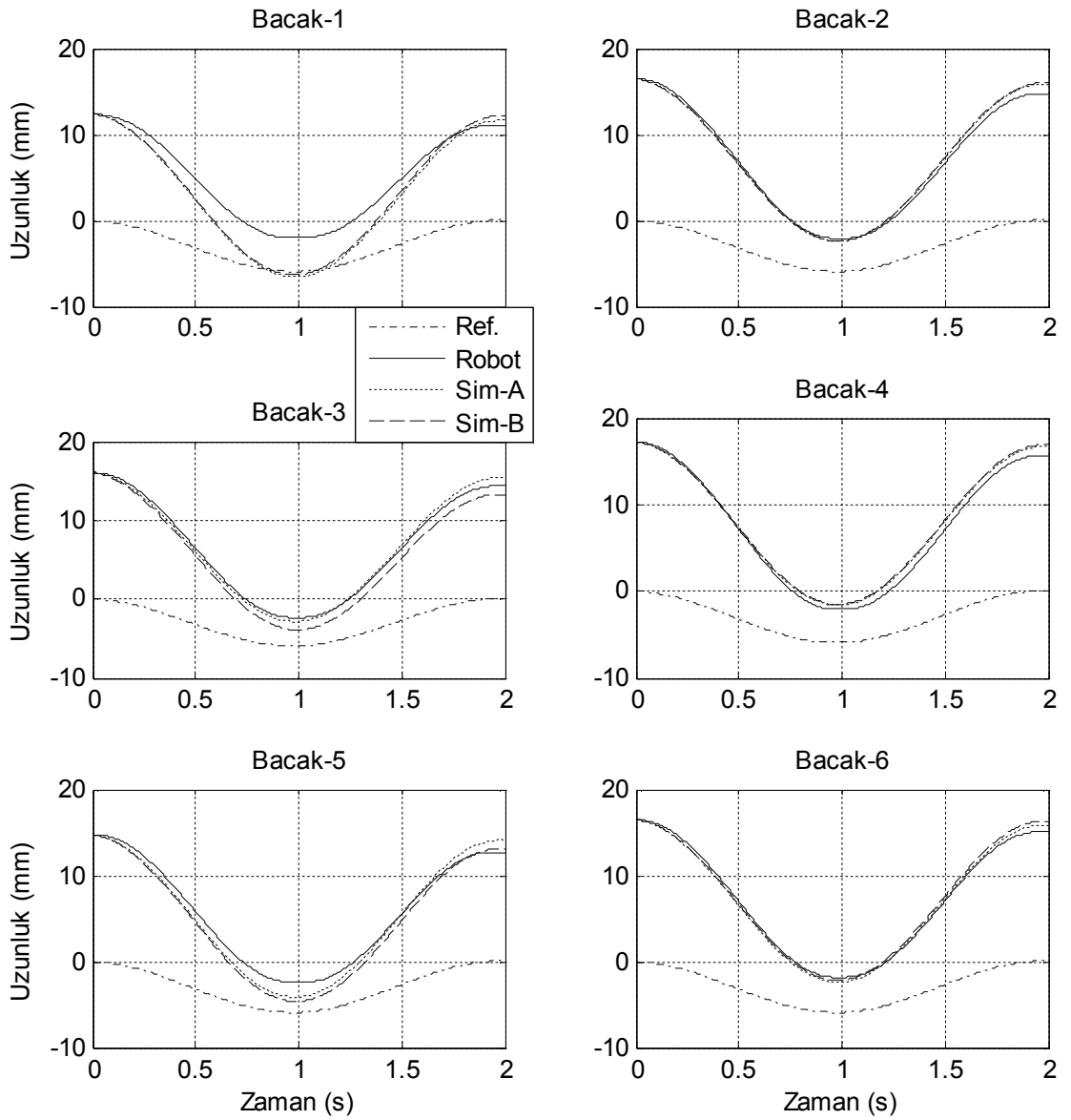
Tablo 5.4. Deney 2'ye göre dinamik modellerin maliyet fonksiyon değerleri

Model	Bacak 1	Bacak 2	Bacak 3	Bacak 4	Bacak 5	Bacak 6
SIM-A	1.9294	0.2858	0.2874	0.3186	0.4072	0.3129
SIM-B	2.1130	0.3469	1.1384	0.4544	0.5289	0.3533

5.6.3. Deney 3

Bu deneyde, hareketli platform yalnızca z yönünde hareket ettirildiğinde elde edilen dinamik modellerin ve robotun gerçekleştirdiği cevaplar incelenmiştir. Hareketli platformun konum ve yönelimini veren yörünge denklemi aşağıda verildiği gibidir.

$$\begin{aligned}x(t) &= 0 & \alpha(t) &= 0 \\y(t) &= 0 & \beta(t) &= 0 \\z(t) &= 0.25 + 10\sin(\pi t) & \gamma(t) &= 0\end{aligned}\tag{5.156}$$



Şekil 5.18. Deney 3 için verilen yörünge boyunca simülasyon sonuçlarının karşılaştırılması

Hem deneysel robota hem de SIM-A ve SIM-B simülasyon modellerine uygulanan referans bacak yörüngelerine göre robotun ve modellerin gerçekleştirdiği yörünge takipleri, Şekil 5.18’de gösterilmektedir. Şekilde verilen deneysel sonuçlara bakıldığında, SIM-A simülasyon dinamik modelin, robotun gerçekleştirdiği bacak yörüngelerini daha yakın bir şekilde takip edebildiği görülmektedir. Şekilde görüldüğü gibi 1. ve 5. bacaklarda dinamik modellerin yapmış olduğu hatalar daha büyüktür.

Yapılan model doğrulama deneyi sonucu, SIM-A ve SIM-B dinamik modellerin gerçekleştirdiği bacak uzunlukları hatalarına göre hesaplanan maliyet değerleri her bir bacak için Tablo 5.5’te verilmiştir.

Tablo 5.5. Deney 3’e göre dinamik modellerin maliyet fonksiyon değerleri

Model	Bacak 1	Bacak 2	Bacak 3	Bacak 4	Bacak 5	Bacak 6
SIM-A	2.2137	0.4212	0.4109	0.5940	0.8760	0.3701
SIM-B	2.0560	0.4418	1.0693	0.6839	1.0779	0.4468

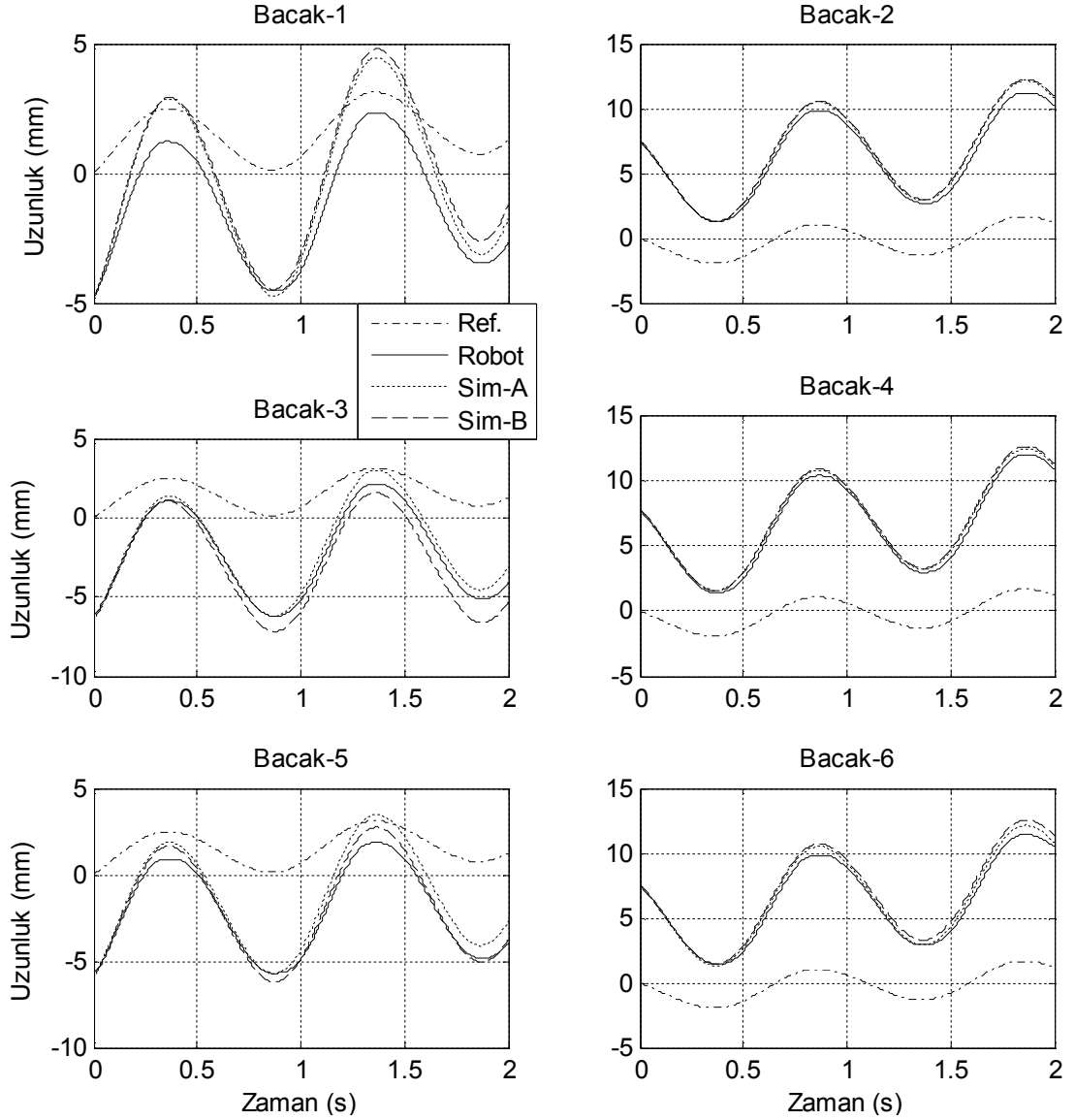
Tabloda verilen değerlere göre genel olarak yöntemler karşılaştırıldığında, Yöntem B’ye göre oluşturulan SIM-B’nin maliyet değerleri, Yöntem A’ya göre oluşturulan SIM-A’nın sonuçlarından daha büyüktür. Bu sonuç, Yöntem A tabanlı SIM-A simülasyon modelinin gerçek sisteme daha yakın bir dinamik model olduğunu ve bu model kullanıldığında belirgin olarak yüksek performansın elde edileceğini gösterir.

5.6.4. Deney 4

SIM-A ve SIM-B dinamik modellerin ve deneysel robotun gerçekleştirdiği bacak konumlarını karşılaştırmak için yalnızca z yönünde dönme hareketini sağlayan hareketli platformun yörünge denklemi aşağıda verilmiştir.

$$\begin{aligned}
 x(t) &= 0 & \alpha(t) &= 0 \\
 y(t) &= 0 & \beta(t) &= 0 \\
 z(t) &= 0.25 & \gamma(t) &= 15 \cos(2\pi t)
 \end{aligned} \tag{5.157}$$

Verilen yörünge denklemine göre ters kinematik sonucu elde edilen referans bacak uzunlukları, hem dinamik modellere hem de deneysel robota uygulandığında elde edilen deneysel ve simülasyon sonuçları Şekil 5.19’da gösterilmektedir.



Şekil 5.19. Deney 4 için verilen yörünge boyunca simülasyon sonuçlarının karşılaştırılması

Şekil 5.19 incelendiğinde, SIM-A dinamik modelin, SIM-B dinamik modele göre daha iyi cevap verdiği ve daha düşük hata ile robotun gerçekleştirdiği bacak yörüngelerini takip edebildiği görülmektedir. Grafiğe bakıldığında, her iki yöntem için 1. bacakta meydana gelen takip hatasının diğerlerine göre daha yüksek olduğu anlaşılmaktadır.

Yapılan model doğrulama deneyi sonucu, SIM-A ve SIM-B dinamik modellerin gerçekleştirdiği bacak uzunlukları hatalarına göre hesaplanan maliyet değerleri her bir bacak için Tablo 5.6'da verilmiştir.

Tablo 5.6. Deney 4'e göre dinamik modellerin maliyet fonksiyon deęerleri

Model	Bacak 1	Bacak 2	Bacak 3	Bacak 4	Bacak 5	Bacak 6
SIM-A	0.9325	0.4095	0.3781	0.3203	0.7251	0.3205
SIM-B	1.1611	0.4653	0.6904	0.3821	0.3313	0.5515

Tabloda verilen deęerlere bakıldığında, SIM-A dinamik modelin daha düşük maliyet deęerlerini başarabildięi grlmektedir. Ancak 5. bacakta verilen maliyet deęeri aısından SIM-A dinamik modelin bu bacakta gerekleřtirdięi yrnge takip doęruluęu SIM-B'ye gre daha azdır.

Verilen grafik ve tablodaki deęerler gzlemlendięinde, doęru bir Őekilde elde edilen dinamik modellerden SIM-A, kontrol performansının geliřmesine daha ok katkı saęlayacaęı anlařılmaktadır.

5.6.5. Deney 5

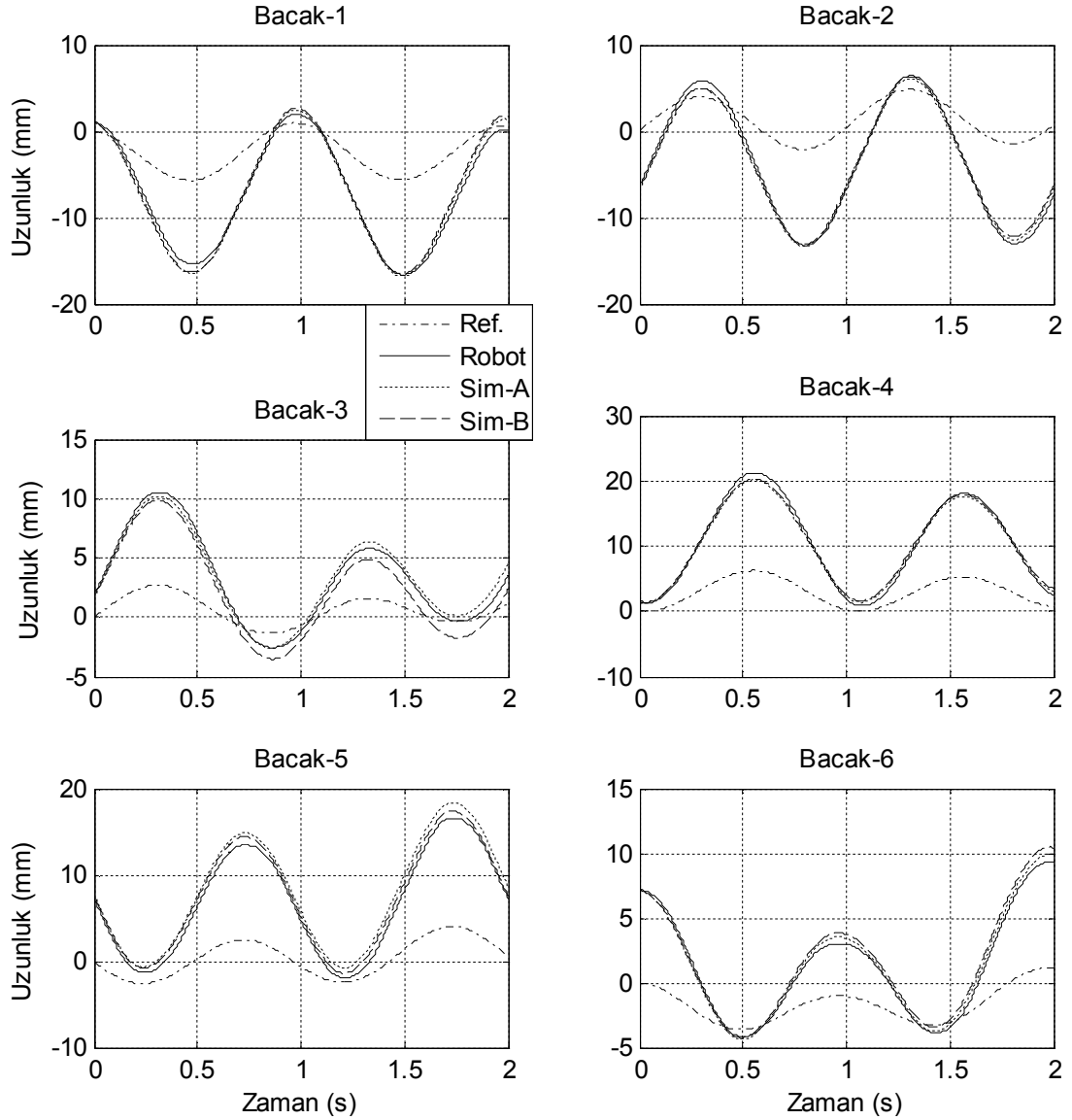
Bu deneyde hareketli platform, tm eksenlerde hem teleme hem de dnme hareketi yaptığında yntemlere gre elde edilen dinamik modellerin ve deneysel robotun gerekleřtirdięi bacak konumları gzlemlenecektir. Ařaęıda, hareketli platformun yrnge denklemi verilmiřtir.

$$\begin{aligned}x(t) &= 5 \sin(\pi t) & \alpha(t) &= 10 \cos(2\pi t) \\y(t) &= 5 \cos(\pi t) & \beta(t) &= 10 \sin(2\pi t) \\z(t) &= 0.25 + \sin(\pi t) & \gamma(t) &= 10 \cos(2\pi t)\end{aligned}\tag{5.158}$$

Őekil 5.20, SIM-A ve SIM-B dinamik modellerin gerekleřtirdięi ve deneysel robottan alınan bacak konumları ile robotun ters kinematik problemi zmnde elde edilen referans bacak uzunluklarını gstermektedir.

Őekil incelendięinde, simlasyon ve lm verilerinin birbirine yakın olduęu grlmekte ve her iki yntemin de, robotun gerekleřtirdięi bacak konumlarını bařarıyla takip edebildięi anlařılmaktadır. Grafikte 5. bacaęa bakıldığında, SIM-B dinamik modelin SIM-A dinamik modele gre daha az hata ile yrngeyi takip edebildięi grlmektedir.

Yapılan model doğrulama deneyi sonucu, SIM-A ve SIM-B dinamik modellerin gerçekleştirdiği bacak uzunlukları hatalarına göre hesaplanan maliyet değerleri her bir bacak için Tablo 5.7’de verilmiştir.



Şekil 5.20. Deney 5 için verilen yörünge boyunca simülasyon sonuçlarının karşılaştırılması

Tablo 5.7. Deney 5’e göre dinamik modellerin maliyet fonksiyon değerleri

Model	Bacak 1	Bacak 2	Bacak 3	Bacak 4	Bacak 5	Bacak 6
SIM-A	0.5258	0.4622	0.3974	0.5223	1.2380	0.3012
SIM-B	0.5722	0.4772	0.9447	0.5052	0.6898	0.5317

Tabloda verilen değerler açısından, SIM-A dinamik modeli, genel olarak daha düşük maliyet değerlerine sahiptir. Ancak dinamik modeller arasındaki maliyet farkı çok

azdır. Bunun dışında, 5. bacakta verilen değerlere göre meydana gelen takip hatasının, SIM-A'da büyük olurken SIM-B'de ise daha küçük olduğu anlaşılmaktadır. Şekil ve tabloda görülebildiği gibi, elde edilen simülasyon dinamik modelleri, birbirine yakın davranış sergileyerek benzer sonuçlar üretmiştir.

5.6.6. Deney 6

Bu doğrulama deneyinde, hareketli platform için yüksek frekanslı yörünge seçildiğinde SIM-A ve SIM-B dinamik modelleri ile deneysel robotun vermiş olduğu cevaplar karşılaştırılacaktır. Seçilen hareketli platformun yörünge denklemi aşağıda verildiği gibidir.

$$\begin{aligned}
 x(t) &= 10 \sin(2\pi t) & \alpha(t) &= 5 \cos(4\pi t) \\
 y(t) &= 10 \cos(3\pi t) & \beta(t) &= 5 \sin(5\pi t) \\
 z(t) &= 0.25 + 3 \sin(\pi t) & \gamma(t) &= 5 \cos(3\pi t)
 \end{aligned} \tag{5.159}$$

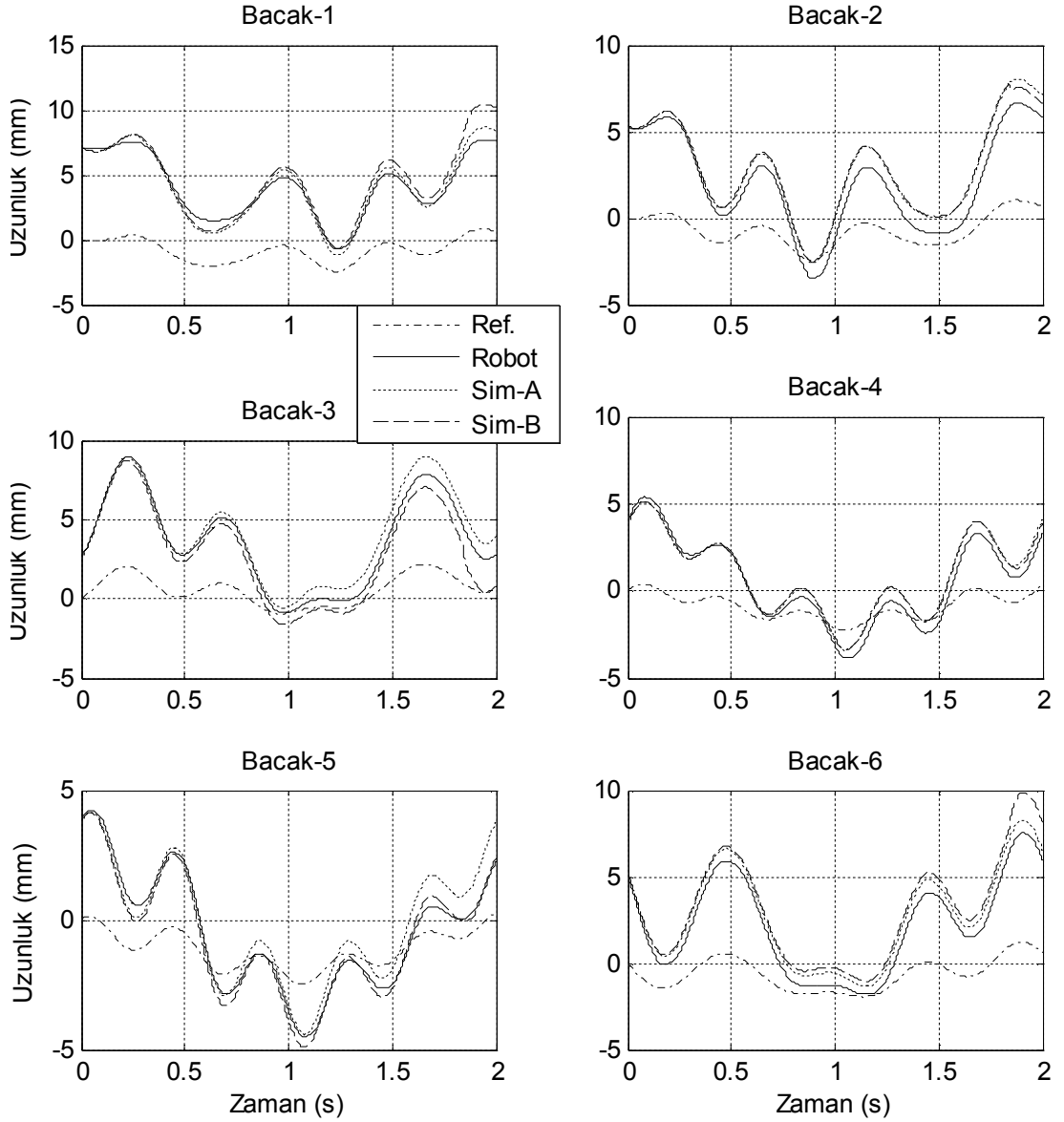
Verilen hareketli platformun yörünge denklemine göre ters kinematik sonucu elde edilen referans bacak uzunlukları, SIM-A ve SIM-B dinamik modelleri ile deneysel robotun gerçekleştirdiği bacak uzunlukları, Şekil 5.21'de gösterilmektedir.

Şekilde görülebildiği gibi, her iki dinamik modelde, robotun gerçekleştirdiği bacak yörüngelerini yakın bir şekilde takip edebilmektedir. Bunun yanında, dinamik modeller karşılaştırıldığında, SIM-A dinamik modelin vermiş olduğu cevapların daha iyi olduğu görülmektedir. Ancak şekilde de görüldüğü gibi 5. bacakta, SIM-B dinamik modelin çıkışı, deneysel robotun çıkışını daha yakın izleyebilmektedir.

Yapılan model doğrulama deneyi sonucu, SIM-A ve SIM-B dinamik modellerin gerçekleştirdiği bacak uzunlukları hatalarına göre hesaplanan maliyet değerleri her bir bacak için Tablo 5.8'de verilmiştir.

Tablo 5.8. Deney 6'ya göre dinamik modellerin maliyet fonksiyon değerleri

Model	Bacak 1	Bacak 2	Bacak 3	Bacak 4	Bacak 5	Bacak 6
SIM-A	0.3998	0.8512	0.5402	0.4673	0.4990	0.6340
SIM-B	0.6719	0.8591	0.6953	0.4556	0.2516	0.9915



Şekil 5.21. Deney 6 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması

Tablo 5.8’de verilen maliyet fonksiyon değerleri incelendiğinde, modellerin gerçekleştirdiği hataların küçük olduğu görülmektedir. Bu sonuç, her iki yöntemle göre doğru bir şekilde Stewart Platform dinamik modelinin çıkarılabildiği anlaşılmaktadır.

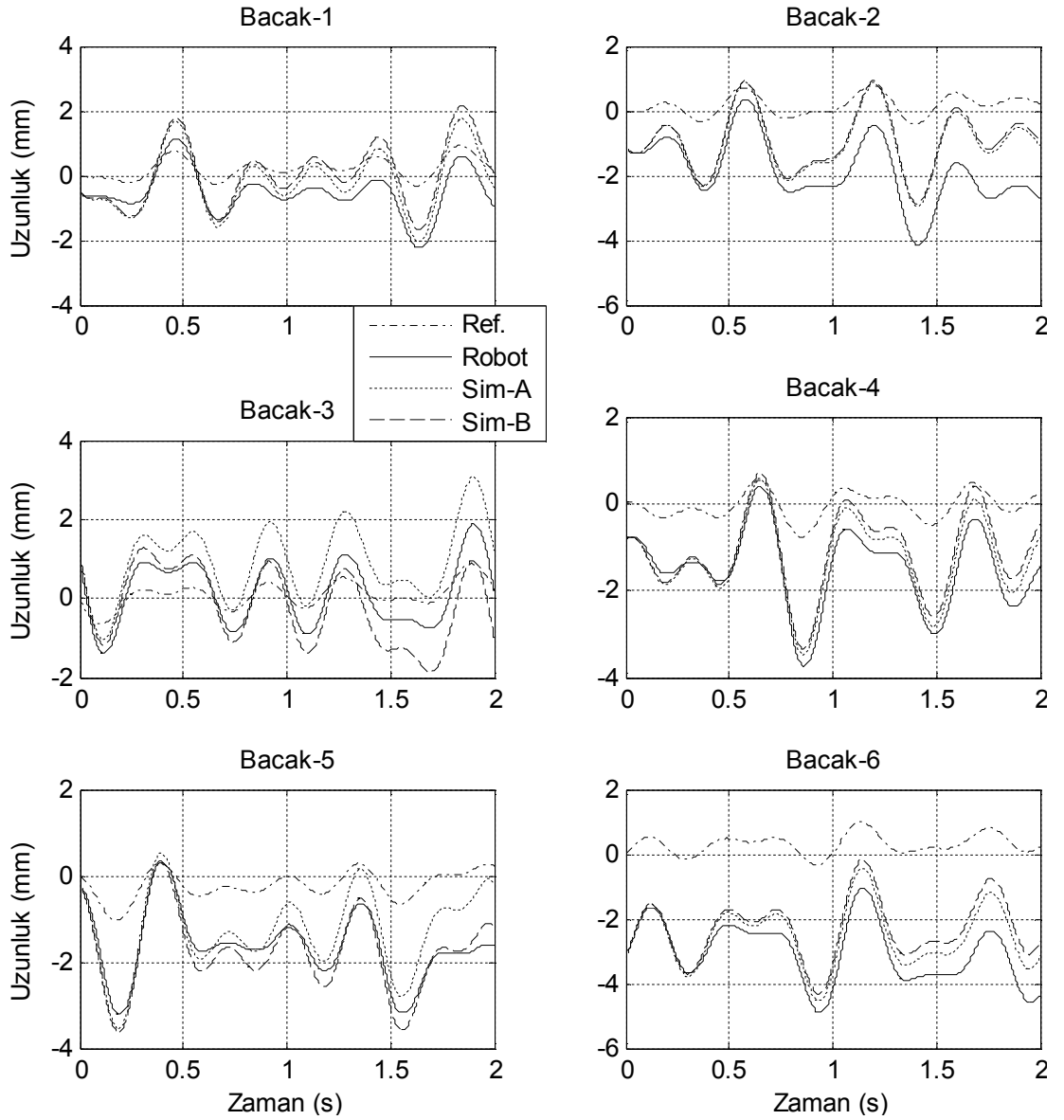
5.6.7. Deney 7

Bu doğrulama deneyinde, bir önceki deneye göre daha yüksek frekanslı yörünge seçilerek SIM-A ve SIM-B dinamik modelleri ile deneysel robotun vermiş olduğu

cevaplar incelenecektir. Seçilen hareketli platformun yörünge denklemi aşağıda verildiği gibidir.

$$\begin{aligned}
 x(t) &= 2 \sin(4\pi t) & \alpha(t) &= 3 \cos(6\pi t) \\
 y(t) &= 2 \cos(5\pi t) & \beta(t) &= 3 \sin(6\pi t) \\
 z(t) &= 0.25 + 2 \sin(3\pi t) & \gamma(t) &= 6 \cos(4\pi t)
 \end{aligned}
 \tag{5.160}$$

Yukarıda verilen yörünge denklemine göre ters kinematik sonucu elde edilen referans bacak uzunlukları ile deneysel robotun ve dinamik modellerin gerçekleştirdiği bacak uzunlukları Şekil 5.22’de gösterilmektedir.



Şekil 5.22. Deney 7 için verilen yörünge boyunca simülasyon sonuçların karşılaştırılması

Şekil incelendiğinde, yüksek frekanslı yörünge altında SIM-A ve SIM-B dinamik modelleri, robotun gerçekleştirdiği yörüngeleri takip edebilmiştir. Ancak grafikte de görüldüğü gibi izleme hataların büyük olduğu anlaşılmaktadır. Modeller karşılaştırıldığında, 3. ve 5. bacaklarda SIM-B dinamik modelin vermiş olduğu cevaplar daha iyidir. Bunun dışında diğer bacaklarda SIM-A dinamik modeli, robotun gerçekleştirdiği yörüngeyi daha yakın takip edebilmektedir.

Yapılan model doğrulama deneyi sonucu, SIM-A ve SIM-B dinamik modellerin gerçekleştirdiği bacak uzunlukları hatalarına göre hesaplanan maliyet değerleri her bir bacak için Tablo 5.9’da verilmiştir.

Tablo 5.9. Deney 7’ye göre dinamik modellerin maliyet fonksiyon değerleri

Model	Bacak 1	Bacak 2	Bacak 3	Bacak 4	Bacak 5	Bacak 6
SIM-A	0.4319	0.8666	0.7839	0.2584	0.4467	0.5114
SIM-B	0.6225	0.9334	0.4522	0.4098	0.2183	0.7323

Tablo 5.9’da verilen değerlere bakıldığında, genel olarak SIM-A dinamik modelin yörünge takip doğruluğu ve frekans artışına karşı duyarlılığı daha iyi olduğu görülmektedir. Diğer deneylerle karşılaştırıldığında, deneysel robotun gerçekleştirdiği bacak yörüngelerine benzer davranışı, SIM-A ve SIM-B dinamik modelleri, az bir hata artışı ile sergilemiştir.

SIM-A ve SIM-B dinamik model sistemleri için yapılan doğrulama deneylerinde meydana gelen bacak yörünge hatalarına göre tüm sistemin göstermiş olduğu maliyet değeri, aşağıda verilen denkleme göre hesaplanmıştır.

$$E_{sys}(k) = \frac{1}{N} \sum_{i=1}^N |e_1(i)| + |e_2(i)| + |e_3(i)| + |e_4(i)| + |e_5(i)| + |e_6(i)| \quad (5.161)$$

Denklemden yer alan $e_1(i) + e_2(i) + \dots + e_6(i)$, i . örnekteki eklemlerde meydana gelen hatayı (robotun gerçekleştirdiği ile modellerin gerçekleştirdiği bacak uzunlukları arasındaki fark), N , örnek sayısını ve k ise deney numarasını göstermektedir.

Yukarıdaki denklem kullanılarak SIM-A ve SIM-B dinamik modellerine göre tüm sistemin göstermiş olduğu maliyet değerleri, Tablo 5.10’da verilmektedir.

Tablo 5.10. Yöntemlere göre tüm sistemin maliyet fonksiyon değerleri

DENEY	SIM-A	SIM-B
1-EXP (α, β, γ)	4.7366	5.7904
2-EXP (x, y, z)	3.5413	4.9350
3-EXP (z)	4.8860	5.7756
4-EXP (β)	3.0861	3.5817
5-EXP ($x, y, z, \alpha, \beta, \gamma$)	3.4468	3.7209
6-EXP ($x, y, z, \alpha, \beta, \gamma$)	3.3916	3.9251
7-EXP ($x, y, z, \alpha, \beta, \gamma$)	3.2990	3.3685

Tablo incelendiğinde, her iki dinamik modele göre tüm sistemin göstermiş olduğu maliyet değerleri birbirine yakın olduğu görülmektedir. Ayrıca, tabloda verilen bu yörünge takip keskinliğinin sayısal dinamik cevapları, modellerin deneysel sonuçlarla herhangi bir kontrolör olmadan iyi bir eşleşme yapabildiğini göstermektedir.

Deneysel robotun ve dinamik modellerin referans bacak yörüngelerine göre açık çevrim hatalarını kullanarak Denklem (5.161)'e göre maliyet değerleri her bir deney için Tablo 5.11' de verilmektedir.

Tablo 5.11. Deneysel robot ve dinamik modellerin açık çevrim hatalarına göre maliyet fonksiyon değerleri

DENEY	ROBOT	SIM-A	SIM-B
1-EXP (α, β, γ)	40.7758	39.8335	38.7157
2-EXP (x, y, z)	37.8575	38.7473	38.9282
3-EXP (z)	42.3781	43.7965	43.4171
4-EXP (β)	26.4962	27.8837	28.8491
5-EXP ($x, y, z, \alpha, \beta, \gamma$)	39.2577	40.0266	39.3616
6-EXP ($x, y, z, \alpha, \beta, \gamma$)	17.5795	18.6464	18.4749
7-EXP ($x, y, z, \alpha, \beta, \gamma$)	9.5929	8.2264	8.1201

Tabloda verilen değerlere bakıldığında, dinamik modellerin, referans yörüngelerini tüm bacaklarda takip ederken gerçekleşen hatalar sonucu elde edilen maliyet değerleri, deneysel robotta oluşan maliyet değerlerine yakın olduğu görülmektedir. Sonuç olarak, yöntemlere göre elde edilen SIM-A ve SIM-B dinamik modelleri, tüm doğrulama deneylerinde birbirine yakın yörünge takip hataları yaparak hemen hemen benzer davranışlar sergilemişlerdir.

5.7. Sonuç

Bu bölümde, 6 serbestlik dereceli Stewart Platform manipülatörün ters kinematik problem çözümü ve hareket denklemleri açık bir şekilde elde edildi. Bu dinamik denklemler kullanılarak Matlab-Simulink ortamında modeller geliştirildi. Geliştirilen bu modelleri geçerli kılmak için farklı yörüngeli doğrulama deneyleri, hem gerçek robota hem de dinamik modellere uygulandı. Sonuçlar, tablo ve grafiksel olarak verildi.

İlk olarak, referans bacak uzunluklarını elde edebilmek için üst platformun konumu ve yönelimi verildiğinde bu bacak uzunluklarının ne olacağını belirleyen ters kinematik problemin çözümü yapıldı. Daha sonra Stewart Platform'unun katı gövde dinamik denklemlerini çıkarabilmek için Lagrange formülasyonu kullanıldı. Bu formülasyon ile önce hareketli platformun dinamiği sonra da bacakların dinamiği çıkarıldı. Bu formülasyonda kullanılan Jacobian matrislerinin çıkarımı için iki farklı yöntem kullanıldı. Çıkarılan iki farklı Jacobian matrisinden dolayı, iki farklı şekilde toplam kütle matrisi, Coriolis ve merkezkaç matrisi ve son olarak yerçekimi kuvvet vektörü elde edildi. Son olarak, bacaklarda bulunan eyleyicilerin elektriksel ve mekaniksel dinamikleri çıkarılarak elde edilen katı gövde dinamiği ile birleştirildi. Böylece, Stewart Platform mekanizmasının tüm sistem dinamiği iki farklı (SIM-A ve SIM-B) şekilde çıkarılarak gerçekleştirildi.

Elde edilen iki farklı dinamik modelin davranışlarını gözlemlemek amacıyla 7 farklı yörünge denklemini kullanarak doğrulama deneyleri yapılmıştır. Deneysel ve simülasyon sonuçları karşılaştırıldığında, dinamik modellerin (SIM-A ve SIM-B), deneysel robotun gerçekleştirdiği bacak yörüngelerini başarılı bir şekilde takip edebildiği görülmüştür. Sonuç olarak, Stewart Platform yapısının genel bir sınıfı için ters kinematik ve dinamik model elde edilebilmiştir.

6. STEWART PLATFORM'UN KONTROL ANALİZİ

6.1. Giriş

Stewart Platform'unda birbirine paralel kinematik bir zincir şeklinde bağlanmış eyleyiciler vasıtasıyla bacak uzunluklarını kısaltarak ya da uzatarak, hareketli platformun sabit platforma göre hareketi sağlanır. Bacak uzunluklarının uygun yörüngeleri ile hareketli platformun, yüksek doğrulukla karmaşık yörüngeleri takip etmesi mümkün kılınır. Karmaşık paralel bir mekanizma türü olarak Stewart Platform manipülatörlerin son derece doğrusal olmayan dinamiğe, geniş bir şekilde zamanla değişen parametrelere, eyleyicilerdeki sürtünme, bozucu etkiler ve yük değişiminden kaynaklanan çeşitli belirsizliklere ve de karmaşık bir yapıya sahip olmasından dolayı Stewart Platform manipülatörün kontrol tasarımı, zor bir problem olarak ortaya çıkmaktadır. Tüm bu belirsizlikler, doğrusal ve doğrusal olmayan kontrol yaklaşımlarıyla giderilebilir. Böylece, yüksek performanslı kontrol yaklaşımları, kayda değer bir şekilde paralel manipülatörlerin takip performansını geliştirir.

Stewart Platform manipülatörlerinde genel olarak, eklem uzayı kontrolü ve kartezyen (görev veya çalışma) uzayı kontrolü olmak üzere iki kontrol yapısı kullanılır. Eklem uzayı kontrol yapısı, tek giriş tek çıkışlı (SISO) bir sistem olarak her bir bacağın kontrol edilmesiyle geliştirilir. Gerçekleşen ve istenen eklem konumları arasındaki hata, kontrolöre geribesleme sinyali olarak kullanılır. Bu kontrolör yapısı, gerçekleşen bacak uzunluklarını, ters kinematik yardımıyla hareketli platformun kartezyen koordinatlarından hesaplanarak istenen bacak uzunluklarına uydurmak için tasarlanır. Burada, hareketli platformun koordinatları ile 6 bacağın uzunlukları arasında kesin bir ilişki vardır. Eğer bacak uzunlukları uygun bir şekilde kontrol edilirse bu paralel manipülatörün hareketli platformu istenen yörüngeleri takip edebilir.

Diğer kontrol yapısı ise kartezyen uzayı kontrolüdür. Bu kontrol yapısı, kartezyen uzayına göre elde edilen dinamiğe dayalıdır ve hareketli platformun kartezyen koordinatlarına gereksinim duyar. Bu hareketli platformun konum bilgisi, 6 serbestlik dereceli Kartezyen konum ve yönelim sensörlerinden ya da ileri kinematik gibi hareketli platformun konum bilgisini kestirebilen diğer yöntemlerden elde edilebilir. Ancak 6 serbestlik dereceli sensörler çok pahalı olabilir ya da istenilen doğruluklarda bulmak mümkün olmayabilir. 6 serbestlik dereceli paralel robot manipülatörlerin ileri kinematiği, her ne kadar sayısal yöntemlere bel bağlasa da dijital teknoloji gelişiminin ilerlemesinden dolayı birkaç milisaniye içerisinde çözülebilir. Böylece, gerçek zamanlı kontrolde ileri kinematiği kullanmak mümkün olur. Her iki kontrol yapısında üstünlük görünen yönleri ve sakıncaları vardır. Manipülatörün takip hataları doğrudan elde edilebildiği için kartezyen uzayı kontrolü tercih edilebilir. Fakat, hareketli platformun kartezyen koordinatlarını elde etmek zordur. Stewart Platform manipülatörün ters kinematik modelinin kapalı form çözümü olması nedeniyle eklem uzayı kontrolü, basit bir yapıya sahip olsa da, bu kontrol şeması yüksek genlikli manevra olduğu durumlarda yüksek takip performansını sağlayamayabilir.

Platform kontrol tasarımında önemli meselelerden birisi, kontrol kuralının seçimidir. Oransal (P), oransal-integral (PI), oransal-türev (PD) ve de oransal-integral-türev (PID) gibi birkaç model arasında en iyisinin seçilmesi düşünülür. Klasik bir PID kontrolörü endüstride yaygın bir şekilde kullanılmaktadır. Ancak paralel bir manipülatör için yüksek performansı her zaman garanti edemez. Bu yüzden, kontrol performansını arttıracak ileri düzeyli bir kontrol yapısına gereksinim duyulur. Paralel manipülatörlerin son derece doğrusal olmayan bir yapıya ve belirsizliğe sahip olması gelişmiş bir kontrol algoritmasının tasarlanmasını zorlaştırır. Uygulamalara bakıldığında, paralel bir manipülatörün yörünge takip keskinliğini arttırmak için bazı yeni modern kontrol yöntemlerin eklenmesiyle geleneksel PID kontrol, hala temel bir kontrol yapısı olduğu görülmektedir.

Genel bir 6 serbestlik dereceli paralel manipülatörün kontrolünü yüksek duyarlılıkla gerçekleştirecek bir doğrusal olmayan PID algoritması kullanılabilir. Sönümlemeyi arttırma ve izleme hatasını azaltma gibi alanlarda geliştirilmiş performans için

kontrol kuralını sentezleyecek kontrol hatası üzerinde etkili olan oransal, integral ve türevsel parametrelerin doğrusal olmayan bir birleşimi kullanılır. Bilindiği gibi doğrusal birleşimli bir PID, sistemin cevap hızı ve kararlılığı açısından makul bir performansı başarabilir. Bunun yanında, ileri düzeyde sistem performansını başarmak için doğrusal olmayan bir PID, ekstra serbestlik derecesi sağlayabilir. Hata üzerinde etkili olan oransal, integral ve türevsel parametrelerin doğrusal olmayan bir birleşimini kullanarak yüksek duyarlılık sağlanabilir.

Paralel manipülatörlerin asıl harici bozucuları, yük bozucularıdır ve sistemin performansı üzerinde her zaman oldukça etkilidir. Sistem yükü belirsiz ve değişken olduğunda bir kontrolör için aynı anda doğruluğu ve sağlamlığı garanti etmek zordur. Araştırmaların birçoğu, harici yük bozucuları, sürtünme etkileri gibi doğrusal olmayan sistem dinamiği ve belirsizliklerden meydana gelen zıt etkilere karşı aynı anda kendini uyarlayabilen sağlam kontrolör tasarımı üzerine yapılan çalışmalardır. Stewart Platform manipülatörün hareket denklemlerindeki doğrusal olmayan etkileri ve belirsizlikleri gidererek kontrol performansını arttırmak için literatürde birçok kontrol yaklaşımı sunulmaktadır. Stewart Platform manipülatöründe ya model tabanlı kontrol ya da hareket kontrolü şeklinde uygulanan kontrol stratejileri; uyarlamalı kontrol [151], ters dinamik kontrol [152-156], doğrusal olmayan PID ve PD kontrol [157-162], kayma kipli kontrol [163-170], bulanık mantık kontrol [171-178] ve yapay sinir ağı kontrol [179-180] yöntemleridir. Buradaki amaç tasarlanabilen sağlam bir kontrolör yapısını oluşturmaktır.

Nguyen ve arkadaşları [151], Lyapunov yöntemini kullanarak elektromekanik olarak sürülen Stewart Platform manipülatörü için eklem uzayında adaptif bir kontrol yapısı sunmuşlardır. Bu adaptif kontrol yapısı, oransal-türev (PD) kontrolörlerinden oluşmaktadır. Kontrolörlerin kazançlarını, eyleyicinin yörünge takip hatası ile sürülen bir uyarlama kuralıyla ayarlamışlardır. Ayrıca, adaptif kontrol yapısının performansını değerlendirmek için değişken yük altında platformun, verilen dik ve dairesel yörüngeleri takip etmesi için deneysel bir çalışma yürütmüşlerdir. Çalışmada sunulan deneysel sonuçlar, önerilen kontrol yapısının, sabit kazançlı kontrolörlere göre daha üstün takip performansını başarabildiğini göstermiştir. Davliakos ve arkadaşları [152], 6 serbestlik dereceli elektrohidrolik bir Stewart-Gough Platform

manipülatorü için model tabanlı yeni bir kontrolör yapısı geliştirmişlerdir. Geliştirilen geribeslemeli kontrolör, platformun katı gövde dinamik denklemlerini ve servo-eyleyici sistemin hidrolik dinamiğini çalıştırır. Burada, kontrolöre sürülecek kartezyen uzayındaki yörünge hatasını üretmek için ileri kinematik, geribeslemeli kontrol döngüsü içerisine gömülür. Kontrolör, bu hatayı kullanarak servovalflere giriş akımı üretir. Makalede yapılan simülasyon çalışmasında, istenilen yörünge girişleri ile simülasyonlar yapılarak önerilen kontrolörün tatmin edici bir performans sergilediği gösterilmiştir.

Fu ve Yaou [153], kısmi ters dinamik dengelemeli Stewart Platform manipülatorün istenilen bir yörüngeyi takip etmesi için kartezyen uzayında, sağlam doğrusal olmayan bir kontrolör önermişlerdir. Ayrıca, bacakların etkisini karşılamak için Newton-Euler yöntemiyle elde edilen ters dinamik formülasyonunu kullanmışlardır. Önerilen bu takip kontrolörün, model hatasını ve bozucu etkileri içeren belirsizliklerin üstesinden gelebildiğini göstermişlerdir. Ayrıca, PID kontrolör ile önerilen kontrolör arasında performans karşılaştırması yapmışlardır. Sunulan simülasyon sonuçları, önerilen kontrolörün, PID kontrolörden daha ufak yörünge hatasını ve kuvvet girişini ürettiğini göstermiştir. Lee ve arkadaşları [154], 6 serbestlik dereceli bir UPS Stewart Platform manipülatorü için H_{∞} kontrol ile birleştirilmiş yaklaşık dinamikli ters dinamik kontrol yapısını önermişlerdir. Burada, yaklaşımdan kaynaklanan hataları gidermek için H_{∞} kontrolörü sunulmuştur. Bundan başka makalede, basit bir PID kontrolör ve tam dinamikli ters dinamik kontrolör yapılarını kullanarak karşılaştırma yapmışlardır. Yapılan deneysel sonuçlar, tam dinamikli ters dinamik kontrol yapısına ve PID kontrolöre göre, önerilen yaklaşık dinamikli ters dinamik kontrol yapısı, yüksek frekanslı referans yörüngelerini daha iyi takip edebildiğini göstermiştir. Ghobakhloo ve arkadaşları [155], Stewart-Gough Platform manipülatorün yörünge takibi için manipülator dinamiğinin bağlaşım etkisinin azaltılmasında ve doğrusallaştırılmasında kullanılan ters dinamik kontrol (hesaplamalı tork kontrol) yapısının bir uygulamasını sunmuşlardır. Ters dinamik kontrol yapısının verimliliğini göstermek amacıyla hareketli platformun tüm eksenlerde hem dönme hem de öteleme hareketi yapan bir yörünge seçerek bir simülasyon çalışması yaptılar. Seçilen oransal ve türev kazanç matrislerini kullanarak yapılan bu sayısal simülasyon çalışmasında, önerilen kontrol

yapısının geçerliliğini gösterdiler. Zubizarreta ve arkadaşları [156], 6 serbestlik dereceli Stewart Platform manipülatörü için yörünge takip hatasını azaltmada ve model parameter belirsizliklerine karşı sağlamlığı arttırmada kullanılacak extra sensörlerle sağlanan verinin doğrudan kontrolöre sunulmasıyla tasarlanan genişletilmiş hesaplamalı tork kontrolör yapısını önermişlerdir. En iyi performansı elde etmek için doğru bir ekstra sensör dağılımının gerekli olduğunu belirtmişlerdir. Ayrıca makalede, farklı extra sensör dağılımlarını değerlendirmek için yaklaşıma dayanan bir duyarlılık analizini, manipülatöre uygulamışlardır. Önerilen yaklaşımın geçerliliğini göstermek için 3 farklı referans yörüngesine göre hem eklem hem de kartezyen uzaylarında çalışmalar yapmışlardır. Elde edilen sonuçlar, önerilen yöntemin verimliliğini göstermiştir.

Su ve arkadaşları [157], genel bir 6 serbestlik dereceli Stewart Platform manipülatörün yüksek keskinlikli yörünge takip kontrolünü gerçekleştirmek için otobozucu etkisini giderecek eklem uzayında tasarlanan basit sağlam bir kontrol yapısı önermişlerdir. Bu kontrol yapısı, sistemde gürültü ve diğer bozucu etkiler olduğunda daha etkili ve sağlam performans için geleneksel PID kontrolörün türev kısmını gösteren doğrusal olmayan bir takip türevliyecisinden ve de bu kontrol yapısının üstün bir performans vermesi için doğrusal olmayan bir PD kontrolörden oluşmaktadır. Makalede önerilen kontrol yönteminin kolay uygulanabilirliğini ve verimliliğini geçerli kılmak için simülasyonlar ve deneysel çalışma sonuçları sunulmuştur. Verilen deneysel sonuçlar, Stewart Platform manipülatörü için yüksek takip keskinliğinin ve sağlamlığın başarılabildiğini göstermiştir.

Yang ve arkadaşları [158], 6 serbestlik dereceli hidrolik Stewart-Gough paralel platformun doğrusal olmayan karakteristiğini göz önünde bulundurarak kontrol performansını arttıracak ve kalıcı durum hatalarını giderecek bir model tabanlı kontrolör sunmuşlardır. Ağır yüklü 6 serbestlik dereceli hidrolik bir manipülatörde, yerçekimi yönünde büyük kalıcı durum hataları ve manipülatörün dinamik hataları meydana geldiğinden bu yerçekimi yönündeki hataları giderecek ve azaltacak dinamik yerçekim karşılamalı bir PD kontrolörü tasarlamış ve gerçekleştirmişlerdir. Ayrıca, önerilen kontrolörün kalıcı, doğruluk ve sağlamlık açısından performansını, teorik ve simülasyonla analiz etmişlerdir. Yapılan analiz çalışmaları, önerilen

kontrolörün hidrolik Stewart-Gough platformu için mükemmel bir performans sergilediğini, kararlı olduğunu ve de sistemin ağır yükünden dolayı sistemde meydana gelen kalıcı durum hatalarını sıfıra yaklaşımlı olarak yakınsadığını göstermiştir. Su ve arkadaşları [159], 6 serbestlik dereceli genel bir paralel manipülâtörün yüksek doğruluklu yörünge kontrolünü başarmak için eklem uzayında doğrusal olmayan bir PID kontrol algoritmasını geliştirmişlerdir. Bozucu etkiler ve ölçüm gürültüsü olduğunda yüksek kaliteli türev sinyallerini üretmek için geliştirilen bu PID kontrolör, doğrusal olmayan 2 takip türevliyicisini kullanır. Oransal, integral ve türevsel kısımların doğrusal olmayan birleşiminden oluşan bu kontrol yapısını kullanarak kolayca uygulanabildiğini ve yüksek takip doğruluğunu garanti edebildiğini göstermişlerdir. Ayrıca, deneysel sonuçlarla doğrusal olmayan PID kontrolörün verimliliğini geçerli kılmışlardır. Verilen deneysel sonuçlar, önerilen kontrol algoritmasının, yörünge takip performansını geliştirdiğini ve mühendislik uygulamalarında kolayca gerçekleştirilebileceğini göstermiştir.

Omran ve arkadaşları [160], 6 serbestlik dereceli Stewart Platform manipülâtörün eklem uzayında kontrolü için genetik algoritma ile kazançları ayarlanan bir PD kontrol yapısını sunmuşlardır. Kontrol yapısını test etmek amacıyla üç boyutlu dairesel hareket simülasyon deneyi yapmışlardır. Ayrıca, parameterleri ayarlanan kontrolörün, değişken yük altında sağlamlığı ve algoritmanın verimliliği, verilen simülasyon sonuçlarıyla gösterilmiştir. Kassem ve Omran [161], Stewart Platform manipülâtörün kartezyen uzayında ideal bir kontrolünü tasarlamak için üst platformun tüm manevralarına göre genetik algoritma ile kontrol kazançları ayarlanan bir PD kontrol algoritmasını önerdiler. Yapılan deneysel sonuçlarda, bu en iyi kazançlara göre kontrol edilen Stewart manipülâtörün vermiş olduğu cevaplar, ölçülen ile istenilen konum ve hız değerleri arasında yüksek bir eşleşme olduğunu göstermiştir. Sun ve arkadaşları [162], 3-2-1 yapılı Stewart Platform manipülâtörün kartezyen uzayında PID kontrolü için parçacık sürü optimizasyonu ile klonal seçim algoritmasını (biyolojik bağışık sistemindeki eşeysiz çoğalan bireyler) birleştiren bir akıllı optimizasyon algoritmasını önerdiler. Ayrıca, önerilen kontrolörün kontrol performansını karşılaştırmak amacıyla PSO ve klonal seçim algoritmalarına göre iki PID kontrolör daha tasarladılar. Hareketli platforma y-ekseninde verilen birim basamak sinyali ile kontrolörlerin performanslarını karşılaştırdılar. Verilen

simülasyon sonuçları, önerilen kontrol algoritmasının daha iyi performansa sahip olduğunu göstermiştir.

Kim ve arkadaşları [163], 6 serbestlik dereceli bir Stewart Platform manipülatörün yüksek hızlı yörünge takip kontrolü için performansı genişletilmiş kayma kipli bir kontrolör geliştirmişlerdir. Bu genişletilmiş kayma kipli kontrolü, bozucu etkeni karşılama ve ulaşma fazını yumuşatma fonksiyonlarını, geleneksel kayma kip kontrolüne dahil ederek oluşturmuşlardır. Önerilen kontrol yapısının performansını gözden geçirmek için verilen yörüngeler altında kontrol deneyleri yapılmıştır. Deneysel sonuçlar, yüksek yük ve bozucu etkenler altındaki Stewart Platform sistemi için yüksek performanslı yörünge takip kontrol sisteminin tasarlanabildiğini göstermiştir. Huang ve arkadaşları [164], 6 serbestlik dereceli hidrolik Stewart Platform manipülatörün sistem parametrelerindeki belirsizlikler altında yüksek performanslı takip kontrolünü başarmak için kayma kipli bir kontrol yaklaşımını sunmuşlardır. Kontrol performansını göstermek amacıyla x-y düzleminde dairesel ve de y ekseninde dönme hareketini sağlayan yörünge deneylerini yapmışlardır. Ayrıca, tasarlanan kontrolörün kararlılığını garanti etmek için Lyapunov teorisine dayanan kararlılık analizini yapmışlardır. Sonuç olarak, sıfıra yaklaşmalı olarak yakınsayan takip hatalarını sürülebilir kontrolörün verimliliğini deneysel sonuçlar ile teyit etmişlerdir.

Wu ve arkadaşları [165], 6 serbestlik dereceli elektro-hidrolik Stewart Platform manipülatörü için bulanık ayarlamalı sağlam bir integral kayma kipli kontrol yapısını geliştirmişlerdir. Bu kontrol yapısı, parametre değişimi ve harici bozucular olduğu durumda etkiyi giderecek kayma kipini devreye sokan bir anahtarlama kontrolünden ve de hızlı yörünge takibini arttıracak ve istenmeyen yüksek frekans ya da modellenmemiş dinamiklerin etkisini azaltacak bulanık kontrolden oluşmaktadır. Makalede, önerilen kontrol algoritmasının performansını değerlendirmek için iyi ayarlanmış PID kontrol ile birlikte yörüngeleri takip deneyleri yapılmıştır. Simülasyon ve deneysel sonuçlardan, önerilen kontrol algoritmasının, yörüngeleri takip kabiliyetinin daha iyi olduğunu ve parametrik belirsizlikler ve yük değişimlerine karşı dayanıklı olduğu ve de dışarıdan gelen bozucu etkenlere karşı kararlılık sergilediği anlaşılmaktadır. Ayrıca, kayma kip kontrolöründe kalıtımsal

olan kayma yüzeyi civarında oluşan yüksek frekanslı salınımlar, bu algoritma sayesinde, kayda değer bir şekilde azaltılmıştır. Dongsu ve Hongbin [166], uçuş simülatörü olarak kullanılan 6-3 yapılu bir Stewart Platform mnaipülatörün parametre belirsizlikleri ve harici bozucular altında yüksek hassasiyetli hareket kontrolünü başarmak için kartezyen uzayında doğrusal olmayan uyarlamalı kayma kip kontrol yapısını önermişlerdir. Bu kontrol yapısında, kayma kip kontrol, zamanla değişen parametreler ve harici bozucularla başa çıkarken doğrusal olmayan öz-uyarlamalı kontrol ise çevirim-içi olarak belirsiz sabit parametreleri tanımlamak için kullanılır. Önerilen kontrol yapısının verimliliğini geçerli kılmak ve de harici bozucular ve belirsizliklere karşı sağlamlığını değerlendirmek amacıyla bilgisayar simülasyonları yapılmıştır. Simülasyondan elde edilen sonuçlar, önerilen kontrol algoritmasının, platform yükleri, atalet momentleri, kütle merkezi gibi hareketli platformun parametrelerini tam olarak teşhis edebildiğini ve de belirsiz parametreler ve harici bozucular olduğunda hassas hareket kontrolünü başarabildiğini göstermiştir.

Guo ve arkadaşları [167], 6 serbestlik dereceli hidrolik bir paralel robot manipülatörün yörünge takip kontrolünü gerçekleştirmek için eklem uzayında, kayma kipine dayanan bir ardarda (kaskat) kontrol algoritmasını sundular. Bu kontrol algoritması, eyleyici yük basıncını kontrol eden bir iç döngüden ve de eyleyici yükünün kontrolünü ve sistemin harici bozucularını gideren kayma kip kontrollü bir dış döngüden oluşmaktadır. Makalede, bu deneysel paralel robot manipülatörünü kullanarak önerdikleri kontrol algoritmasının verimliliğini geçerli kılmak için iyi ayarlanmış ileri beslemeli oransal kontrol ile karşılaştırmalı çalışmalar yapmışlardır. Bu çalışmalara göre verilen deneysel sonuçlar, ileribeslemeli iyi ayarlanmış oransal kontrolörle karşılaştırıldığında, kayma kipli kaskat kontrolörün daha üstün konum takip performansını başarabildiğini göstermiştir.

Negash ve Mitra [168], 6 serbestlik dereceli Stewart Platform manipülatörün eklem uzayında kontrolünü sağlamak için tasarlanan integral kayma kipli bir kontrolörün değişken yük altında kararlılık ve performans analizini sunmuşlardır. Ayrıca, integral kayma kipli kontrolörün performansını, geleneksel PID ve kayma kipli kontrolörler ile karşılaştırmışlardır. Bunun yanında, kontrolörleri test etmek için kartezyen uzayında hızlı bir yörünge seçerek simülasyon çalışması yapmışlardır. Simülasyon

sonuçları, önerilen kontrolörün daha üstün takip performansını ve belirsizliklere karşı sağlamlığını ortaya koymuştur. Ayrıca, makalede, bu kontrolün ürettiği kontrol sinyalinin daha yumuşak olduğu ve kayma yüzeyi civarındaki salınımları azalttığını belirtmişlerdir. Pi ve Wang [169], belirsiz yük bozucuları altında 6 serbestlik dereceli hidrolik bir paralel manipülatörün kartezyen uzayında yörünge takip performansını geliştirmek için uyarlamalı kontrol kurallarına dayanan bir kayma kipli kontrol yapısını önermişlerdir. Ayrıca, önerilen kontrol yapısını test etmek ve verimliliğini geçerli kılmak için kuvvet sensörlü doğrusal olmayan kontrol, kuvvet gözlemcili doğrusal olmayan kontrol ve iyi ayarlanmış PID kontrol yapılarını sunarak simülasyon ve deneylerle karşılaştırma yapmışlardır. Simülasyon ve deneysel çalışmalar, belirsiz yük bozucular olduğunda istenilen yörünge takibini, önerilen kontrol yapısının daha iyi bir performansla gerçekleştirdiğini göstermiştir.

Fraguela ve arkadaşları [170], uzaktan gizlice izlemeli bir cihaz olarak kullanılan 3 serbestlik dereceli Stewart Platform manipülatörü için yalnızca konum verildiğinde hareketli platformu dengede tutmak için sağlam tam bir kontrol yapısını tasarladılar. Bunun için, konum ölçümlerinden bilinmeyen harici sarsımları tanımlamayı ve hızları yeniden düzenlemeyi sağlayan yüksek dereceli kayma kipli bir gözlemci geliştirmişlerdir. Bu yeniden düzenleme ve sarsımı tanımlama durumlarını kullanarak belirsizlikleri giderecek kontrol kuralını tasarlamışlardır. Sonuç olarak, 3. dereceden kayma kipli gözlemci için elde edilen bilgi, platformun kontrol kuralında hızları yeniden düzenleme ve sarsınımı dengeleme için en uygun olduğu sonucuna varmışlardır.

Chung ve arkadaşları [171], 6 serbestlik dereceli hidrolik Stewart Platform manipülatörün eklem uzayında konum kontrolü için kararlılık analizli bulanık kontrolör geliştirmişlerdir. Tasarlanan kontrolörün kararlılık analizi için Popov kriter kararlılık teorisini kullanmışlardır. Deneysel sonuçlar, önerilen tasarım yapısının geçerliliğini ve performansını yeterince doğrulamıştır.

Mann ve Surgenor [172], 6 serbestlik dereceli Stewart-Gough Platform manipülatörün eklem uzayında kontrolü için 3 boyutlu bulanık bir PID kontrolör yapısını önermişlerdir. Her biri 7 üyelik fonksiyon içeren hata, hatanın değişimi ve

bu deęişimin karesi řeklinde giriřleri olan 7x7x7'lik kural tabanına sahip bir bulanık PID kontrol yapısını tasarlamıřlardır. Bunun yanında, dene-ve-hata (trial-and-error) yaklařımını kullanarak doęrusal bir PID kontrolörün kazanç parametrelerini ayarlamıřlardır. Kontrolörleri test etmek amacıyla, hareketli platformun z ekseninde basamak cevabı, daire hareketi ve de bu daire yörüngesi sırasında platform aęırlıęını iki katına ıkaran bir yükün eklenmesiyle 3 simülasyon testini yapmıřlardır. Simülasyon sonuçları, bulanık PID kontrolör, doęrusal PID kontrolör ile karřılařtırıldıęında hem geici hem de kalıcı durum takip performansını geliřtirebildięini göstermiřtir. Ayrıca bozucu yük altında, üç boyutlu bulanık kontrolörün kullanılması, geleneksel iki boyut tipli bulanık kontrolörlerin kullanılmasından daha dayanıklı bir kontrol sunduęunu göstermiřlerdir.

Remillard ve Boukas [173], Gough-Stewart Platform manipölatörün kontrolünde saęlam bir kontrolörün tasarımı ve analizini yerine getirmek için Takagi ve Sugeno (T-S) bulanık modelini önermiřlerdir. Bu kontrolör yapısında sıfır kalıcı durum hatasını elde etmek için, durum geribeslemeli bir integratör ekleyerek bulanık modeli geniřletmiřlerdir. Ayrıca, verilen bilgisayar simülasyon örneęi ile paralel manipölatör için önerilen yöntemin geçerlilięini ve etkinlięini göstermiřlerdir.

Wu ve arkadaşları [174], eyleyici doyumundan kaynaklanan kontrolör performansının bozulmasını ve yörünge sapmasını gidermek için Stewart Platform manipölatörün yeni bir uyarlamalı bulanık yörünge takip algoritmasını önerdiler. Bu algoritma, gerçek zamanlı yörünge biçimlendiricisi ile eklem uzayında uyarlamalı bulanık kontrolörden oluřmaktadır. Kartezyen uzayında verilen referans yörüngesi için eyleyicilerin herhangi birisi ne zaman doyuma ulařırsa referans yörüngenin istenilen ivmesi, platformun dinamik modeli kullanılarak çevrimii düzeltileceęini belirtmiřlerdir. Giriř olarak düzeltilen bu yörüngeyi kullanarak eklem uzayında hesaplamalı tork kontrolör, bacak hız gözleyicisi ve bulanık bozucu gözleyicisinden oluřan bir kontrolör geliřtirmiřlerdir. Ayrıca, sinüs dalgalı bir referans yörüngesini hem X eksenine hem de Y eksenine uygulayarak ve 100N'luk bir harici bozucu etkiyi sisteme dahil ederek bir simülasyon alıřması yapmıřlardır. Yalnızca konum ölçümleri ve bozucu etki olduęunda, önerilen kontrolörün yörünge takip performansı

ve kararlılığı sağlanabilmiştir. Simülasyon sonuçları, önerilen kontrol yapısının verimliliğini doğrulamıştır.

Hsu ve arkadaşları [175], 6 serbestlik dereceli elektriksel Stewart Platform manipülatörün eklem uzayında, maksimum anlık ivmeyi ve doğru konumlama performansını başarabilmesi için fonksiyonel sinirsel bulanık kontrolöre dayanan bir kullanıcı kontrolör arayüzünü geliştirmişlerdir. Fonksiyonel sinirsel bulanık kontrol modelini oluşturmak için çevrimiçi yapısal geliştirme ve parametre öğrenme algoritmalarını kullanmışlardır. Yapısal geliştirmede, uygun giriş uzayı bulanık bölüntüleri bulunmuş ve bulanık mantık kurallarını ortaya koymuşlardır. Gözlemci bir öğrenmeye dayanan parametre öğrenmede ise geriye yayılma algoritmasını kullanarak üyelik fonksiyonların parametrelerini ve sonuç kısmındaki bağlantı ağırlıklarını ayarlamışlardır. Makalede, PID kontrolör ve önerilen kontrolörün doğruluklarını ve kararlılıklarını karşılaştırmak için gerilim sürümlü eyleyicilere adım fonksiyonu, karesel eğri ve sinüs dalgalı şekilde gerilim uygulamışlardır. Deneysel sonuçlara göre, daha geçerli yörünge takibi açısından önerilen kontrolörün daha üstün olduğu gözlemlenmiştir.

Zhu ve arkadaşları [176], doğrusal olmayan, belirsiz karakteristiklere sahip ve harici bozucu etki altında 6 serbestlik dereceli Stewart Platform manipülatörün kontrolü için kayma kipli kontrole dayanan bir bulanık destek vektör mekanizmalı kontrol yapısını önermişlerdir. Bu kontrol yapısını, kayma kipli kontrolde meydana gelen salınımları azaltmak için tasarlamışlardır. Ayrıca, en küçük kareler ve geliştirilmiş genetik algoritmayı birleştiren bir karma öğrenme algoritmasını kullanarak önerilen kontrolörün parametrelerini en iyilemişlerdir. Optimize ettikleri kontrolör ile daha iyi bir kontrol sistemi elde etmişlerdir. Önerilen kontrol yönteminin geçerliliğini araştırmak için manipülatör üzerinde deneysel simülasyon yapmışlardır. Simülasyon sonuçları, tasarlanan kontrol yönteminin yüksek keskinlik ve hızla yörünge takibini elde edebildiğini ve de model hatası ve harici bozucuların olduğu durumda kayma yüzeyi civarındaki salınımları azaltabildiğini göstermiştir.

Bo ve arkadaşları [177], değişen yük durumu altında Stewart Platform manipülatörün dayanıklılığını, kararlılığını ve yörünge takip kabiliyetini elde etmek için PID kontrol

algoritması ile birleştirilen bulanık mantık kontrolüne dayanan bir kontrol algoritmasını geliştirmişlerdir. Hata ve hatanın türevi şeklinde iki girişe ve de PID kontrolörün oransal, integratör ve türev parametrelerini ayarlayan üç çıkışa sahip bir bulanık PID kontrolör şeklinde tasarlamışlardır. Ayrıca, kontrol altında sisteme, basamak ve kare dalga girişleri uygulamışlardır. Bunun yanında, aynı girişleri kullanarak sistemin performansı üzerinde bozucunun etkisini test etmek için harici bir bozucu kuvveti kullanmışlardır. Tasarlanan kontrolörün, sistemi daha dayanıklı yapabildiğini ve de bu kontrolör ile Stewart Platform manipülatörün güçlü dayanıklılık, yüksek doğruluk ve üstün performans gereksinimlerinin karşılanabileceğini simülasyon sonuçları ile göstermişlerdir.

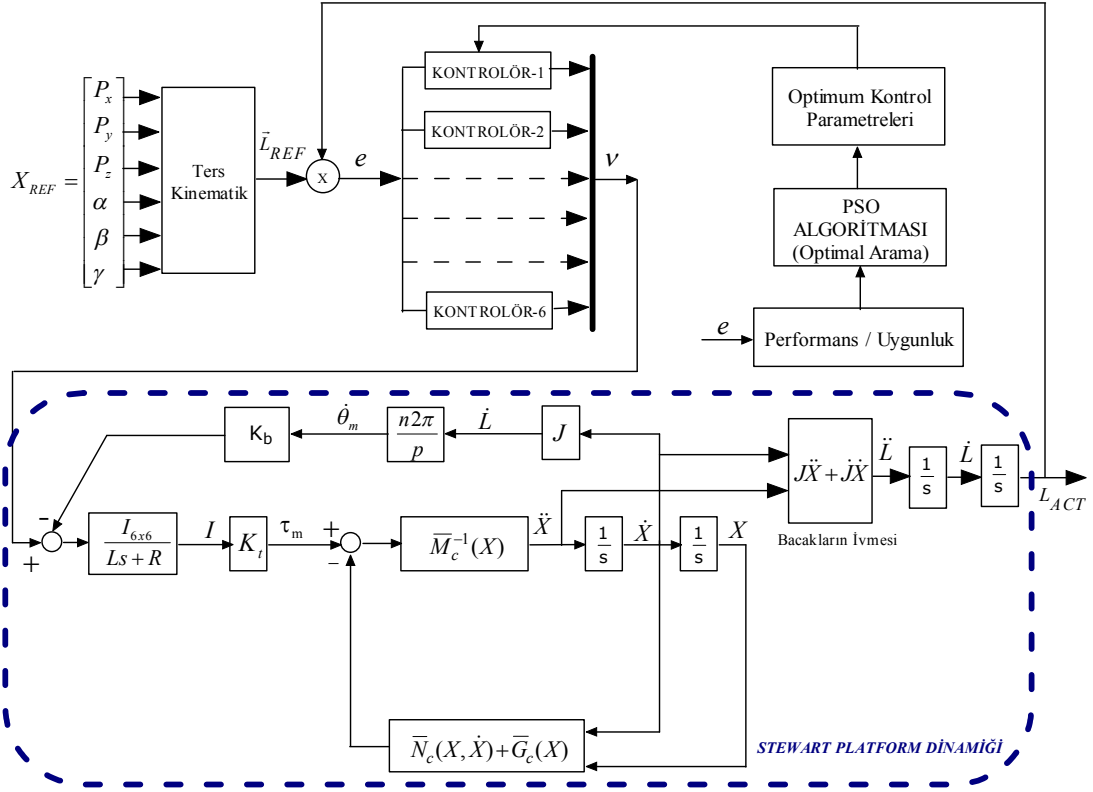
Ömürlü ve Yıldız [178], uzaysal hareket eden araçları, tek noktadan, kuvvet/tork geri beslemeli, kumanda edebilmek amacıyla tasarlanmış deneysel bir 3x3 bir Stewart Platform paralel mekanizmasının kuvvet/tork kontrolü için, katılık kontrolü tabanlı kazanç ayarlamalı bulanık PD (STFPD+PD) kontrol algoritmasını tasarlamışlardır. Ayrıca tasarlanan paralel bağlı, kazanç ayarlamalı bulanık PD ve PD kontrol için, uygun başlangıç kontrol katsayıları bulunarak, diğer alternatif kontrol algoritmalarıyla (PD, bulanık PD ve kazanç ayarlamalı bulanık PD) kuvvet/tork cevaplarını karşılaştırmışlardır. Bunun yanında, kontrol tasarım denemeleri için deneysel platformla simülasyon ortamındaki cevapların, basit katılık kontrolü uygulanarak karşılaştırılmasıyla, model uyumluluğunu göstermişlerdir. Sonuç olarak, makalede, paralel bağlı kazanç ayarlamalı bulanık mantık PD ve PD olarak tasarlanan kontrolörün, daha temel, PD, bulanık mantık PD ve kazanç ayarlamalı bulanık PD kontrolörleriyle karşılaştırıldığında, hem kuvvet hem de tork geri beslemesinde verimlilikle kullanılabileceğini göstermişlerdir.

Ma ve arkadaşları [179], aktif titreşim yalıtımı için model belirsizlikli elektriksel olarak sürülen bir Stewart Platform manipülatörün kontrol problemini çözecek, kontrolör amaçlı radyal temel fonksiyonlu bir bozucu gözleyicisini önermişlerdir. Bu gözleyiciyi, harici bozucuları ve model hatalarını gözlemek için geliştirmişlerdir. Aktif titreşim yalıtım kontrolörünü tasarlamak için geri adım yöntemiyle iki radyal temel fonksiyonlu bozucu gözleyiciyi birleştirmişlerdir. Bu gözleyicilerin ağırlıkları için e1-düzeltim algoritmasına dayanan uyarlanır güncelleme kurallarını

tasarlamışlardır. Ayrıca, karşılaştırma amacıyla geleneksel hesaplamalı tork kontrolörünü simülasyonlarda kullanmışlardır. Geri adım yöntemini kullanarak önerilen kontrolör amaçlı gözleyicinin titreşim yalıtım performansını göstermek için hareketli platformun tüm eksenlerinde harici sinüzoidal ve vurma gibi titreşimleri uygulayarak iki hesapsal deneyi yürütmüşlerdir. Sonuç olarak, önerilen kontrolörün, hesaplamalı tork kontrolörü ile karşılaştırıldığında daha tatmin edici titreşim yalıtım performansına sahip olduğunu simülasyon sonuçları ile göstermişlerdir.

Yıldız ve arkadaşları [180], 3x3 Stewart Platform mekanizmasının küresel bir çalışma uzayını tarayarak elde ettikleri optimum PID katsayıları ile yine aynı noktaları ele alarak bulanık PID ve yapay sinir ağları ile yapılan kontrolleri karşılaştırmışlardır. Optimum kontrol parametreleri seçildikten sonra, belirlenen noktalar için Stewart Platform mekanizması üzerinde bu kontrolörleri uygulayarak simülasyon sonuçlarını elde etmişlerdir. Ayrıca kıyaslamak açısından, verilen noktalar için her bir kontrolörün ürettiği maksimum bacak kuvvetlerini hesaplamışlardır. Elde edilen sonuçlardaki veriler göz önünde tutularak, yapay sinir ağı kontrolörün, en iyi performans gösteren kontrolör olduğunu belirtmişlerdir. Ancak bacak kuvvetleri sistem boyutlarına göre gerçekçi olamayacak kadar büyük çıktığından, bu kontrolörün gerçek sisteme uygulanmasının mümkün olamayacağını ifade etmişlerdir.

Literatürde yapılan çalışmaların ışığı altında bu bölümde, Stewart Platform manipülatörün kontrol performansını, özellikle yörünge takip performansını geliştirmek için geleneksel PID, kesir dereceli PID (FOPID), bulanık PID (Fuzzy PID) ve bulanık kesir dereceli PID (Fuzzy FOPID) kontrol yapıları tasarlanacaktır. Stewart Platform manipülatörün hem katı gövde dinamik denklemlerini hem de bacalarda yer alan eyleyici dinamiğini içeren dinamik modeli kullanarak etkili bir arama tekniği olan parçacık sürü optimizasyon (PSO) algoritması ile bu kontrolörlere ait parametrelerin en iyi değerleri bulunacaktır.



Şekil 6.1. Stewart Platform'un yörünge takip kontrolü için kontrol blok diyagramı

Stewart Platform manipülâtörün eklem uzayı yörünge takip kontrolü için kontrol blok diyagramı, Şekil 6.1'de gösterilmektedir. Şekil incelendiğinde, hareketli platformun verilen yörüngesinden $(P_x, P_y, P_z, \alpha, \beta, \gamma)$ referans bacak uzunlukları (L_1, L_2, \dots, L_6) , ters kinematik bloğu ile hesaplanarak dinamik modelin gerçekleştirdiği bacak uzunlukları ile referans bacak uzunlukları karşılaştırılır. Karşılaştırma sonucu elde edilen sinyal kontrolörlere beslenerek üretilen kontrol sinyali ile sistem kontrol altında tutulmaya çalışılır. Bu arada, kontrolörün dayanıklılığını, kararlılığını, kalıcı ve dinamik yörünge takip doğruluğunu kapsayan kontrol performansının başarılabilmesi için bu kontrolörlere ait parametreler, belirlenen uygunluk fonksiyonuna göre çevrimiçi olarak PSO algoritması ile ayarlanır. Böylece en iyi yörünge takip performansını gösteren kontrolör tasarlanır.

Tasarlanan kontrolörlerin yörünge takip doğruluğunu, sağlamlığını ve kararlılığını içeren kontrol performansı, deneysel paralel robot olan Stewart Platform manipülâtörü üzerinde değerlendirilecektir. Bu kontrol yapılarının dinamik performanslarını değerlendirme açısından, hareketli platformun hem tek serbestlik

dereceli hem de tüm serbestlik dereceli hareketi için önerilen kontrolörlerin cevapları, yapılan deneyler boyunca gözlemlenecektir. Son olarak, gözlemlenen sonuçlar grafiksel ve tablolar halinde verilecektir.

6.2. PSO Tabanlı PID Kontrol Tasarımı

PID kontrolör, basitliği ve güvenilirliği bakımından endüstride en çok tutulan ve yaygın olarak kullanılan bir kontrol yapısıdır. PID kontrol katsayıları olarak adlandırılan K_p oransal, K_I integral ve K_D türev kazançları, PID kontrolörün performansını etkileyen en önemli unsurdur. Stewart Platform'unda yörünge takip kontrolünün istenilen şekilde yapılabilmesi, bu katsayıların uygun değerlerde atanmasıyla doğrudan alakalıdır.

Şekil 6.1'de görüldüğü gibi eklem uzayında kontrol tasarlanacağı için bu kontrol yönteminde hata sinyali $e(t)$, ters kinematik sonucu elde edilen bacak referans uzunluğu ile robotun gerçekleştirdiği bacak uzunluğu arasındaki farktır. Bu hata sinyalini en aza indirmek için PID kontrol parametrelerin (K_p , K_I ve K_D) uygun şekilde ayarlanması gerekir.

En iyi kontrol performansını sağlamak için PID konum kontrolörü kazançları, bu çalışmada önerilen PSO yaklaşımı kullanılarak ayarlanır. Stewart Platform manipülatöründe 6 bacağın konum kontrolü yapılacağından her bir bacak için PID kontrolör geliştirilir. Bacaklarda yer alan eyleyiciler aynı olduğundan, optimizasyonun başlangıç değerlerinde tüm PID kontrolörleri için aynı kazanç değerleri atanır.

Her bir PID kontrolör 3 parametreden (K_p , K_I ve K_D) oluştuğundan toplam 18 parametre optimize edilecektir. Hesapsal yük göz önünde bulundurularak PSO algoritmasında sürü boyutu 10 olarak seçilmiştir. Algoritmada, her bir parçacık 18 eleman içerdiğinden bir adımda toplam 180 parametre optimize edilir.

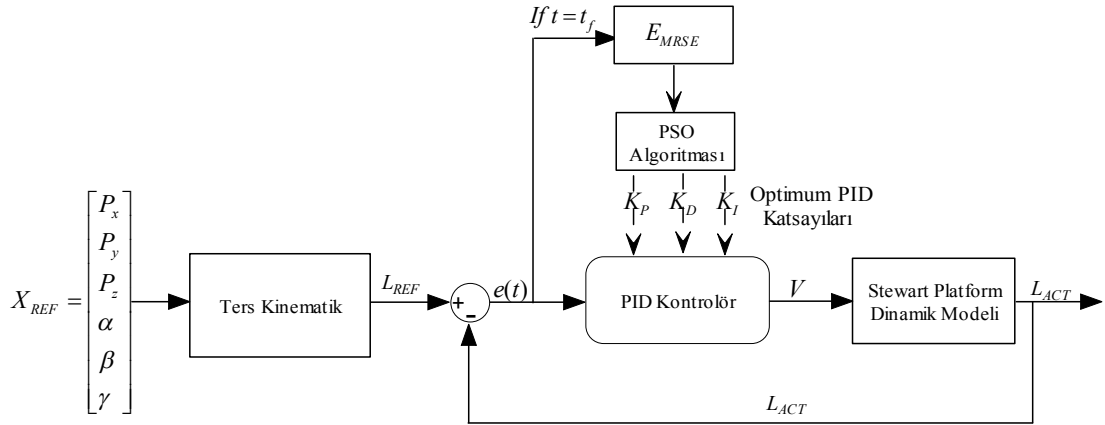
Denklem (6.1), PSO algoritmasında yer alan bir parçacığın optimize edilecek PID kontrol kazançlarını göstermektedir.

$$P_i = [K_{P_1}, K_{I_1}, K_{D_1}, K_{P_2}, K_{I_2}, K_{D_2}, \dots, K_{P_6}, K_{I_6}, K_{D_6}] \quad (6.1)$$

Yukarıdaki denklemde $K_{P_1}, K_{I_1}, K_{D_1}, K_{P_2}, K_{I_2}, K_{D_2}, \dots, K_{P_6}, K_{I_6}, K_{D_6}$, birinci PID kontrolörden altıncı PID kontrolöre kadar olan kazançları gösterir. Bu parametreler optimize edilirken, sistemin verilen zaman boyunca bir adım çalıştırılması sonucunda elde edilen hatalara göre her bir parçacık için maliyet fonksiyonu hesaplanır. Optimizasyon süreci boyunca kullanılacak maliyet fonksiyonu aşağıda verilmektedir.

$$E_{MRSE}(k) = \frac{1}{N} \sum_{i=1}^N \sqrt{e_1^2(i) + e_2^2(i) + e_3^2(i) + e_4^2(i) + e_5^2(i) + e_6^2(i)} \quad (6.2)$$

$e_1(i), e_2(i), \dots, e_6(i)$, eklem 1'den eklem 6'ya kadar bacaklarda oluşan i^{th} örnekteki hataları gösterir. N örnek sayısını, k ise iterasyon sayısını gösterir. MRSE ise Mean of Root of Squared Error'dur. Bu denklemde, hata değeri kullanılarak kontrolörün performansı, PSO için maliyet değerine çevrilir.



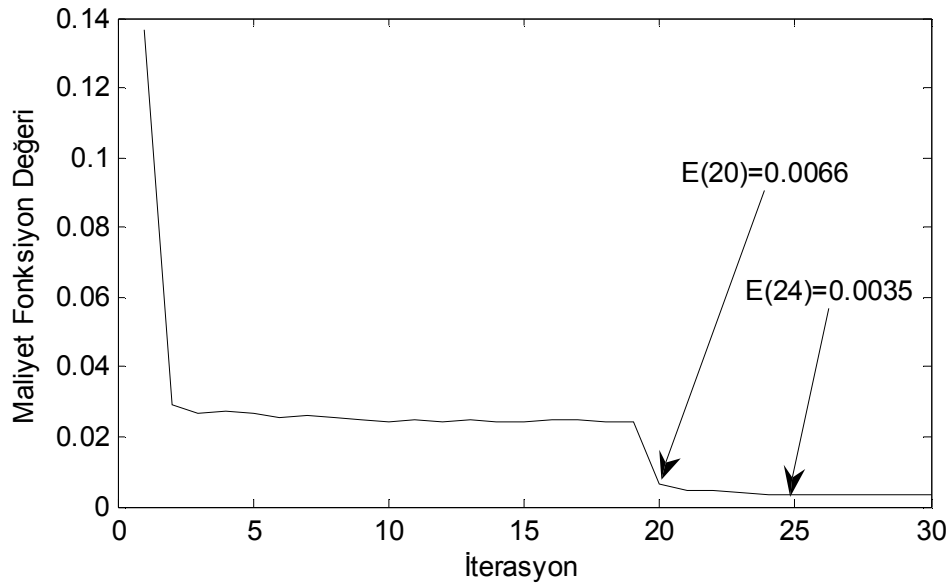
Şekil 6.2. PSO ile PID kontrolörün optimizasyonu

PSO algoritması ile PID kontrolörün optimize edilme sürecinin blok yapısı, Şekil 6.2,'de gösterilmektedir. Eklemlerde bulunan PID kontrolörlerin tüm parametreleri her simülasyon zamanı bitiminde (t_f) optimize edilen parçacıklara göre güncelleştirilir. Bu güncelleme sırasında en küçük maliyet fonksiyon değerine sahip parçacık en iyi parçacıktır. Bu sebeple PSO işletilirken bu maliyet fonksiyonu, sürekli azalır bir seyir izlemesi beklenir.

Simülasyon ortamında gerçekleştirilen Stewart Platform'un dinamik ve ters kinematik modeli kullanılarak PID kontrol parametreleri, verilen hareketli platform referans yörüngesine göre ayarlanır. PID kontrolörlerin parametrelerini optimize etmek için üst platformun tüm eksenlerde hareketini sağlayan yörünge denklemi aşağıda verildiği gibi seçilmiştir.

$$\begin{aligned}
 x(t) &= 7 \sin(2\pi t) & \alpha(t) &= 5 \cos(2\pi t) \\
 y(t) &= 7 \cos(2\pi t) & \beta(t) &= 5 \sin(2\pi t) \\
 z(t) &= 0.25 + 3 \sin(2\pi t) & \gamma(t) &= 3 \cos(2\pi t)
 \end{aligned} \tag{6.3}$$

Şekil 6.3'te, verilen referans yörüngesi için bacakların yörünge takip kontrolünü sağlayan PID kazançlarının optimizasyon süreci boyunca en iyi parçacığın elde ettiği maliyet fonksiyon değerleri gösterilmektedir.



Şekil 6.3. Optimizasyon süreci boyunca PSO tabanlı PID kontrolörün yakınsama eğilimi

Şekle 6.3 incelendiğinde, yaklaşık ilk 4 iterasyon boyunca maliyet fonksiyonu değerinde hızlı bir azalma ve sonrasında 18. iterasyona kadar çok değişmeyen değerler görülmektedir. Yaklaşık 20. iterasyonda maliyet fonksiyonunun 0,0066 değerine düşüp sonraki iterasyonlarda 0,0035 değerine yakınsayarak kontrolör için en iyi kazanç değerleri elde edilmiştir.

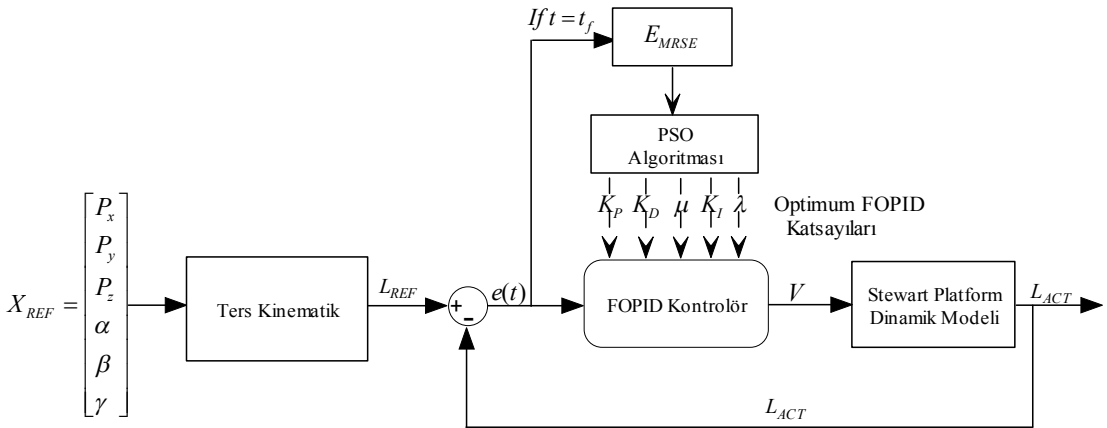
6.3. PSO Tabanlı FOPID Kontrol Tasarımı

PID kontrolörlerin kalitesini ve sağlamlığını arttırmanın yollarından biri de, tam sayı olmayan türev ve integral kısımlara sahip kesir dereceli kontrolörler kullanmaktır. Türev ve integratörün derecelerinin kesirli olması, kontrol sisteminin frekans cevabını doğrudan ve sürekli olarak ayarlanmasını sağlar.

Verilen bir sistem için FOPID ($PI^\lambda D^\mu$) kontrolörün tasarımında K_p, K_I, K_D ve ondalık dereceli λ, μ toplam 5 parametre en iyi olarak ayarlanır. Buradaki amaç, Şekil 6.1’de görüldüğü gibi eklem uzayında konum kontrolünü sağlayacak FOPID kontrolörün parametrelerini ayarlamaktır. Stewart Platform sisteminin istenilen cevabı ve kararlılığı göstermesi için FOPID kontrolörünün $K_p, K_d, K_i, \lambda, \mu$ parametreleri önerilen PSO algoritması ile ayarlanır. Her bir kontrolör 5 parametreye sahip olduğundan bu çalışmada toplam 30 parametre optimize edilir. Bundan dolayı PSO algoritması, 30 boyutlu uzayda kontrol parametrelerinin tümünü arar. Algoritmada yer alan sürü içerisindeki bir parçacığın düzeni, aşağıdaki denklemde ifade edildiği gibidir.

$$P_i = [K_{p_1}, K_{d_1}, \mu_1, K_{i_1}, \lambda_1, K_{p_2}, K_{d_2}, \mu_2, K_{i_2}, \lambda_2, \dots, K_{p_6}, K_{d_6}, \mu_6, K_{i_6}, \lambda_6] \quad (6.4)$$

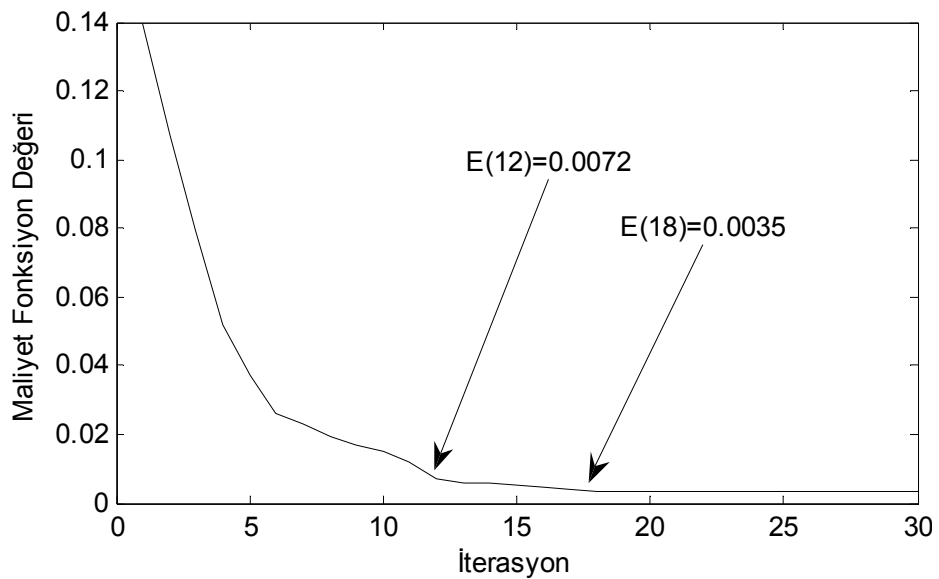
Yukarıdaki denklem, $K_{p_1}, \dots, \lambda_1$ ‘den $K_{p_6}, \dots, \lambda_6$ ‘ya kadar sırasıyla birinci kontrolörden altıncı kontrolöre kadar olan kontrolörlerin elemanlarını göstermektedir.



Şekil 6.4. PSO ile FOPID kontrolörün optimizasyonu

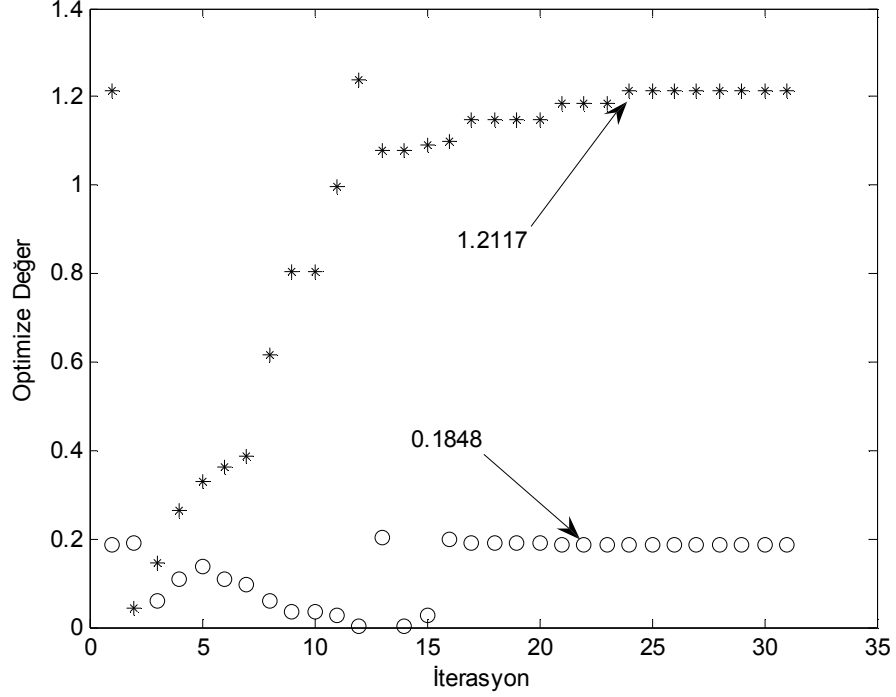
Şekil 6.4, FOPID kontrolörün PSO ile optimize edilme sürecinin blok yapısını göstermektedir. FOPID kontrolörlerin tüm parametreleri her simülasyon zamanı bitiminde (t_f) güncelleştirilir. PID kontrolörü tasarımı olduğu gibi bu çalışmada da aşırı hesapsal yükü göz önünde bulundurarak sürü büyüklüğü 10 parçacık olarak ayarlanır. Her parçacık P_i , 30 elemana sahip olduğundan sürünün boyutu 10x30 şeklindedir. Kontrol parametrelerini içeren her bir parçacığa ait elemanların değerleri, makul aralıklarda tanımlanır. Burada, optimizasyon zamanını azaltmak için ve de pratik gereksinimlerden esinlenmeyle, FOPID kontrolörün parametreleri alt sınır olarak sıfır üst sınır ise $K_{p_{max}} = 50$, $K_{i_{max}} = K_{d_{max}} = 10$ ve $\lambda_{max} = \mu_{max} = 2$ olarak seçilmiştir. Birinci iterasyon için bu üst sınır değerlerine bağlı olarak parçacıkların başlangıç değerleri rastgele üretilmiştir.

Denklem (6.3)'te verilen hareketli platformun yörünge denklemi kullanılarak Şekil 6.4'teki PSO bloğunda, Denklem (6.2)'ye göre her bir parçacık için maliyet fonksiyon değeri hesaplanır. Buna göre, algorithmada yer alan pbest ve gbest parametreleri, her simülasyon zamanı sonunda (t_f) elde edilir. Elde edilen bu parametreler kullanılarak her parçacığın hızı hesaplanıp, konumu güncellenir. Burada, algoritmanın en uyguna yakınsama eğilimini elde etmek için her bir adımdaki en iyi parçacığın göstermiş olduğu maliyet değerleri kaydedilir. Sonuç olarak, en iyi parçacığa göre FOPID kontrolörün parametreleri ayarlanır.



Şekil 6.5. Optimizasyon süreci boyunca PSO tabanlı FOPID kontrolörün yakınsama eğilimi

Verilen yörünge denklemine göre optimizasyon sonunda elde edilen maliyet fonksiyonunun değişimi ve algoritma adımı boyunca en iyi parçacığın içerdiği λ ve μ parametrelerin değişim değerleri, sırasıyla Şekil 6.5 ve 6.6'da gösterilmektedir.



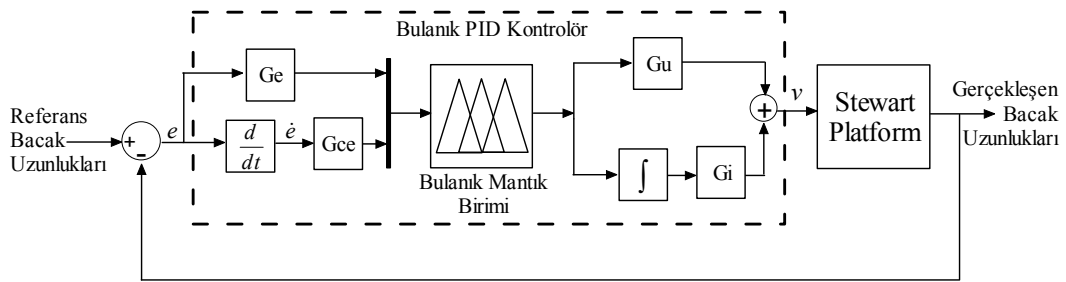
Şekil 6.6. Optimizasyon adımı sırasında μ (*) ve λ (o) için elde edilen optimize değerler

Şekil 6.5 incelendiğinde maliyet değeri, yaklaşık 12 adım boyunca hızlı bir düşüş göstererek 12 iterasyon sonunda 0,0072 değerine ulaşmış ve yaklaşık 6 adım sonra en iyi değere (0,0035) ulaşmıştır. Diğer taraftan PSO'nun yakınsama karakteristiği pürüzsüz ve büyük değişimler göstermemiştir. Şekil 6.6 gözlemlendiğinde, maliyet fonksiyonu en iyi değerine yaklaşıldığında türev (μ) ve integratörün (λ) dereceleri, belli bir değerde kaldığı ya da çok az değişimler gösterdiği görülmektedir.

6.4. PSO Tabanlı Bulanık PID Kontrol Tasarımı

Kontrol edilen sistemlerin doğrusal olmayan sistemler olması durumunda, PID kontrolörlerin performansları sınırlı ve yetersiz kalmaktadır. Bunun üstesinden gelebilmek amacıyla dilsel terimlere dayalı doğrusal olmayan kontrol yapısı ile doğrusal olmayan sistemlerin kontrolünde başarı sağlayan kontrol yapılarından birisi bulanık kontrolörlerin tasarlanmasıdır.

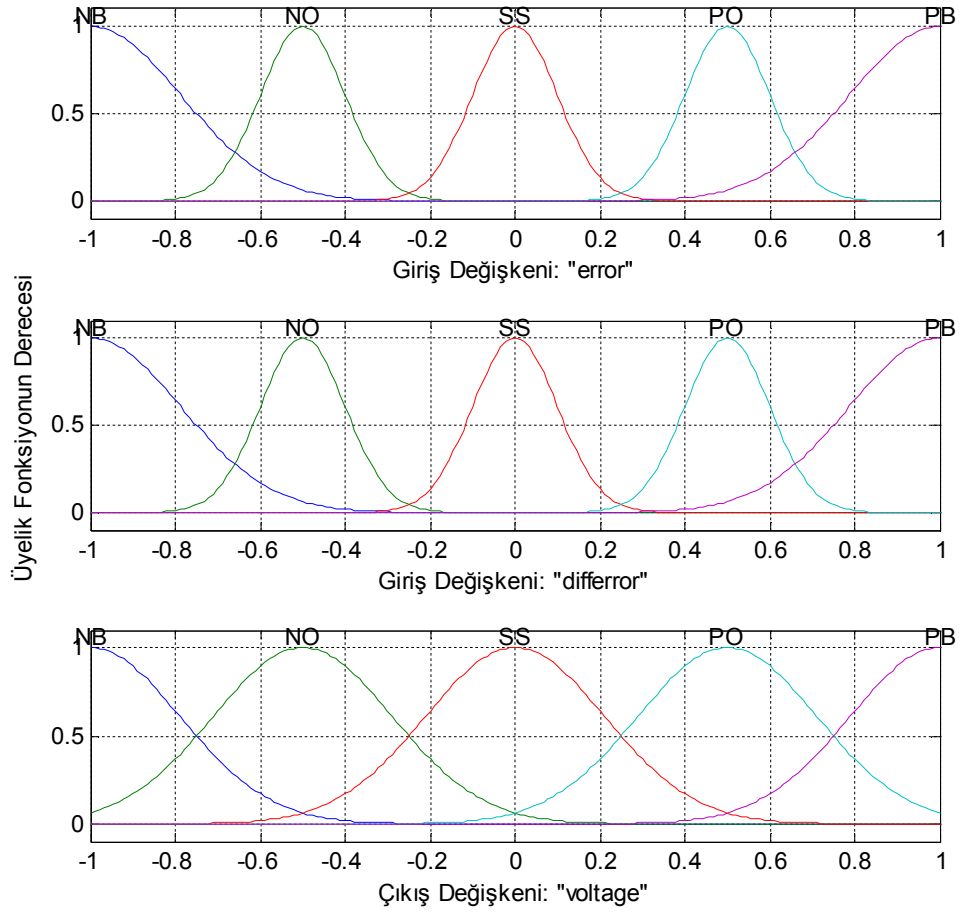
Burada ön görülen kontrol yapısı, Şekil 6.7’de gösterildiği gibi iki girişli ve çıkışına integral ve toplama elemanın konulmasıyla tek kural tabanlı olarak tasarlanan bir bulanık PID kontrolördür. Şekil 6.7’ye bakıldığında bulanık PID kontrolör yapısı, hata (e), hatanın türevi (\dot{e}), kontrolöre gelen hata değerinin kazanç katsayısı, hata değişim değerinin kazanç katsayısı, bulanık birim, çıkış katsayıları ve bunların toplamını oluşturan kontrol sinyalinden oluşmaktadır. Buradaki amaç, Şekil 6.1’de gösterildiği gibi eklem uzayında bu bulanık PID kontrol yapısını kullanarak 6 adet doğrusal tahrik elamanın konum kontrolünü başarmaktır.



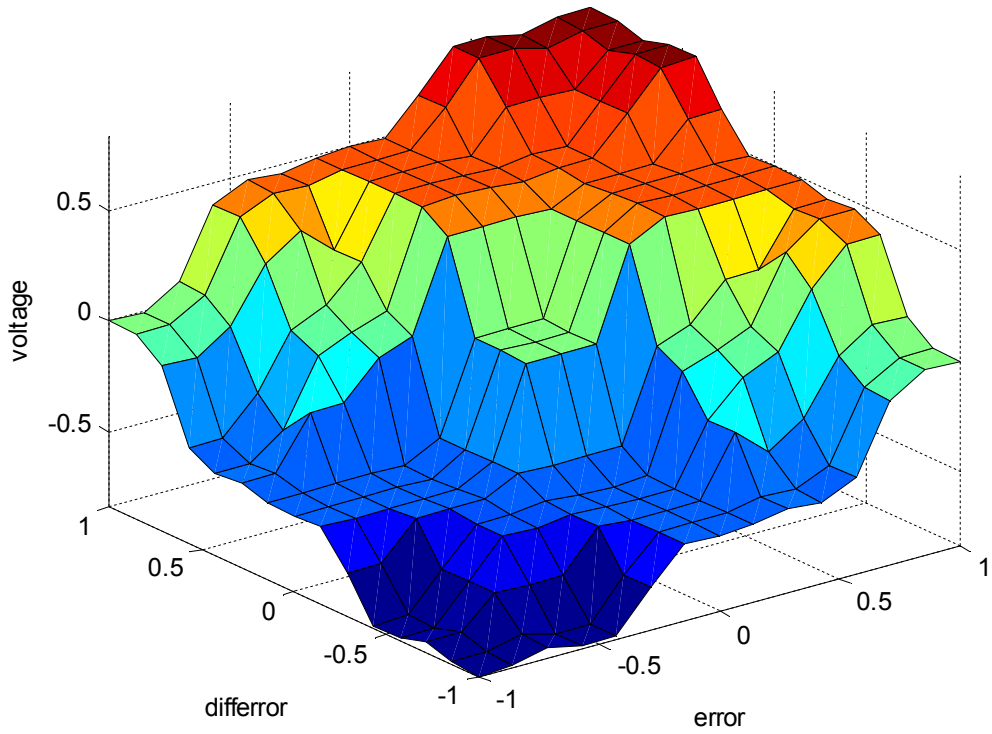
Şekil 6.7. Bulanık PID ile kontrol edilen sistemin blok diyagramı

Her bir bacağın konum kontrolü için tasarlanacak bulanık PID kontrolörler, iki girişli tek çıkışlı, cebirsel çarpım T-normlu, Gauss üyelik fonksiyonlu ve ağırlıklı ortalama durulandırılmalı bir Mamdani bulanık modelidir. Mamdani bulanık modelinde, verilen giriş değerine karşı üyelik değeri küçük olan üyelik fonksiyonun değeri alınır ve çıkış fonksiyonu kesilerek altta kalan alan alınır. Diğer kurallar için de aynı işlem yapılır. Seçilen ağırlıklı ortalama durulaştırma yöntemi ile bulanık olan çıkış değeri keskin bir değere dönüştürülür. Bu değer bulanık PID’nin çıkış katsayıları ile çarpılır. Böylece, çarpımlar toplanarak bacaklar için kontrol sinyali üretilir.

Her bir kontrolör, hata değeri için bacak uzunluğu ve hatanın değişimi için de hatanın türevi şeklinde ikişer girişe, çıkış olarak da platforma ait her bir bacağına uygulanacak gerilim değeri için birer çıkışa sahiptir. Tüm bacakların kontrolü için aynı tip kontrolör yapısı tasarlanmıştır. Tasarlanan bu kontrolörlere ait giriş değişkenleri (error ve differror) ve çıkış değişkeni (voltage) olan, “Gauss” üyelik fonksiyonları, Şekil 6.8’de gösterildiği gibi $[-1,1]$ aralığında ölçeklendirilmiş olarak tanımlanmıştır. Kontrolörlere ait başlangıç kontrol yüzeyi ise Şekil 6.9’da gösterilmiştir.



Şekil 6.8. Bulanık kontrolör girişlerin ve çıkışın, başlangıç Gauss üyelik fonksiyonları ile bulanıklaştırılması



Şekil 6.9. Bulanık kontrolörün optimizasyon öncesi başlangıç kontrol yüzeyi

Ayrıca, kural tabanını oluşturan kural tablosu Tablo 6.1’de verilmiştir. Burada kullanılan dilsel değişkenler; NB, negatif büyük; NO, negatif orta; SS, sıfır; PO, pozitif orta; PB, pozitif büyük olarak kısaltılmıştır. Bu dilsel terimler giriş ve çıkış uzayının bulanık kısmını oluşturur. Tablo 6.1’de verildiği gibi toplam 25 kural tanımlanmıştır.

Tablo 6.1. Bulanık mantık kural atama tablosu

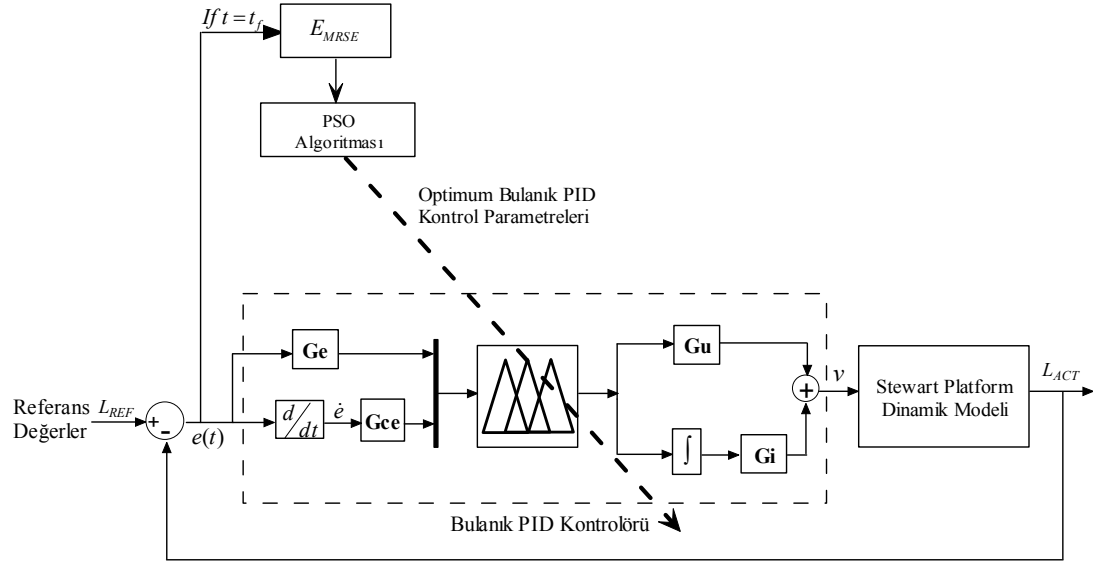
		differror				
		NB	NO	SS	PO	PB
error	NB	NB	NB	NO	NO	SS
	NO	NB	NO	NO	SS	PO
	SS	NO	NO	SS	PO	PO
	PO	NO	SS	PO	PO	PB
	PB	SS	PO	PO	PB	PB

Yukarıdaki kural tablosu EĞER - O HALDE (IF-THEN) yapısı kullanılarak oluşturulmuştur. Kurallar incelendiğinde kontrol işareti, hatanın büyümesiyle büyümektedir. İkinci giriş olan hatanın türevi ise, hatanın değişimi hakkında bilgi vererek kontrol işaretinin duruma göre, arttırılmasını veya azaltılmasını sağlamaktadır.

Stewart Platform manipülatörün eklem uzayında istenilen cevabı, kararlılığı ve bozululara karşı ideal kontrol performansını sağlamak için önerilen bu bulanık PID kontrol yapısının parametreleri ayarlanır. Bu sebeple, Denklem (6.2)’de verilen maliyet fonksiyonuna göre bulanık PID kontrolöründe yer alan Ge, Gce, Gi, Gu ve 15 Gauss üyelik fonksiyonlarına ait parametreler önerilen PSO algoritması ile optimize edilir. Her bir kontrolör 34 parametreye sahip olduğundan toplam 204 parametre en iyi olarak ayarlanır. Bundan dolayı PSO algoritması, 204 boyutlu uzayda kontrol parametrelerinin tümünü arar. Bu kontrol parametrelerini içeren sürüdeki bir parçacığın düzeni, aşağıda verildiği gibi ifade edilebilir.

$$\begin{aligned}
 P_i = & [Ge_1, Gce_1, Gi_1, Gu_1, \sigma_{11}, c_{11}, \dots, \sigma_{15}, c_{15}, \sigma_{21}, c_{21}, \dots, \sigma_{25}, c_{25}, \sigma_{31}, c_{31}, \dots, \sigma_{35}, c_{35}, \\
 & Ge_2, Gce_2, Gi_2, Gu_2, \sigma_{41}, c_{41}, \dots, \sigma_{45}, c_{45}, \sigma_{51}, c_{51}, \dots, \sigma_{55}, c_{55}, \sigma_{61}, c_{61}, \dots, \sigma_{65}, c_{65}, \\
 & \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \dots, \\
 & Ge_6, Gce_6, Gi_6, Gu_6, \sigma_{161}, c_{161}, \dots, \sigma_{165}, c_{165}, \sigma_{171}, c_{171}, \dots, \sigma_{175}, c_{175}, \sigma_{181}, c_{181}, \dots, \sigma_{185}, c_{185}]
 \end{aligned}
 \tag{6.5}$$

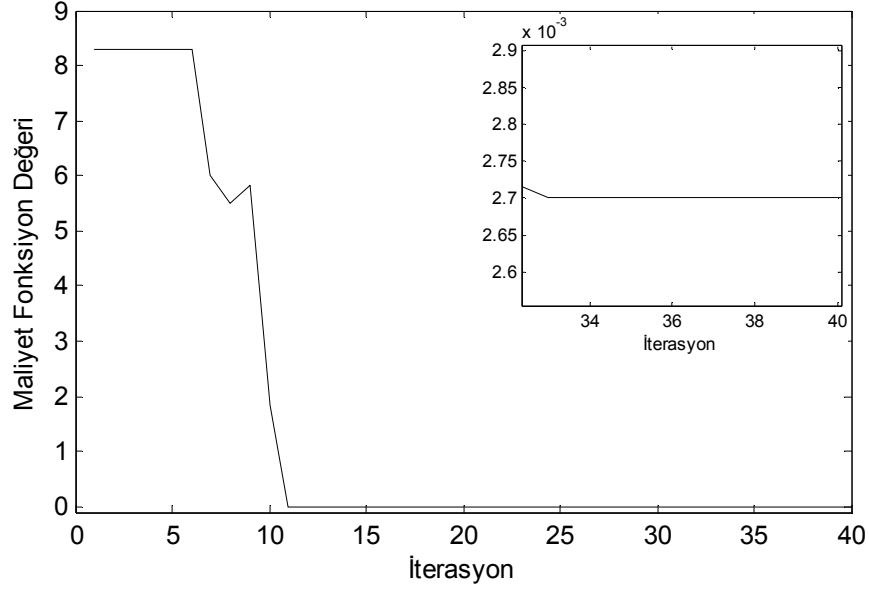
Denklem (6.5)'teki tüm satırlarda, bulanık PID kontrolörüne ait hata değerinin kazanç katsayısı, hata değişim değerinin kazanç katsayısı, çıkış katsayıları ve kontrolörün giriş ve çıkış değişkenlerinin üyelik fonksiyonlarına ait parametreler yer alır. σ_{ij} ve c_{ij} parametreleri Gauss üyelik fonksiyonunun genişliğini ve merkezini göstermektedir. Örneğin; c_{185} , 6. bacağı kontrol eden bulanık PID'nin çıkış değişkenine ait 5. üyelik fonksiyonunun merkezini gösterir. Simülasyon çalışmasında kullanılan PSO algoritması ile bulanık PID kontrolörün parametrelerini optimize etme süreci, Şekil 6.10'da gösterilmektedir. Önceki çalışmalarda olduğu gibi aşırı hesapsal yükü göz önünde bulundurarak sürü büyüklüğü 10 parçacık olarak ayarlanır. Her parçacık P_i , 204 elemana sahip olduğundan sürünün boyutu 10x204 şeklindedir.



Şekil 6.10. PSO ile bulanık PID kontrolörün optimizasyonu

Optimizasyon için verilen üst platformun yörüngesine göre Şekil 6.10'daki sistem, parametre ayarlama işleminin simülasyon modeli olarak çalıştırılır. Sistemin, verilen zaman (t_f) boyunca bir adım çalıştırılması sonucunda, elde edilen hatalara göre Denklem (6.2) kullanılarak her bir parçacık için maliyet fonksiyonu değeri hesaplanır. Daha sonra, bu maliyet değerine göre her parçacık için parçacığın hızı hesaplanarak parçacığın konumu güncellenir. Böylece, bulanık PID kontrolörlerin tüm parametreleri her simülasyon zamanı (t_f) bitiminde güncellenir. İterasyon sonucunda en küçük maliyet değerini sağlayan parçacık en iyi parçacık olarak kontrol parametrelerini içerir. PSO işletilirken maliyet değerlerinin sürekli azalır bir

seyir izlemesi beklenir. Çalıştırılan iterasyon sayısına göre optimizasyon süresi boyunca her bir adımda elde edilen en iyi parçacığın maliyet değerlerin yakınsaması Şekil 6.11’de gösterilmektedir. Şekle bakıldığında yaklaşık olarak 32 iterasyon sonucunda maliyet değerinin 0,0027’ye düşüp belirlenen iterasyon sayısına kadar aynı değerde kaldığı görülmektedir.

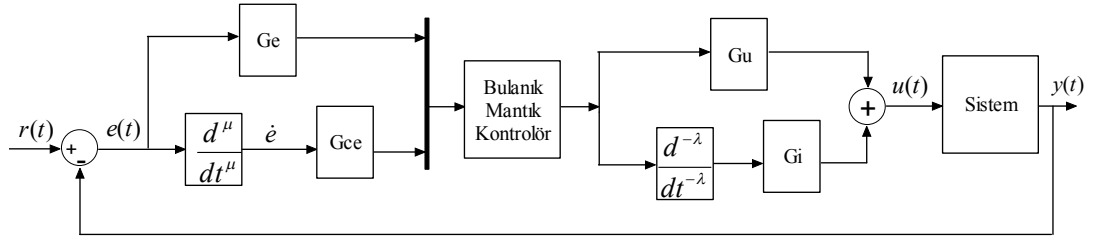


Şekil 6.11. Bulanık PID kontrolörün parametre optimizasyonu boyunca maliyet fonksiyonunun yakınsama karakteristiği

6.5. PSO Tabanlı Bulanık FOPID Kontrol Tasarımı

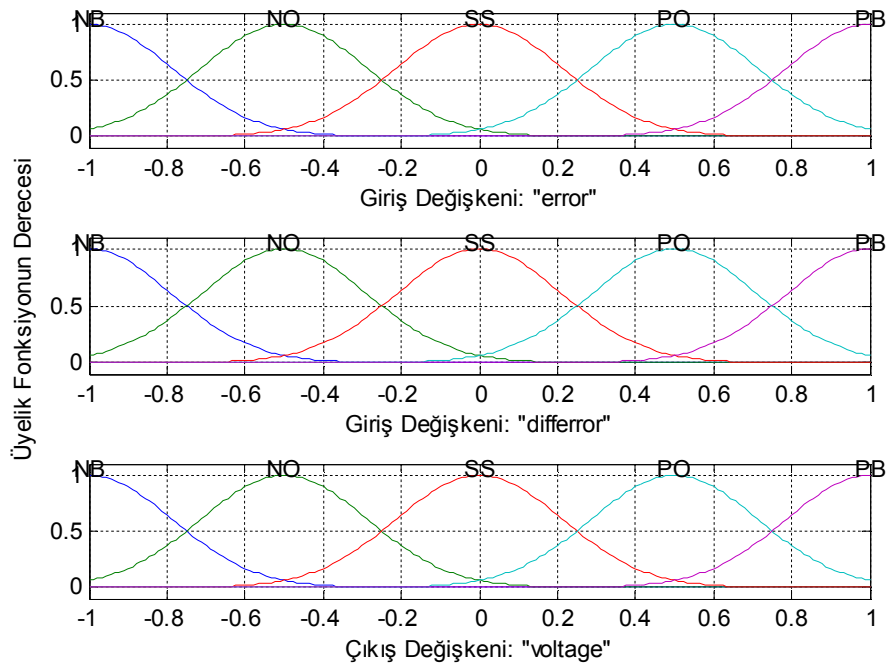
Burada önerilen bulanık PID kontrolör, giriş ve çıkış ölçekleme faktörlü bir bulanık PD tipi kontrolörün çıkışına integral ve toplam elemanı konulan tek kural tabanlı bir yapıdır. Bu kontrol yapısında, hata (e) ve hatanın türevi (\dot{e}) girdiler olmak üzere bir işaret üretilir. Bu işaret, bulanık mantık kontrolörün çıkışı olan G_u ve aynı zamanda integrali alınarak G_i ile çarpılır. Daha sonra, kontrol sinyalini üretmek için bu çıkışlar toplanır. Bu kontrol yapısındaki tasarımı genişletmek için, bulanık mantık kontrolörün girişindeki türevin ve çıkıştaki integratörün tam sayı dereceleri, kesirli sayılar ile (μ ve λ) ile yer değiştirilir. Böylece, Şekil 6.12’de gösterildiği gibi ek tasarım gereksimlerini karşılayacak bir bulanık FOPID kontrol yapısı elde edilir. Bulanık kontrolörler için çıkış ölçekleme katsayısının değiştirilmesi, üyelik fonksiyonların şeklinin değiştirilmesinden kontrolör performansı üzerinde daha etkilidir. Bu nedenle, kontrolörün tüm performansını etkilemede, ayarlama

parametrelerin hepsi aynı derecede etkili değildir. Bu kontrol yapısındaki amaç, kontrol sistemin kapalı döngü performansını arttırmak için kural tabanını ve üyelik fonksiyonlarını sabit tutarak kesir dereceli değişkenlerin optimizasyona katılmasıyla bunların etkisini gözlemektir. Üyelik fonksiyonları ve diğer bulanık çıkartım değişkenleri yerine, bu iki ek parametreyi uygun bir şekilde ayarlayarak kontrol sisteminin performansını etkileyebildiği için uygulanabilirlik ve pratikte kullanım kolaylığı açısından bu yaklaşım daha iyi olabilir.



Şekil 6.12. Kesir dereceli bulanık PID kontrol yapısı

Burada, bulanık modelin kural tabanını sabit tutarak önerilen bulanık FOPID kontrol yapısındaki giriş ve çıkış ölçekleme katsayılarını, üyelik fonksiyonların parametrelerini ve de integratör ve türevin kesir dereceli değişkenlerini optimizasyona katarak Stewart Platform manipülatörün eklem uzayı yörünge takip performansı gözlemlenecektir.



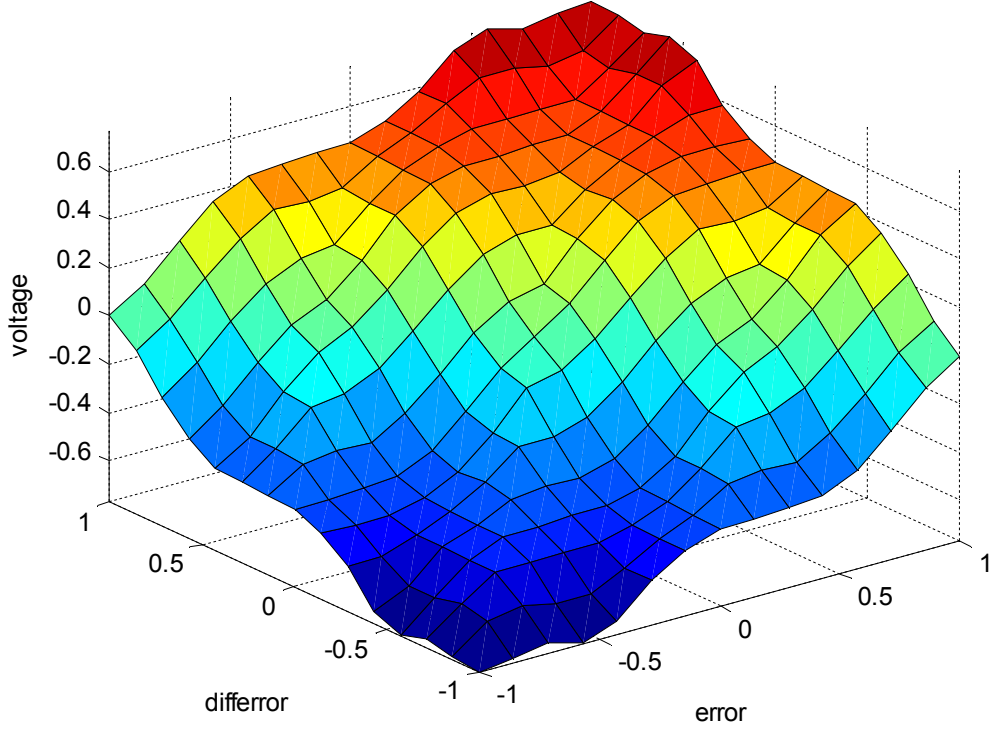
Şekil 6.13. Bulanık kontrolörün hata, kesir dereceli hata değişimi ve çıkış değişkenleri için başlangıç Gauss üyelik fonksiyonları

Stewart Platform manipülatörün eklem uzayı konum kontrolünde tasarlanacak bulanık FOPID kontrolörler için, bulanık PID kontrolörlerin tasarımında kullanılan aynı özellikli Mamdani bulanık modeli seçilmiştir. Şekil 6.13'te gösterilen giriş değişkenlerine (error ve differror) ve çıkış değişkenine (voltage) ait üyelik fonksiyonları ve de Tablo 6.2'de verilen kural tabanı, bir önceki bölümde tasarlanan bulanık PID kontrol yapısına benzer seçilmiştir. Tablodan anlaşılacağı üzere yalnızca 5 dilsel değişken kullanıldığından 25 kural tanımlanmıştır. Bu dilsel değişkenler, kontrol hareketinin kalitesini belirlediğinden ve kontrol performansı üzerinde oldukça etkili olduğundan daha iyi kontrol hassasiyeti için fazla sayıda dilsel değişken kullanılabilir. Ancak bu durumda, dilsel değişkenlerin sayısını gösteren n arttığında kural tabanı da n^2 'nin derecesinde artacaktır. Bu da, kontrolün kalitesi ile hesaplama zamanı arasında bir uzlaşmaya gereksinim duylamasına ve de gerçek zamanlı uygulamalarda zorluklara neden olacaktır.

Tablo 6.2. Hata, hatanın kesir dereceli türevi ve bulanık çıkış için kural tabanı

		$\frac{d^n e}{dt^n}$	differror				
			NB	NO	SS	PO	PB
error	NB	NB	NB	NO	NO	SS	
	NO	NB	NO	NO	SS	PO	
	SS	NO	NO	SS	PO	PO	
	PO	NO	SS	PO	PO	PB	
	PB	SS	PO	PO	PB	PB	

Şekil 6.14, bulanık FOPID kontrolörü için oluşturulan kurallar sonucu giriş ve çıkışa ait doğrusal olmayan başlangıç kontrol yüzeyini göstermektedir. Hesapsal verimlilik, bellek kullanımı ve performans analizi gereksinimleri açısından bakıldığında, üyelik fonksiyonların tek tipi tercih edilir. Bu sebeple, "Gauss" üyelik fonksiyonun parametrik fonksiyonel ifadesi diğer üyelik fonksiyonları arasında daha ekonomik olduğundan gerçek zamanlı uygulamalar için kontrolör tasarımında daha çok benimsenir. Dolayısıyla, burada önerilen bulanık FOPID kontrol yapısında, hata ve hatanın kesir dereceli türevi için iki boyutlu doğrusal kural tabanlı ve "Gauss" üyelik fonksiyonlu Mamdani çıkartımlı bulanık bir model kullanılmıştır.



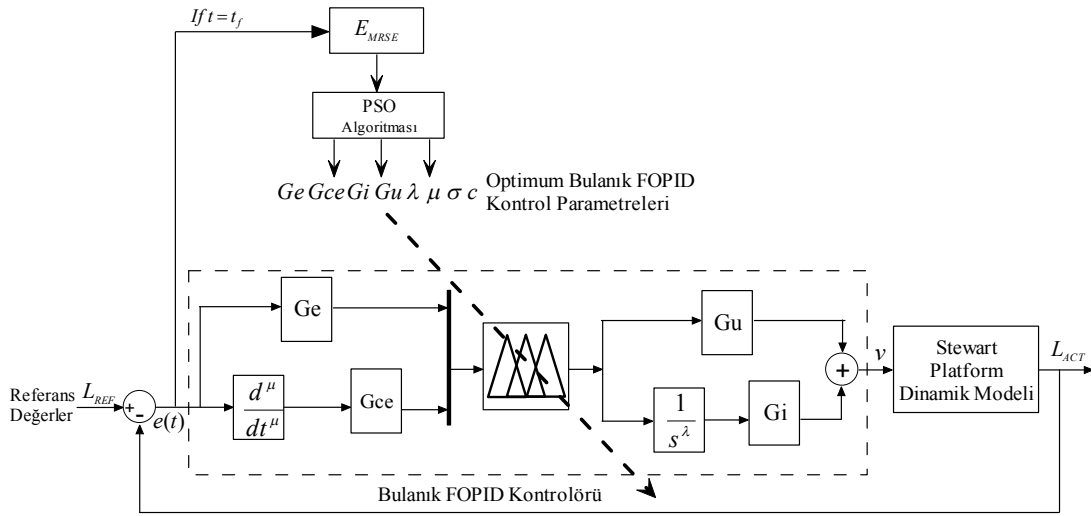
Şekil 6.14. Bulanık FOPID kontrolörü için giriş ve çıkışa ait optimizasyon öncesi başlangıç kontrol yüzeyi

Stewart Platform manipülatörün eklem uzayında istenilen cevabı, kararlılığı ve bozuculara karşı ideal kontrol performansını sağlamak için önerilen bu bulanık FOPID kontrol yapısının parametreleri ayarlanır. Bu sebeple, Denklem (6.2)'de verilen maliyet fonksiyonuna göre bulanık FOPID kontrolöründe yer alan $Ge, Gce, Gi, Gu, \lambda, \mu$ ve 15 Gauss üyeli fonksiyonlarına ait parametreler önerilen PSO algoritması ile optimize edilir. Her bir kontrolör 36 parametreye sahip olduğundan toplam 216 parametre en iyi olarak ayarlanır. Bundan dolayı PSO algoritması, 216 boyutlu uzayda kontrol parametrelerinin tümünü arar. Bu kontrol parametrelerini içeren sürüdeki bir parçacığın düzeni, aşağıda verildiği gibi ifade edilebilir.

$$\begin{aligned}
 P_i = & \left[Ge_1, Gce_1, Gi_1, Gu_1, \lambda_1, \mu_1, \sigma_{11}, c_{11}, \dots, \sigma_{15}, c_{15}, \sigma_{21}, c_{21}, \dots, \sigma_{25}, c_{25}, \sigma_{31}, c_{31}, \dots, \sigma_{35}, c_{35} \right. \\
 & Ge_2, Gce_2, Gi_2, Gu_2, \lambda_2, \mu_2, \sigma_{41}, c_{41}, \dots, \sigma_{45}, c_{45}, \sigma_{51}, c_{51}, \dots, \sigma_{55}, c_{55}, \sigma_{61}, c_{61}, \dots, \sigma_{65}, c_{65} \\
 & \dots \\
 & \dots \\
 & \dots \\
 & Ge_6, Gce_6, Gi_6, Gu_6, \lambda_6, \mu_6, \sigma_{161}, c_{161}, \dots, \sigma_{165}, c_{165}, \sigma_{171}, c_{171}, \dots, \sigma_{175}, c_{175} \\
 & \left. \sigma_{181}, c_{181}, \dots, \sigma_{185}, c_{185} \right]
 \end{aligned} \tag{6.6}$$

Yukarıdaki denklemin tüm satırlarında, önerilen bulanık FOPID kontrol yapısına ait hata değerinin kazanç katsayısı, hata değişim değerinin kazanç katsayısı, çıkış katsayıları, türev ve integratörün kesir dereceleri ve kontrolörün giriş ve çıkış değişkenlerinin üyelik fonksiyonlarına ait parametreler yer alır. σ_{ij} ve c_{ij} parametreleri Gauss üyelik fonksiyonunun genişliğini ve merkezini göstermektedir. Örneğin; c_{65} , 2. bacağı kontrol eden bulanık FOPID'nin çıkış değişkenine ait 5. üyelik fonksiyonunun merkezini gösterir.

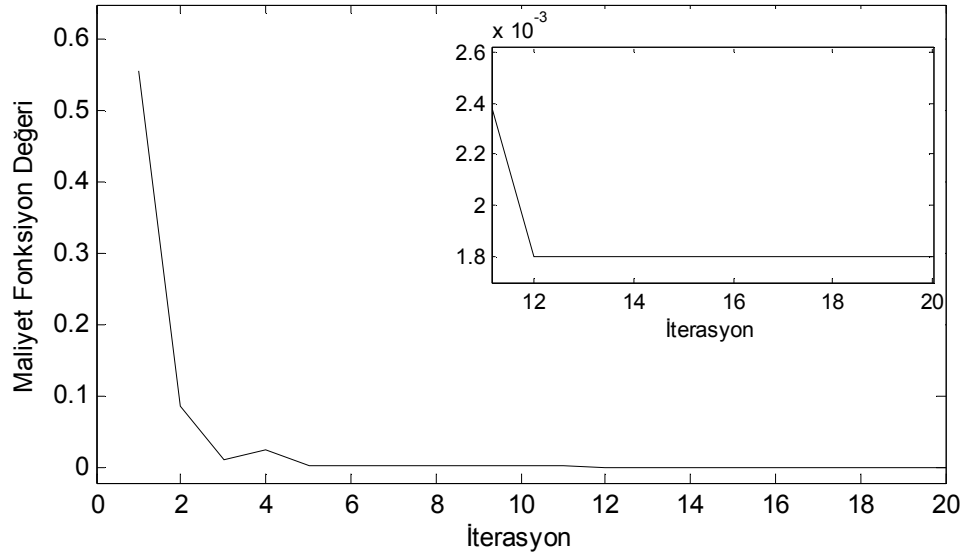
Simülasyon çalışmasında kullanılan PSO algoritması ile bulanık FOPID kontrolörün parametrelerini optimize etme süreci, Şekil 6.15'de gösterilmektedir. Önceki çalışmalarda olduğu gibi aşırı hesapsal yükü göz önünde bulundurarak kullanılan bu sezgisel algoritmada sürü büyüklüğü 10 parçacık olarak ayarlanır. Her parçacık P_i , 216 elemana sahip olduğundan sürünün boyutu 10×216 şeklindedir.



Şekil 6.15. Önerilen PSO ayarlı bulanık FOPID kontrol sistemi

Optimizasyon için verilen üst platformun yörüngesine göre Şekil 6.15'teki sistem, bulanık FOPID kontrolörünün en uygun parametrelerini bulma işlemi için simülasyon modeli olarak çalıştırılır. Sistemin, verilen zaman (t_j) boyunca bir adım çalıştırılması sonucunda, elde edilen hatalara göre Denklem (6.2) kullanılarak her bir parçacık için maliyet fonksiyon değeri hesaplanır. Daha sonra, bu maliyet değerine göre her parçacık için parçacığın hızı hesaplanarak parçacığın konumu güncellenir. Böylece, bulanık FOPID kontrolörlerin tüm parametreleri her simülasyon zamanı (t_j)

bitiminde güncellenir. İterasyon sonucunda en küçük maliyet değerini sağlayan parçacık en iyi parçacık olarak kontrol parametrelerini içerir. PSO işletilirken bulanık FOPID parametrelerin ayarlanabilmesi için bu maliyet fonksiyonun minimize edilmesi beklenir. PSO koşturulurken her bir adımda elde edilen en iyi parçacığın maliyet fonksiyon değerlerinin iterasyonla değişimi, Şekil 6.16’da gösterilmektedir. Şekle bakıldığında yaklaşık olarak 12 iterasyon sonucunda maliyet değerinin 0,0018’e düşüp 20. İterasyona kadar aynı değerde kaldığı görülmektedir.



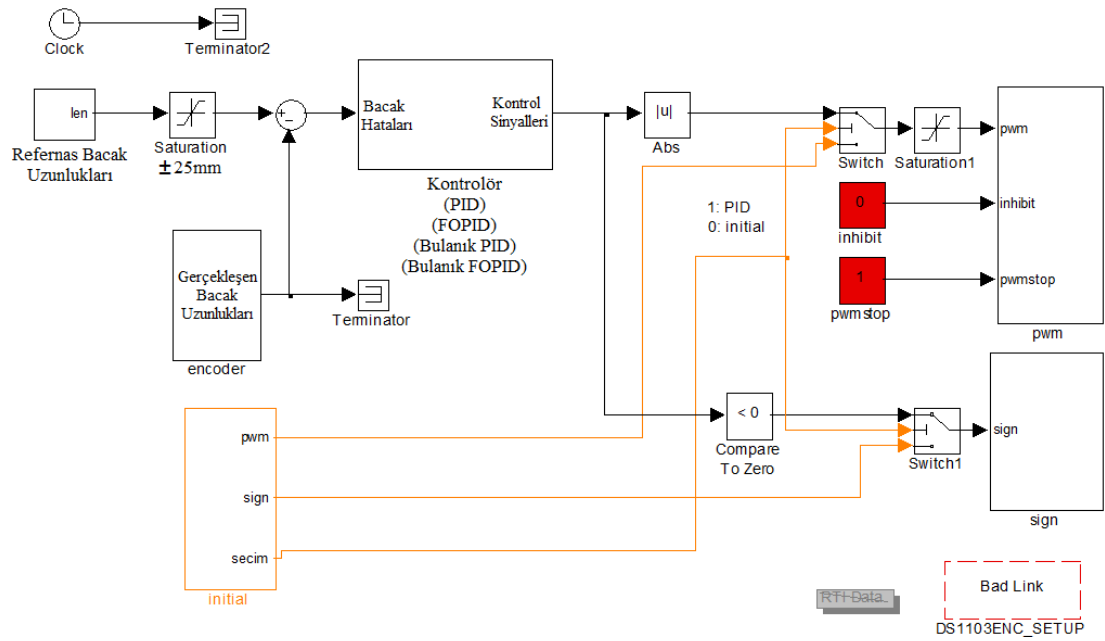
Şekil 6.16. Bulanık FOPID kontrolörün parametre optimizasyonu boyunca maliyet fonksiyonunun yakınsama karakteristiği

6.6. Deneysel Kurulum

Stewart Platform manipülatöründe hareketli platformun konum ve yönelimi, uzamaları ayarlanabilen 6 bacağa bağımlı olarak değişir. Bu manipülatör, boyları prizmatik eklemler vasıtasıyla değişebilen altı bacak ile birbirine bağlı altıgen biçimli bir hareketli platform ve bir sabit platformdan oluşmaktadır. Şekil 5.9’da görüldüğü gibi bu çalışmada kullanılan Stewart Platform’u, altı doğrusal motor, kontrolör, güç kaynağı, acil durdurma devresi ve kontrolör-robot ara geçiş kartından oluşmaktadır.

MATLAB/Simulink ortamında gerçekleştirilen tüm sistemin dinamik modeli kullanılarak bacakların yörünge takip kontrolünü başarmak için PSO algoritması ile parametreleri ayarlanan PID, FOPID, bulanık PID ve bulanık FOPID kontrol yapıları, Şekil 5.9’da gösterilen deneysel robota uygulanacaktır. Ayrıca geliştirilen

kontrolörlerin kontrol performansını değişik yörunge durumları için değerlendirmek ve bu kontrol algoritmalarını deneysel robota uygulamak için dSPACE DS1103 gerçek zamanlı kontrolör kullanılacaktır. Gerçek zamanlı sistemde yer alan hareketli platformun yörungesine göre ters kinematik ile referans bacak uzunlukları hesaplanarak optimize edilen bu kontrol yapıları, deneysel Stewart Platform manipülatörüne eklem uzayında Şekil 6.17’de gösterildiği gibi Simulink ortamında tasarlanmış ana modele ile uygulanacaktır. Bu kontrol modeli için örnekleme zamanı 1ms olarak ayarlanmıştır.



Şekil 6.17. Uygulama deneyleri için sistemin ana kontrol modeli

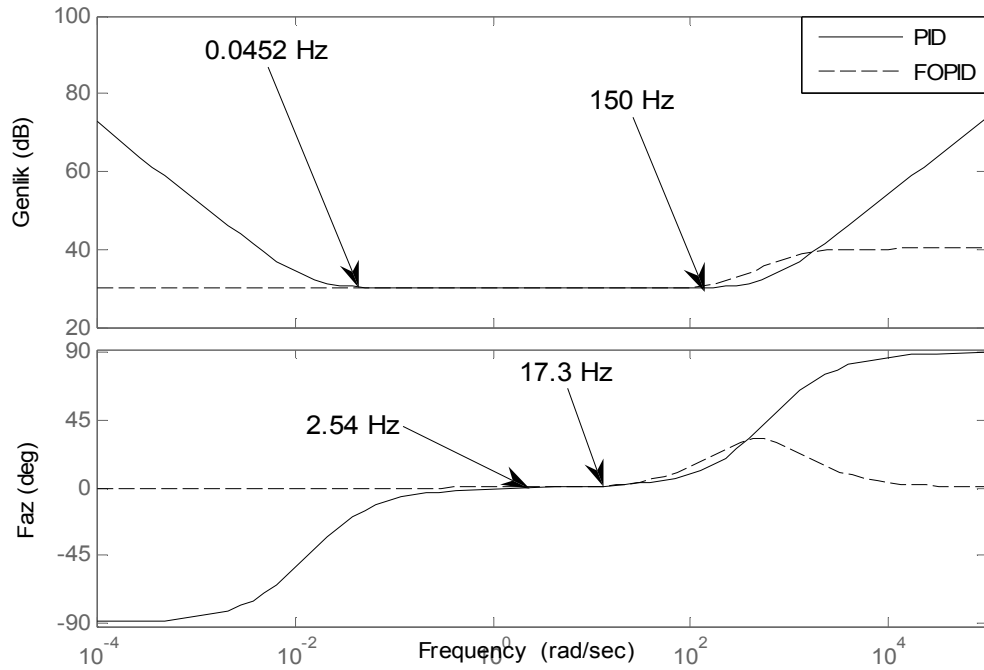
Şekil 6.17, dSPACE kontrolör kartını ve sistemi birbirine entegre eden Simulink modelini göstermektedir. Bu model, Referans Bacak Uzunlukları, encoder, initial, Kontrolör, pwm and sign alt bloklarını içermektedir. Bu alt bloklar sırasıyla: ters kinematik çözümü, bacak uzunluklarının ölçümü, başlatım yordamı, PID, FOPID, Bulanık PID veya Bulanık FOPID konum kontrolü, PWM ve yön çıkış sinyallerinin üretilmesi şeklinde görevleri yerine getirirler.

PSO ile optimize edilen bu kontrol yapılarına ait parametreler, karşılaştırmalı olarak grafiksel ve tablolar şeklinde sırasıyla verilecektir. Buna göre ilk olarak, PSO tabanlı optimize edilen PID ve FOPID kontrolörlerine ait en iyi parametreler, Tablo 6.3’te verilmektedir.

Tablo 6.3. Optimum PID ve FOPID kontrolörlerin kazanç katsayıları

Parametre	FOPID	PID
K _p	31.5841	31
K _i	0.02528	0.41460
K _d	0.03280	0.04990
μ	1.21171	-
λ	0.18487	-

Tablo incelendiğinde her iki kontrol yönteminde, oransal ve türev kazancının birbirine çok yakın değerde olduğu görülmektedir. Kontrolörlerin integral kazançları ise optimizasyon sonucu farklı değerlerde elde edilmiştir. FOPID kontrolünü farklı kılan μ ve λ parametreleri, tanımlanan uzay aralığının ([0 2]) içinde kalmıştır.

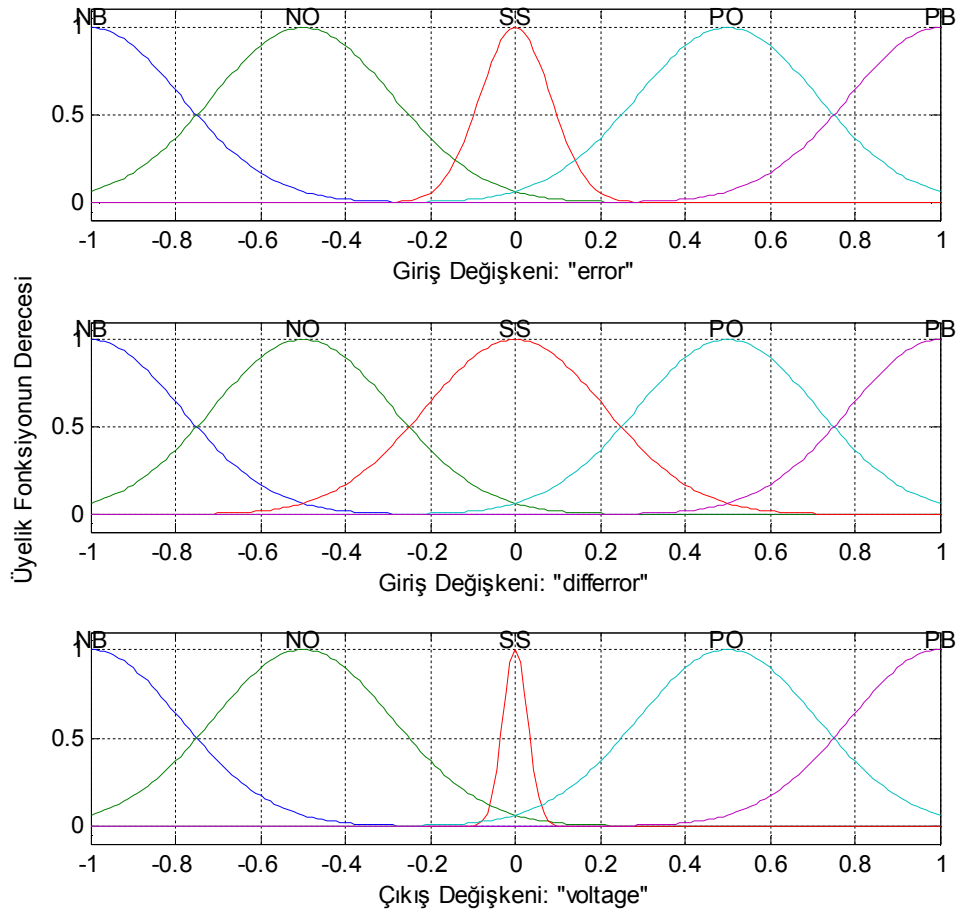


Şekil 6.18. PID ile kesir dereceli PID'nin Bode diyagramlarının karşılaştırılması

Şekil 6.18, parametreleri optimize edilen PID ve FOPID kontrolörlerin genlik ve faz eğrilerini göstermektedir. Şekil gözlemlendiğinde, her iki kontrol yapısının da 2,54 ile 17,3 rad/s faz aralıklarında ve 0,0452 ile 150 rad/s genlik aralıklarında aynı davranışı sergiledikleri görülmektedir. Bu aralıkların dışına çıkıldığında FOPID kontrolörün kazancı, PID kontrolörün kazancına göre çok az değişim göstermektedir. Öte yandan, FOPID kontrolörün fazı sıfıra giderken, PID kontrolörün fazı ise \pm

90°'ye gitmektedir. Şekilden anlaşılacağı üzere geniş frekans aralığında, FOPID kontrolörü, hem faz hem de genlikte daha etkili frekans cevaplarına sahiptir.

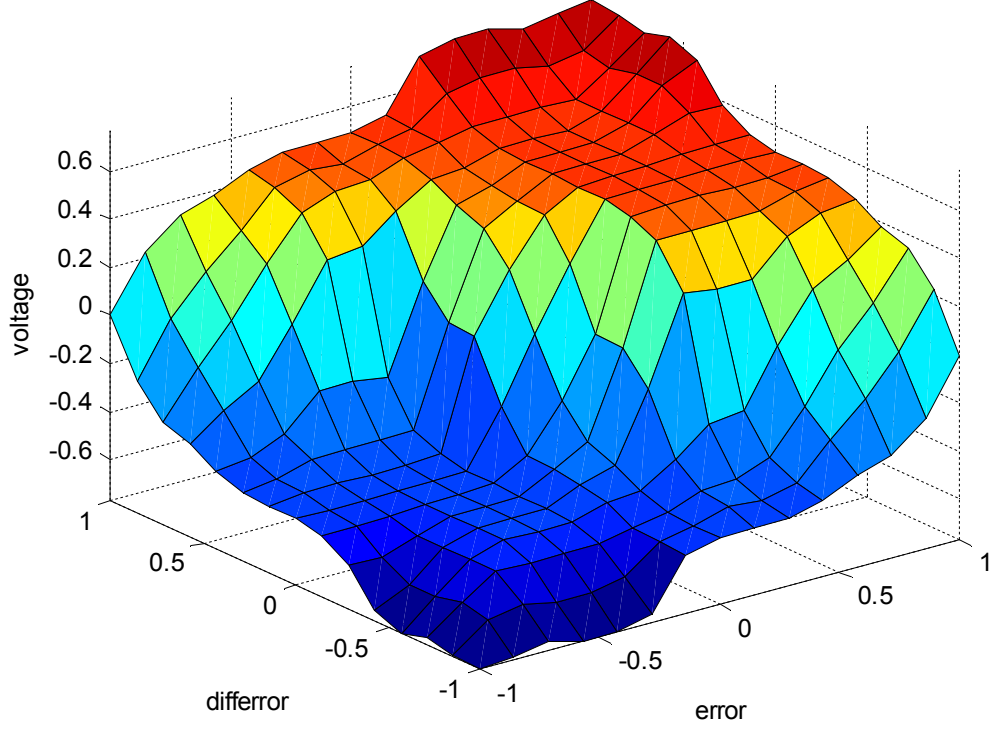
PSO ile parametreleri ayarlanan bulanık PID kontrol yapısının hem giriş hem de çıkış değişkenlerine ait nihai üyelik fonksiyonları ve kontrol yüzeyi sırasıyla Şekil 6.19 ve 6.20'de gösterilmiştir. Şekil 6.19 gözlemlendiğinde giriş değişkenlerindeki üyelik fonksiyonların genişlik değerleri artmış, çıkış değişkeninde ise yalnızca SS dilsel terimli üyelik fonksiyonunun genişlik değerinde azalma olmuştur. Diğer taraftan, error giriş değişkenindeki SS dilsel terimli üyelik fonksiyonunun genişlik değerinde çok az değişim meydana gelmiştir. Ayrıca, optimize edilmiş üyelik fonksiyonların yönelimi, başlangıç üyelik fonksiyonların yönelimi ile karşılaştırıldığında bulanık mantık kontrolörün yüzey şeklini etkilemiştir.



Şekil 6.19. Bulanık PID kontrolörün giriş (error ve differror) ve çıkışına (voltage) ait optimize edilmiş nihai üyelik fonksiyonları

Şekil 6.20'de görüldüğü üzere PSO, belirgin bir şekilde inişli çıkışlı başlangıç kontrol yüzeyini değiştirmiş ve sonuçta, bu üyelik fonksiyonların kullanılmasıyla

başlangıç yüzeyine göre daha dik eğimli bir kontrol yüzeyi elde edilmiştir. Sonuç olarak, üyelik fonksiyonları ve kontrol yüzeyi şekillerinden görülebildiği gibi, optimize edilen parametrelerle, sürüde bulunan parçacıklar, arama uzayı içerisinde uygun bir konuma yönelmiştir.



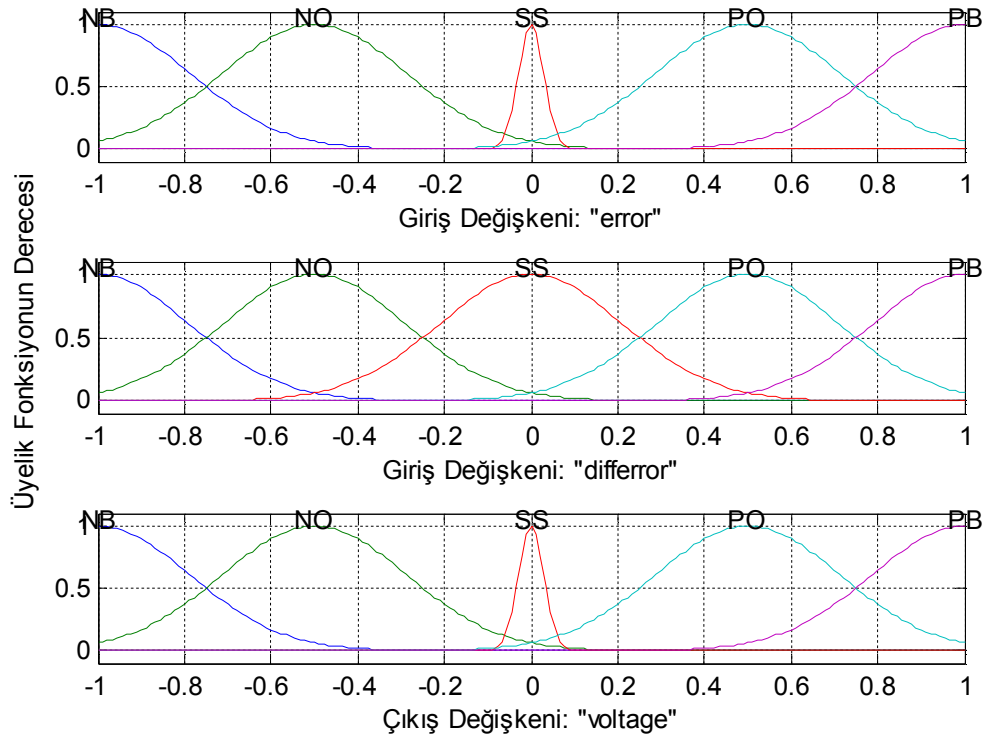
Şekil 6.20. PSO ile optimize edilen bulanık PID kontrolörün kontrol yüzeyi

Son olarak, parametreleri PSO algoritması ile ayarlanan bulanık FOPID kontrolörün giriş ve çıkış değişkenine ait nihai üyelik fonksiyonları ve kontrol yüzeyi sırasıyla Şekil 6.21 ve 6.22’de gösterilmektedir.

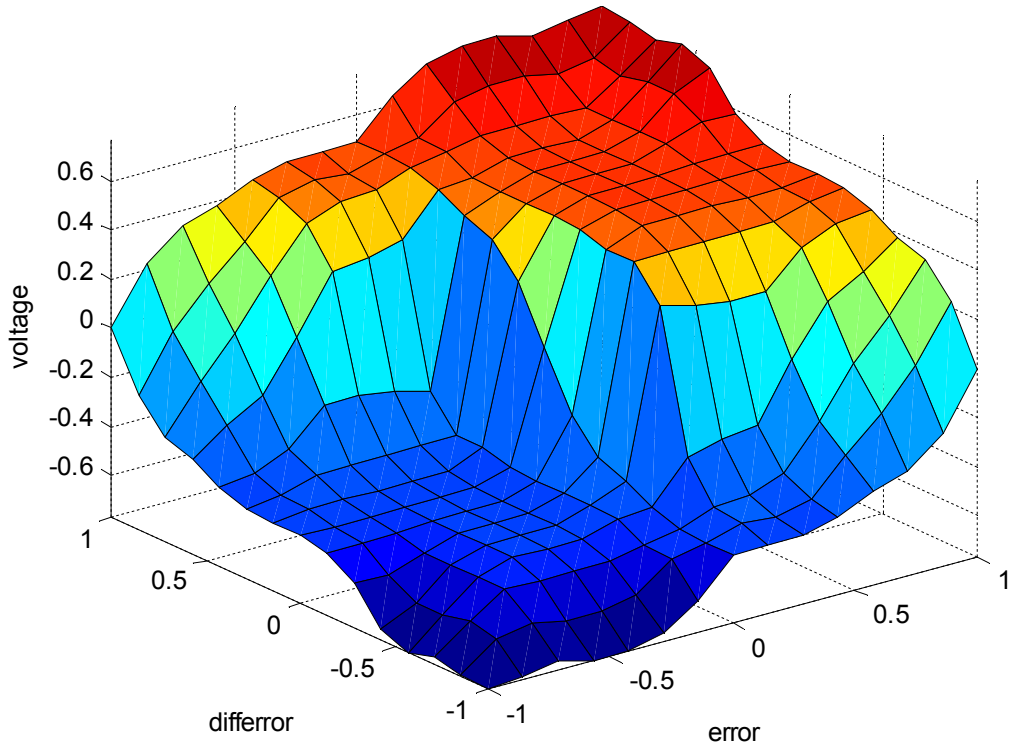
Şekil 6.21’e bakıldığında, optimizasyon sonunda yalnızca error giriş ve voltage çıkış değişkenlerine ait SS dilsel terimli Gauss üyelik fonksiyonunun genişliğinde bir daralma olmuştur. Bu daralma, bulanık olayda daha belirgin bir durumun oluşmasına yol açar. Bu, kontrol hatasının sifıra yakın bölgelerde meydana geldiğini gösteren bir durumdur.

Şekil 6.22’de gösterildiği gibi bulanık FOPID kontrolörün ayarlanan üyelik fonksiyonlarına göre en uygun kontrol yüzeyi elde edilmiştir. Şekle bakıldığında, başlangıç kontrol yüzeyine göre daha sarp eğimin olduğu ve merkeze uzak yerlerde

yüksek düzlüklerin oluştuğu anlaşılmaktadır. Sonuç olarak, elde edilen nihai kontrol yüzeyi, başlangıç kontrol yüzeyinin köşegen düzlemini bozmuştur.



Şekil 6.21. Bulanık FOPID kontrolörün giriş (error ve differror) ve çıkışına (voltage) ait optimize edilmiş nihai üyelik fonksiyonları



Şekil 6.22. PSO ile optimize edilen bulanık FOPID kontrolörün kontrol yüzeyi

PSO algoritmasına göre ayarlanan bulanık PID ve bulanık FOPID kontrolörlerin giriş ve çıkış kazanç parametreleri, Tablo 6.4’de verilmektedir.

Tablo 6.4. Bulanık PID ve bulanık FOPID kontrolörlerin optimum giriş ve çıkış kazanç katsayıları

Kazanç Katsayısı	Bulanık FOPID	Bulanık PID
Ge	5.2108	5.1075
Gce	0.0119	0.0118
Gi	33.3909	30.3900
Gu	1.669	1.5369
μ	0.98	-
λ	0.808	-

Tablo incelendiğinde her iki kontrol yapısında, hatanın türevi için elde edilen giriş kazanç katsayısının çok düşük değerde, kontrolör çıkışındaki integral kazanç katsayısının ise diğerlerine göre çok yüksek değerde elde edildiği görülmektedir. Ayrıca, optimizasyon sonunda kontrolörlere ait giriş ve çıkış kazanç katsayılarının birbirine yakın değerlerde elde edildiği tablodan anlaşılmaktadır. Kontrol sinyalleri, doğrudan çıkış kazanç katsayıları ile çarpıldığından, kontrol sinyalini küçük seviyede tutmak için bulanık kontrolörlerin çıkışının düşük kazançlı olması istenir. Bu sebeple, optimizasyon sonunda elde edilen en uygun giriş kazanç katsayıları, kontrolörün tüm giriş değerleri için bulanık çıkarımın, daima arzu edilen giriş değişkenleri (error ve differror) kümesi uzayı içerisinde kaldığını garanti etmektedir. Ayrıca, bulanık PID kontrolörün genişletilmiş şekli olan bulanık FOPID kontrolör yaklaşımındaki optimum kesir dereceli türev ve integral, kontrolörün performansını belirgin bir şekilde etkilemektedir.

Yukarıda bahsedilen kontrol yapıların parametrelerini ayarlama işlemi için MATLAB ortamında yazılan kodlar, EK-B’de PIDTuningPSO, FOPIDTuningPSO, BulanıkPIDTuningPSO, BulanıkFOPIDTuningPSO isimleri altında verilmiştir.

6.7. Deneysel Sonuçlar

Bu bölümde, parametreleri PSO algoritmasına göre ayarlanan kontrolörlerin eklem esaslı yörünge kontrolüne dayanan yörünge takip doğruluğunu, sağlamlığını ve

kararlılığını değerlendirmek amacıyla gerçek robot üzerinde deneyler yapılacaktır. Her bir deney için hareketli platformun yörünge denklemi verilerek, referans bacak konumları ile optimize edilen kontrolör altındaki robotun gerçekleştirdiği konumlar ve bacaklarda meydana gelen konum hataları, grafiksel olarak gösterilecektir. Ayrıca, daha doğru yörünge takibini başarabilen kontrolörleri belirlemek için deneylerde meydana gelen kontrol hataları, her bir bacak için hesaplanacaktır. Bunun yanında, deneylerden elde edilen örnekler kullanılarak kontrol sinyallerin karekök ortalama (Root Mean Square-RMS) değerleri de, her bir bacak için elde edilecektir.

Aşağıdaki gibi tanımlanan ortalama mutlak hata (Mean of Absolute Error–MAE) kontrol hatalarını, U_{RMS} ise üretilen kontrol sinyallerini belirlemek için kullanılacaktır.

$$MAE_j = \frac{1}{N} \sum_i^N |e_j(i)| \quad (6.7)$$

$$U_{RMS_j} = \sqrt{\frac{1}{N} \sum_i^N (u(i))^2} \quad (6.8)$$

Denklemlerde yer alan e , meydana gelen yörünge hatasını, j , ilgili eklem numarasını, N , deney zamanı bitiminde elde edilen örnek sayısını, u ise üretilen kontrol sinyalini göstermektedir.

PSO tabanlı kontrolörlerin dinamik kontrol performanslarını karşılaştırmak amacıyla her bir bacakta yer alan kontrolörlerin ürettiği kontrol sinyallerin RMS değerleri ve yaptığı yörünge hataların MAE değerleri, yapılan tüm deneyler için tablolar şeklinde verilecektir. Yapılan tüm deneylerde hareketli platformun yörüngesi, hareketin zorluğuna dayalı olarak seçilecektir.

6.7.1. Yörünge takip deneyi 1

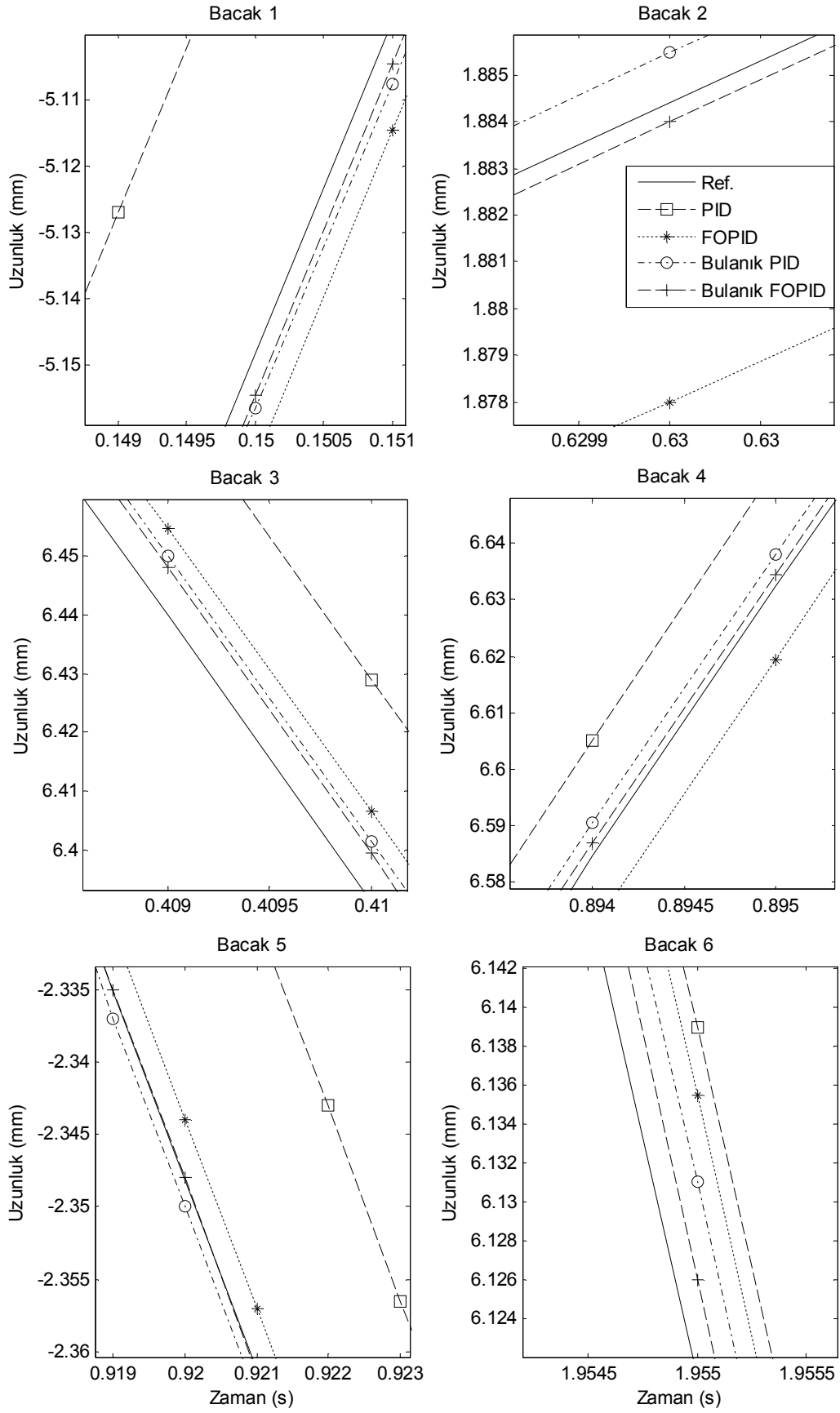
İlk olarak, hareketli platformun x-y-z eksenlerinde dönme hareketini sağlayan yörünge denklemi aşağıda verildiği gibi seçilmiştir.

$$\begin{aligned}
x(t) &= 0 & \alpha(t) &= 2 \cos(2\pi t) \\
y(t) &= 0 & \beta(t) &= 5 \sin(2\pi t) \\
z(t) &= 0.25 & \gamma(t) &= 10 \cos(2\pi t)
\end{aligned} \tag{6.9}$$

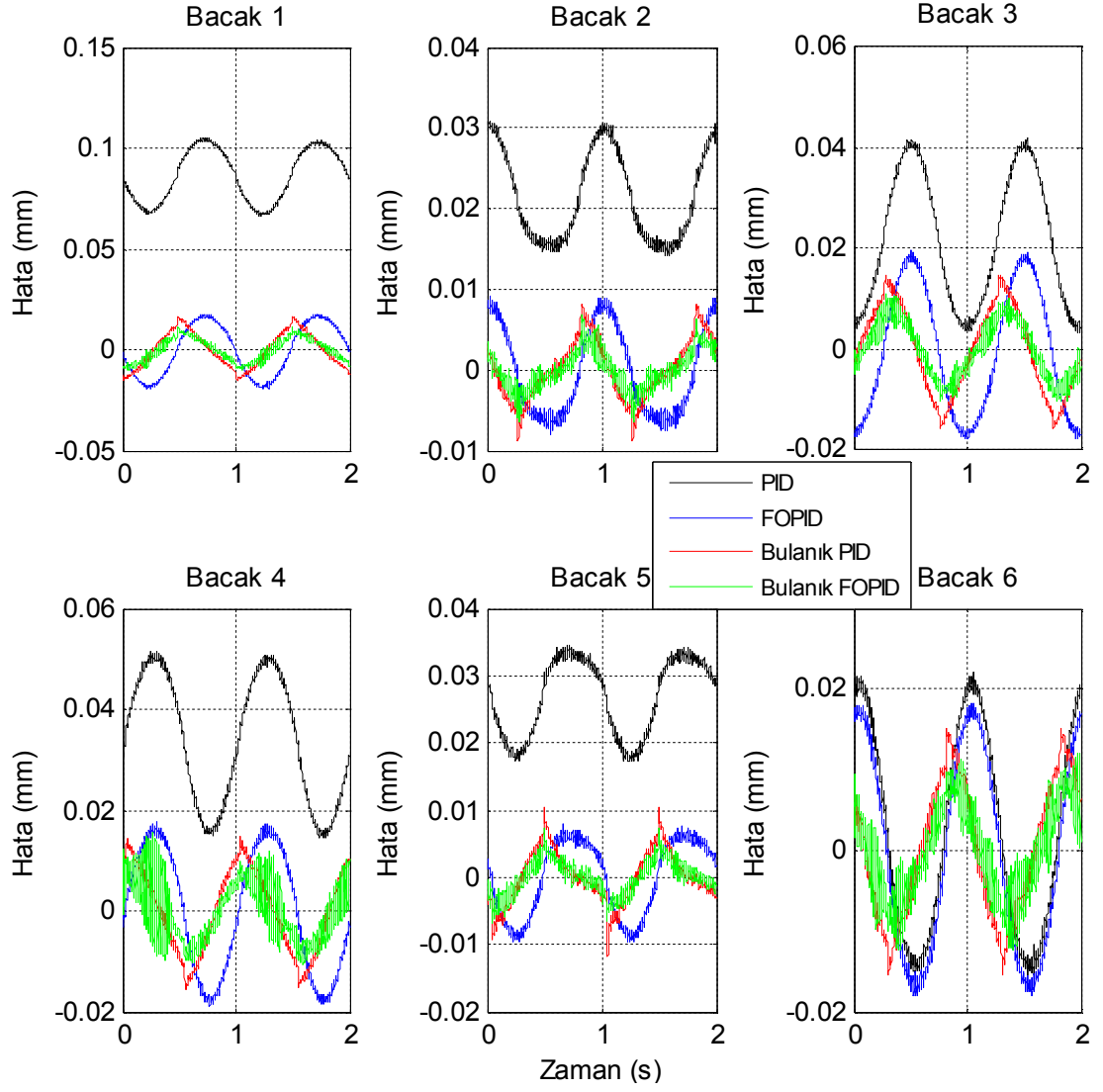
Şekil 6.17’de gösterilen kontrol ana modelinde, verilen yörünge denklemine göre ters kinematik sonucu referans bacak uzunlukları elde edilir. Encoder’lardan Stewart Platform’un gerçekleştirdiği bacak uzunlukları ile referans olarak girilen bacak uzunlukları karşılaştırılarak hata sinyali elde edilir. Bu hata sinyali, her bir bacakta yer alan PSO tabanlı PID, FOPID, bulanık PID ve bulanık FOPID kontrolörlerine giriş olarak verilir. Kontrolörler ise bu girişe göre sisteme kontrol sinyali uygulayarak sistemin çıkışını, referans sinyaline yaklaştırmaya çalışırlar. Yapılan bu deney ve diğer tüm deneyler için veri toplama zamanı 2 saniye olarak belirlenmiştir. Ancak kontrolörlerin vermiş olduğu cevapları daha net görmek amacıyla verilen bu süre içerisinde belirli aralıklar seçilerek deney sonuçları çizdirilmiştir.

PSO algoritmasına göre ayarlanan PID, FOPID, bulanık PID ve bulanık FOPID kontrolörlerin yörünge takip performansları, Şekil 6.23’te gösterilmektedir. Şekil incelendiğinde PSO tabanlı bulanık FOPID kontrolörü, PSO tabanlı diğer 3 kontrolörden daha iyi performans gösterdiği anlaşılmaktadır. Bulanık FOPID ile bulanık PID kontrolörlerin eklem bazında vermiş olduğu cevapların birbirine yakın olduğu görülmektedir. Ayrıca bulanık FOPID kontrolörün, PSO algoritması ile optimize edilen üyelik fonksiyonların, kazanç katsayıların ve de kesir dereceli türev ve integratörün doğrultusunda sistemin verdiği cevabın diğer kontrolörlere kıyasla, istenen referans çizgisine ulaştığı ve referans çizgisini en az hatayla takip edebildiği gözlenmiştir.

Yörünge takibi sırasında bacaklarda meydana gelen konum hataları, Şekil 6.24’te gösterilmektedir. Şekle bakıldığında, bulanık FOPID kontrolörün daha iyi yörünge izleme hata karakteristiğine sahip olduğu görülmektedir. Ayrıca, kalıcı takip sağlandığında bulanık FOPID kontrolörü altındaki bacakların takip hatası, diğerlerine göre $\pm 0,0157$ mm’den daha az aralıklarda değişmektedir.



Şekil 6.23. Deney 1'e göre kontrol altındaki bacakların yörünge takibi



Şekil 6.24. Deney 1'e göre kontrolörlerin yörünge takip hataları

Bu deneyden elde edilen kontrol sinyallerin RMS ve yörünge takip hatalarının MAE değerleri, Tablo 6.5'te verilmektedir.

Tablo 6.5. Deney 1'e göre kontrolörlerin performans karşılaştırması

Eklem	PID		FOPID		Bulanık PID		Bulanık FOPID	
	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}
Bacak 1	0.0865	0.4093	0.0116	0.4025	0.0075	0.3938	0.0049	0.4162
Bacak 2	0.0214	0.1723	0.0049	0.1757	0.0030	0.1665	0.0019	0.1633
Bacak 3	0.0219	0.4040	0.0115	0.4010	0.0076	0.3956	0.0047	0.3997
Bacak 4	0.0337	0.3867	0.0111	0.4010	0.0072	0.3750	0.0053	0.3921
Bacak 5	0.0268	0.1809	0.0052	0.1830	0.0032	0.1732	0.0021	0.1736
Bacak 6	0.0115	0.3922	0.0112	0.3901	0.0073	0.3805	0.0048	0.3982

Tabloda verilen 4 farklı kontrolörün MAE değerleri karşılaştırıldığında, PSO tabanlı bulanık FOPID kontrolörü, PSO tabanlı PID, FOPID ve bulanık PID kontrolörlerinden daha başarılı sonuçlar verdiği görülmektedir. Kontrol sinyallerin RMS değerleri incelendiğinde, tüm bacaklarda kontrolörler, hemen hemen aynı sonuçları vermişlerdir. Deney sonuçlarına bakıldığında, bulanık FOPID kontrolörün, hemen hemen aynı kontrol sinyallerinde daha doğru yörünge takip performansını başarabildiği görülmektedir.

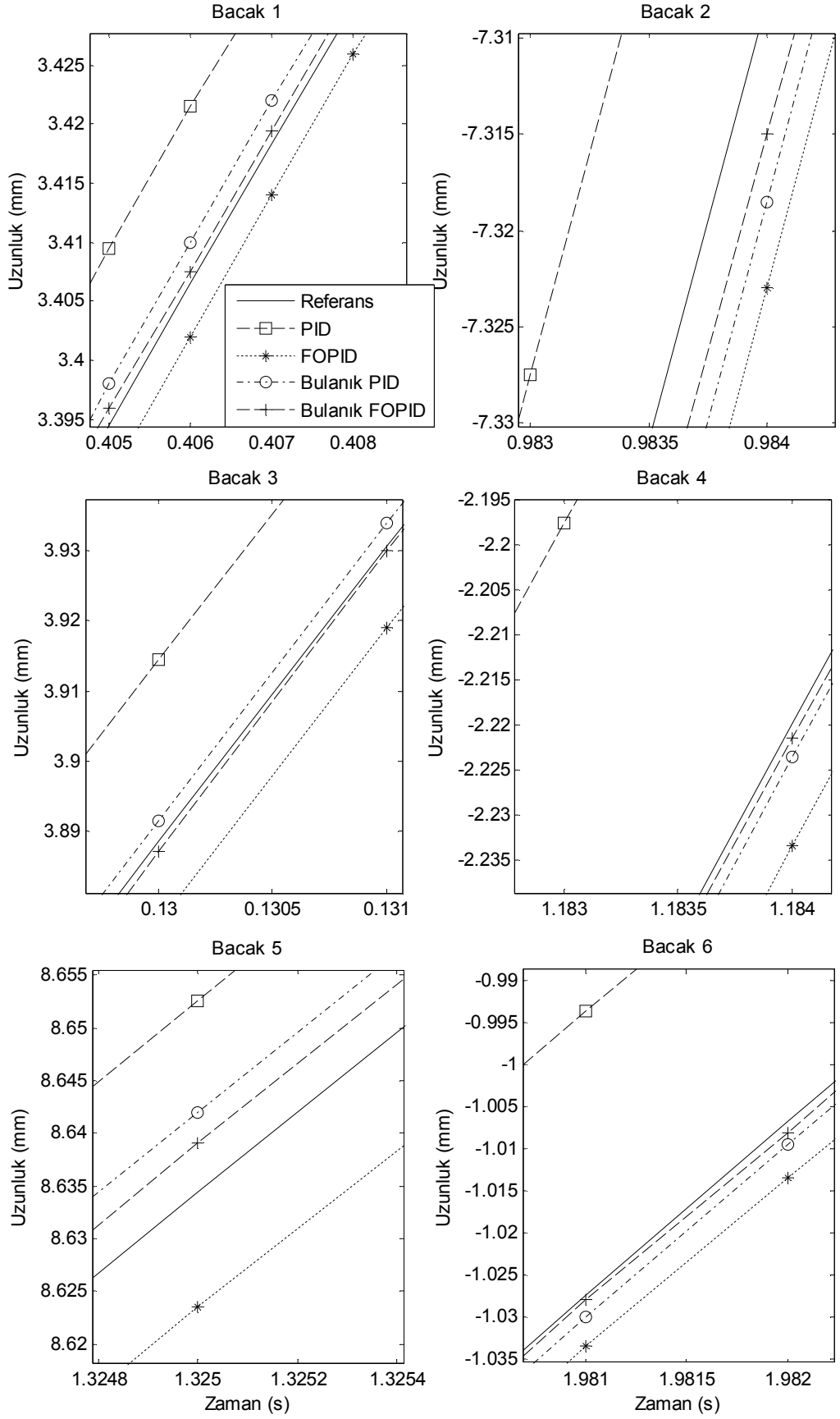
6.7.2. Yörünge takip deneyi 2

Bu deneyde hareketli platform, yalnızca x-y-z eksenlerinde öteleme yaparken kontrolörlerin performans karşılaştırılması yapılacaktır. Bu ötelemeyi sağlayan yörünge denklemi aşağıda verildiği gibi seçilmiştir.

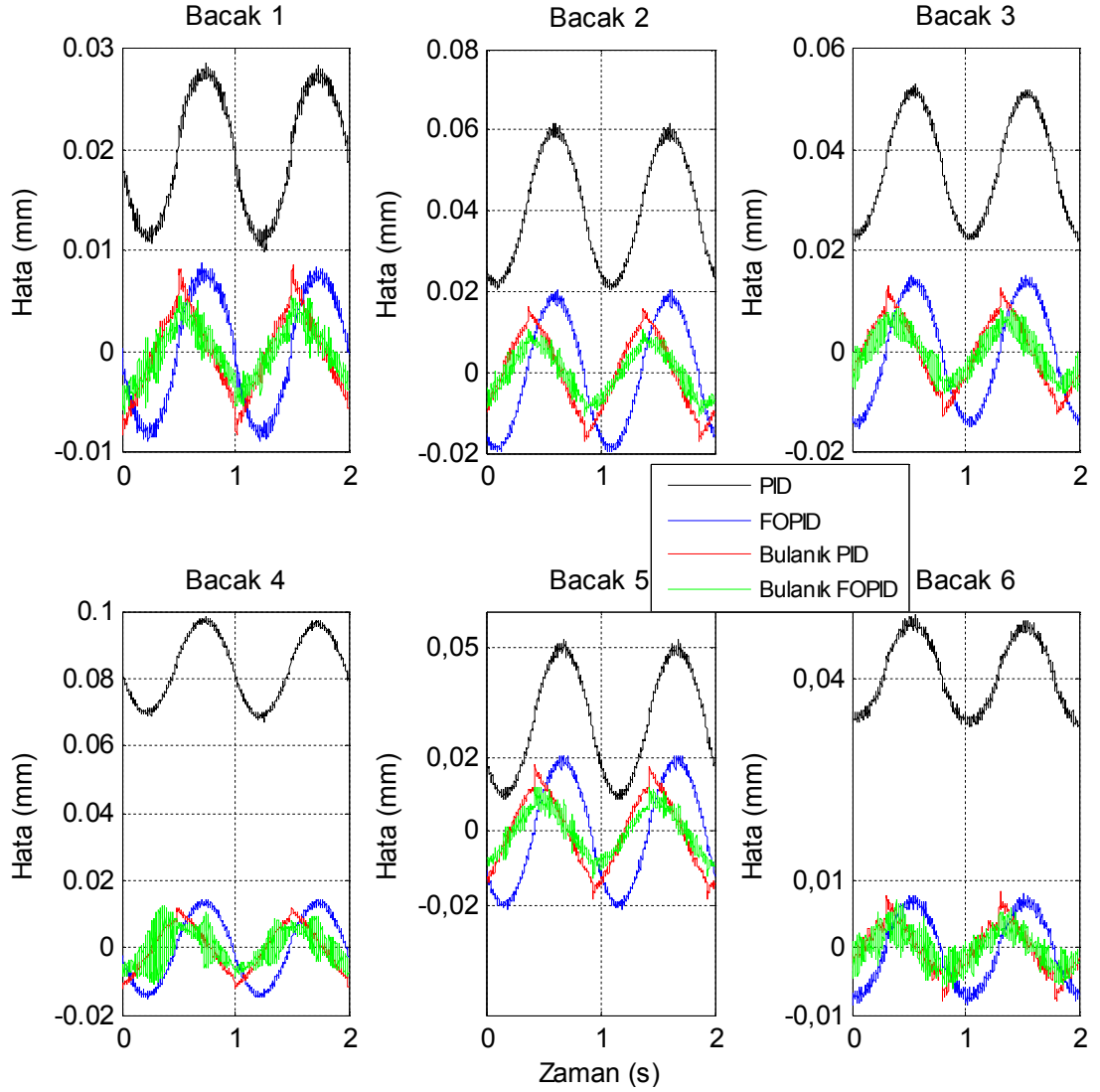
$$\begin{aligned}
 x(t) &= 12 \sin(2\pi t) & \alpha(t) &= 0 \\
 y(t) &= 12 \cos(2\pi t) & \beta(t) &= 0 \\
 z(t) &= 0.25 + 7 \sin(2\pi t) & \gamma(t) &= 0
 \end{aligned} \tag{6.10}$$

Yukarıda verilen referans yörüngenin cevabında, PSO ile ayarlanan PID, FOPID, bulanık PID ve bulanık FOPID kontrolörlerin bacak yörünge takip performansları, Şekil 6.25'te görülmektedir. Şekil incelendiğinde, bulanık FOPID ve bulanık PID kontrolörleri diğer kontrolörler ile karşılaştırıldığında, kalıcı doğrulukta daha üstün kontrol performansı göstermişlerdir. Ayrıca, bulanık FOPID kontrolörü, referans bacak yörüngelerine hızlı ve kararlı bir şekilde tüm bacaklarda cevap verebilmiştir.

Şekil 6.26, parametreleri ayarlanan kontrolörler altındaki bacakların yörünge takip hatalarını göstermektedir. Şekil 6.26 incelendiğinde hem PSO tabanlı bulanık FOPID hem de PSO tabanlı bulanık PID kontrolörlerin hata değişim aralığı, PSO tabanlı PID ve FOPID kontrolörlerine göre daha düşük olduğu görülmektedir. Ayrıca, bulanık FOPID kontrolörü, diğer kontrolörler ile karşılaştırıldığında düşük hata değeri ile daha doğru takip cevabını başarabilmiştir. Kontrolörlerin takip performanslarını değerlendirmek amacıyla her bacak için MAE ve RMS değerleri, Tablo 6.6'da verilmektedir.



Şekil 6.25. Deneý 2'ye göre kontrol altındaki bacakların yörünge takibi



Şekil 6.26. Deney 2'ye göre kontrolörlerin yörünge takip hataları

Tablo incelendiğinde, bulanık FOPID kontrolörün MAE değerlerinin daha düşük olduğu görülmektedir. Bu sonuç, sistemin yörünge takip performansını arttırmak için önerilen bulanık FOPID kontrolörün geçerliliğini göstermektedir.

Tablo 6.6. Deney 2'e göre kontrolörlerin performans karşılaştırması

Eklem	PID		FOPID		Bulanık PID		Bulanık FOPID	
	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}
Bacak 1	0.0196	0.1916	0.0054	0.1885	0.0033	0.1804	0.0023	0.1915
Bacak 2	0.0405	0.4267	0.0123	0.4269	0.0080	0.4185	0.0049	0.4047
Bacak 3	0.0370	0.3215	0.0092	0.3209	0.0060	0.3146	0.0038	0.3185
Bacak 4	0.0831	0.3161	0.0090	0.3209	0.0058	0.3061	0.0046	0.3235
Bacak 5	0.0299	0.4495	0.0129	0.4478	0.0084	0.4366	0.0053	0.4370
Bacak 6	0.0406	0.1660	0.0047	0.1684	0.0030	0.1614	0.0022	0.1709

Tabloda verilen her bir bacak için kontrol sinyallerin RMS değerleri karşılaştırıldığında, tüm bacaklarda kontrolörler, hemen hemen aynı sonuçları vermişlerdir. Sonuç olarak, kontrolörler birbirine yakın kontrol girişleri üretmelerine rağmen yörünge değişikliği altında bulanık FOPID kontrolörün sağlamlık ve verimlilik açısından başarımı diğerlerine göre daha üstündür

6.7.3. Yörünge takip deneyi 3

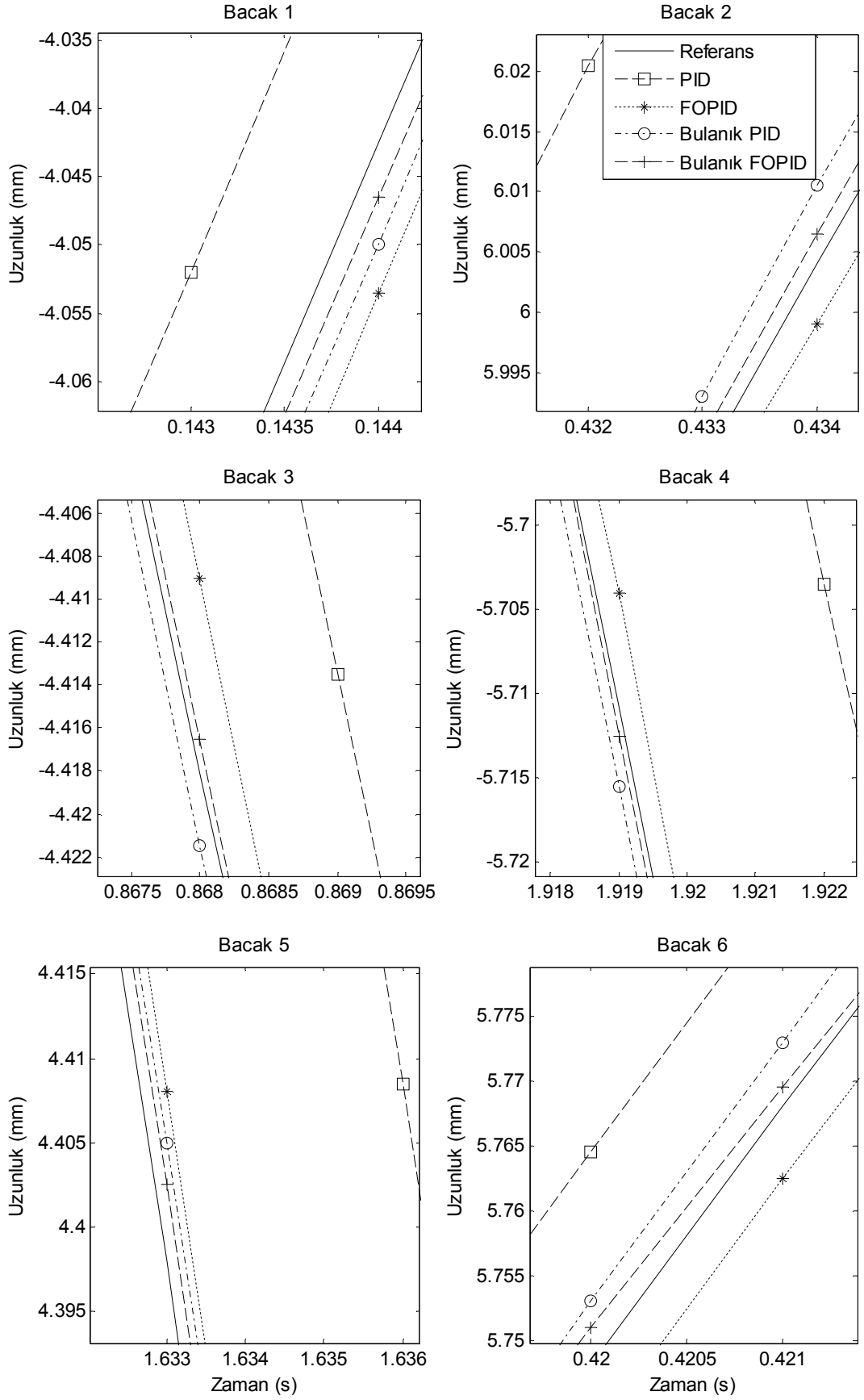
Bu deneyde, tek bir eksen de hareketi sağlayan yörünge seçilmiştir. Bu yörünge denklemini aşağıda verildiği gibidir.

$$\begin{aligned} x(t) &= 0 & \alpha(t) &= 0 \\ y(t) &= 0 & \beta(t) &= 0 \\ z(t) &= 0.25 + 7 \sin(2\pi t) & \gamma(t) &= 0 \end{aligned} \quad (6.11)$$

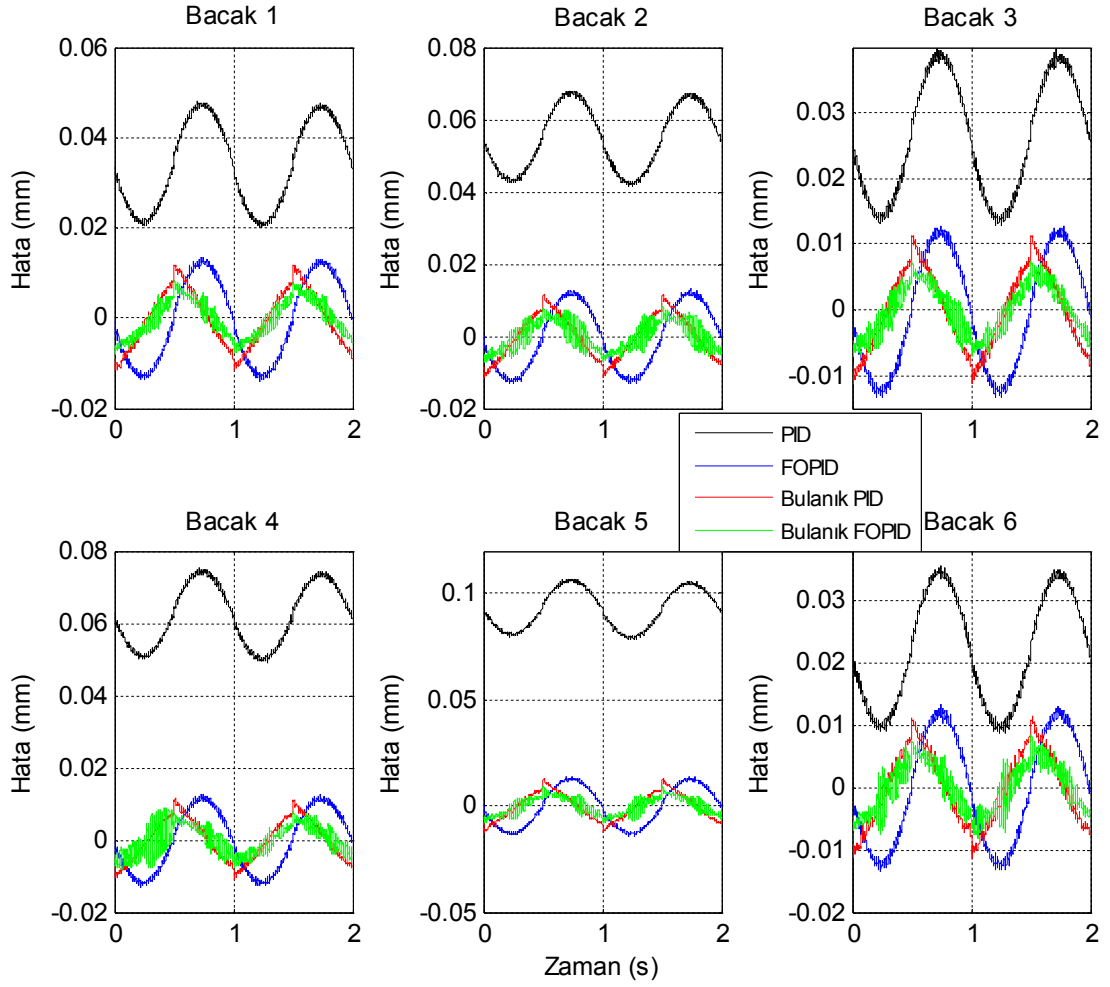
Verilen yörünge denklemine göre ters kinematik ile hesaplanan referans bacak uzunlukları ile PSO tabanlı PID, FOPID, bulanık PID ve bulanık FOPID kontrolörleri altında, bacakların vermiş olduğu dinamik cevaplar Şekil 6.27’de gösterilmektedir.

Şekil 6.27 incelendiğinde PSO tabanlı bulanık FOPID kontrolörün yörünge takip performansı, PSO tabanlı diğer kontrolörlere göre daha iyi olduğu görülmektedir. Ayrıca, bulanık FOPID kontrolörü ile bulanık PID kontrolörün vermiş olduğu cevapların birbirine yakın olduğu ve başarılı bir şekilde sistemi kontrol altında tutabildikleri anlaşılmaktadır.

Parametreleri optimize edilen kontrol yapıları altında bacakların dinamik yörünge takip hataları, Şekil 6.28’de gösterilmektedir. Şekil incelendiğinde, bacaklardaki en yüksek dinamik yörünge hataları, PID, FOPID, bulanık PID ve bulanık FOPID kontrolörleri için sırasıyla 0,1071 mm, 0,0137 mm, 0,130 mm ve 0,0095 mm değerlerinde olduğu görülmektedir. Bu sonuç, bulanık FOPID kontrolörün iyi bir kontrol performansı sergilediğini göstermektedir. Bu deneyden elde edilen kontrol sinyallerin RMS ve yörünge takip hataların MAE değerleri, Tablo 6.7’de verilmektedir.



Şekil 6.27. Deney 3'e göre kontrol altındaki bacakların yörünge takibi



Şekil 6.28. Deney 3'e göre kontrolörlerin yörünge takip hataları

Tabloya bakıldığında, bulanık FOPID kontrolörün toplam yörünge takip hataları diğerlerine göre daha düşük olduğu görülmektedir. Ayrıca, kontrolörlerin tüm eklemlerdeki MAE değerlerine bakıldığında PID kontrolörü, tüm eklemlerin kontrolünde aynı kararlı yapıyı sağlayamadığı anlaşılmaktadır.

Tablo 6.7. Deney 3'e göre kontrolörlerin performans karşılaştırması

Eklem	PID		FOPID		Bulanık PID		Bulanık FOPID	
	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}
Bacak 1	0.0340	0.2995	0.0085	0.2939	0.0054	0.2893	0.0036	0.3011
Bacak 2	0.0548	0.2809	0.0081	0.2815	0.0052	0.2746	0.0036	0.2689
Bacak 3	0.0263	0.2789	0.0080	0.2790	0.0052	0.2742	0.0033	0.2752
Bacak 4	0.0622	0.2709	0.0077	0.2790	0.0050	0.2622	0.0035	0.2710
Bacak 5	0.0927	0.2973	0.0085	0.2970	0.0055	0.2885	0.0038	0.2896
Bacak 6	0.0220	0.2796	0.0080	0.2796	0.0052	0.2723	0.0035	0.2830

Tablo incelendiğinde kontrolörlerin RMS değerlerindeki yakınlık, tüm bacaklar için hemen hemen aynı kontrol sinyalinin üretildiğini gösterir. Bu deneysel sonuçlar, birbirine yakın üretilen kontrol sinyallerinde, bulanık FOPID kontrolörün daha doğru yörünge takip performansını başarabildiğini göstermiştir.

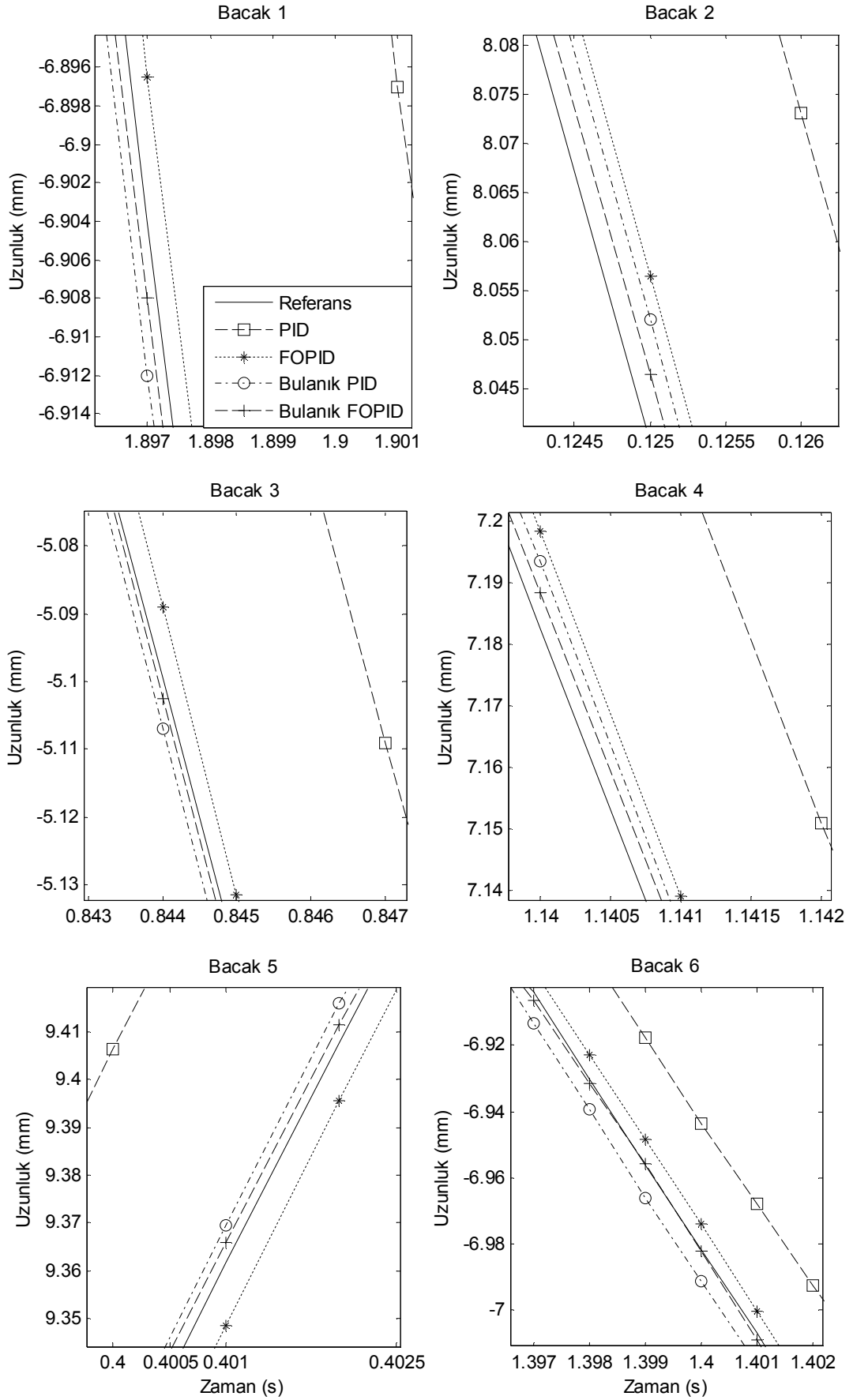
6.7.4. Yörünge takip deneyi 4

Bu deneyde, tek bir ekseninde dönme hareketini sağlayan yörünge seçilmiştir. Bu dönme hareketini veren yörünge denklemi, aşağıda verildiği gibidir.

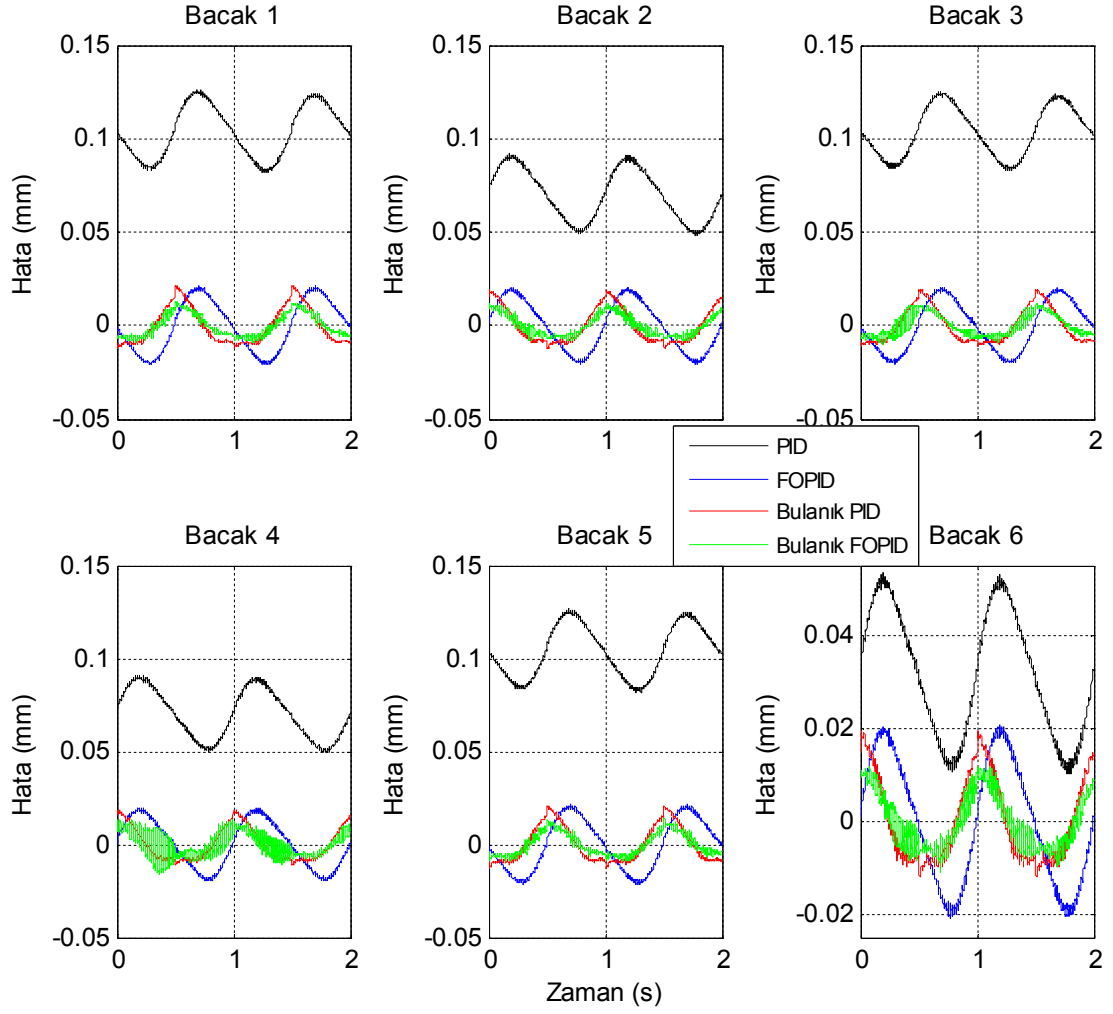
$$\begin{aligned}x(t) &= 0 & \alpha(t) &= 0 \\y(t) &= 0 & \beta(t) &= 0 \\z(t) &= 0.25 & \gamma(t) &= 18 \cos(2\pi t)\end{aligned} \quad (6.12)$$

Şekil 6.29, ters kinematik sonucu elde edilen referans bacak uzunlukları ile PID, FOPID, bulanık PID ve bulanık FOPID kontrolörleri altındaki bacakların gerçekleştirdiği bacak uzunluklarını göstermektedir. Şekil incelendiğinde PSO tabanlı bulanık FOPID kontrolörün yörünge takibi, PSO tabanlı diğer kontrolörlere göre daha iyi olduğu görülmektedir. Ayrıca Şekil 6.29, bulanık FOPID ve bulanık PID ile bacakların referans yörüngelerini çok daha yakın takip ettiğini göstermektedir.

Şekil 6.30, parametreleri ayarlanan kontrolörler altındaki bacakların yörünge takip hatalarını göstermektedir. Şekil incelendiğinde hem PSO tabanlı bulanık FOPID hem de PSO tabanlı bulanık PID kontrolörlerin hata değişim aralığı, PSO tabanlı PID ve FOPID kontrolörlerine göre daha düşük olduğu görülmektedir. Bulanık FOPID kontrol yapısı, $\pm 0,0160$ mm hata aralıkları ile bacakları istenilen konumda tutmayı başarabilmiştir. Ayrıca, PID kontrol yapısı ile bacaklarda büyük kalıcı durum hataları meydana gelirken diğer kontrolörler, daha ufak hatalarla yörünge takibini yerine getirebilmiştir. Bu deneyden elde edilen kontrol sinyallerin RMS ve yörünge takip hataların MAE değerleri, Tablo 6.8'de verilmektedir.



Şekil 6.29. Deney 4'e göre kontrol altındaki bacakların yörünge takibi



Şekil 6.30. Deney 4'e göre kontrolörlerin yörünge takip hataları

Tabloya incelendiğinde bulanık FOPID kontrolörün en iyi MAE değerlerini gerçekleştirerek diğer kontrolörlerden daha sağlam kontrol performansını sergilediği görülmektedir. Diğer taraftan, PID kontrolörün MAE değerlerine bakıldığında, tüm eklemlerin kontrolünde aynı kararlı yapıyı sağlayamadığı anlaşılmaktadır.

Tablo 6.8. Deney 4'e göre kontrolörlerin performans karşılaştırması

Eklem	PID		FOPID		Bulanık PID		Bulanık FOPID	
	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}
Bacak 1	0.1038	0.4392	0.0122	0.4327	0.0086	0.4282	0.0053	0.4407
Bacak 2	0.0700	0.4179	0.0117	0.4165	0.0082	0.4096	0.0049	0.3976
Bacak 3	0.1039	0.4170	0.0117	0.4174	0.0082	0.4114	0.0051	0.4130
Bacak 4	0.0700	0.4031	0.0113	0.4174	0.0079	0.3928	0.0057	0.4100
Bacak 5	0.1039	0.4401	0.0124	0.4387	0.0086	0.4286	0.0053	0.4285
Bacak 6	0.0315	0.4183	0.0118	0.4188	0.0082	0.4078	0.0052	0.4227

Tabloda verilen RMS değerlerine bakıldığında, kontrolörlerin tüm bacaklarda ürettiği kontrol sinyallerinin birbirine yakın olduğu görülmektedir. Sonuç olarak, hemen hemen aynı kontrol sinyalinin üretildiği kontrol yapılarında, PSO tabanlı bulanık FOPID kontrol yapısı ile bacakların daha iyi takip performansı gerçekleştirilmiştir.

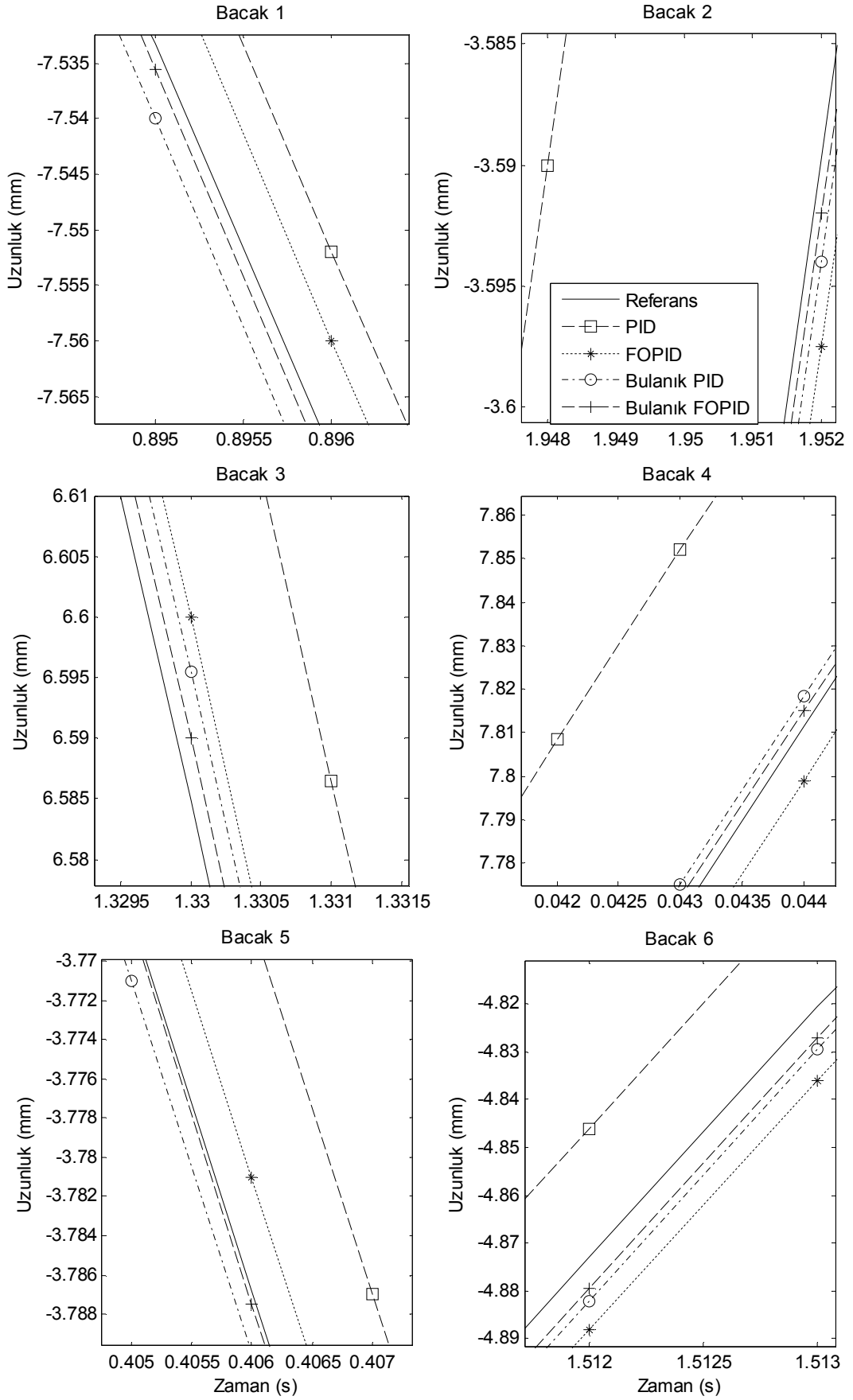
6.7.5. Yörünge takip deneyi 5

Bu deneyde, tüm eksenlerde hareketi sağlayan yörünge denklemi aşağıda verildiği gibi seçilmiştir.

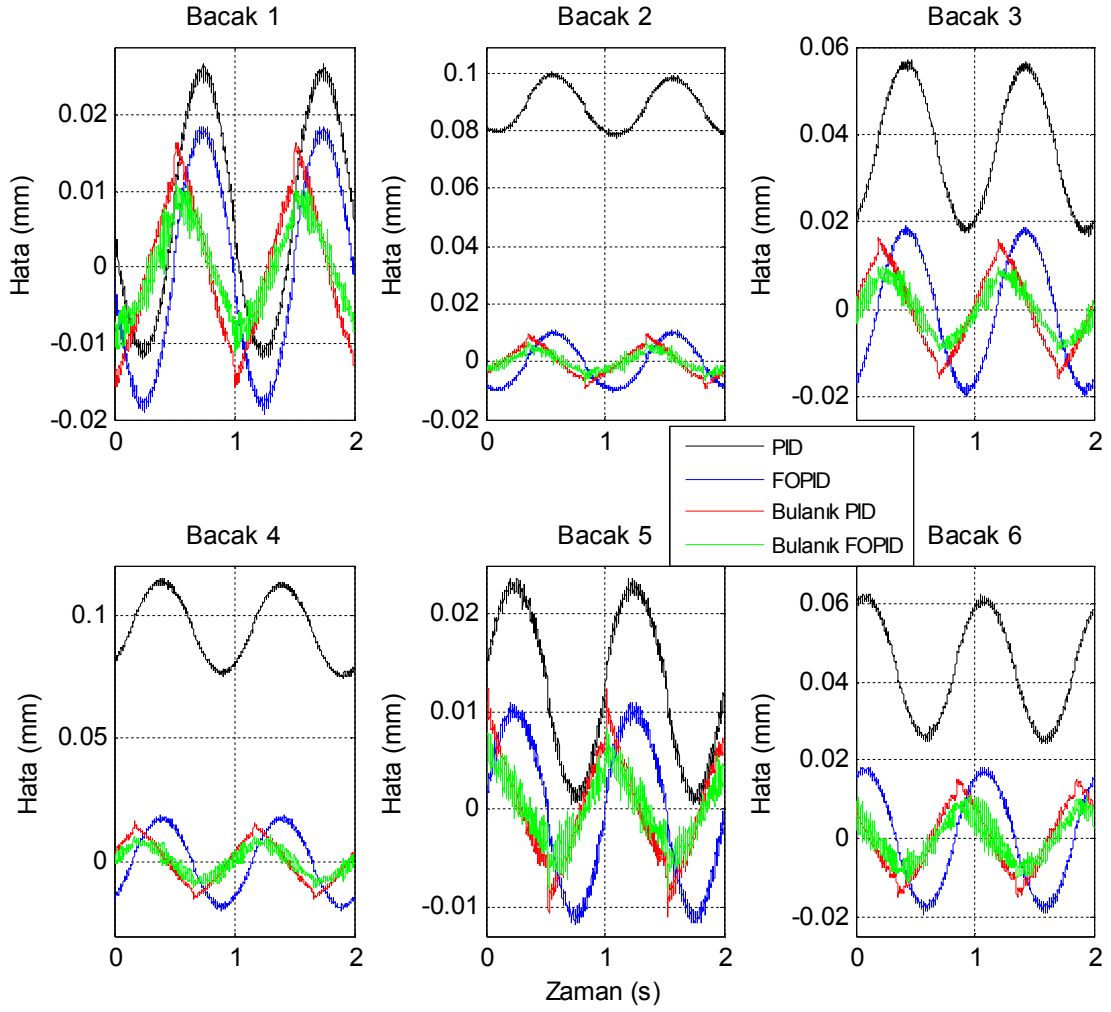
$$\begin{aligned}x(t) &= 7 \sin(2\pi t) & \alpha(t) &= 5 \cos(2\pi t) \\y(t) &= 7 \cos(2\pi t) & \beta(t) &= 5 \sin(2\pi t) \\z(t) &= 0.25 + 3 \sin(2\pi t) & \gamma(t) &= 3 \cos(2\pi t)\end{aligned}\tag{6.13}$$

Şekil 6.31, bu referans yörünge için cevabında PSO tabanlı PID, FOPID, bulanık PID ve bulanık FOPID kontrolörlerin bacak yörünge takip performansını göstermektedir. Şekil gözlemlendiğinde PSO tabanlı bulanık FOPID kontrolörü, bacakların tümünde daha iyi kontrol performansı sergilediği görülmektedir. Ayrıca, diğer PSO tabanlı kontrolörler ile karşılaştırıldığında referans bacak yörüngelerini daha az hata ile başarılı bir şekilde takip edebilmektedir.

Şekil 6.32, parametreleri optimize edilen kontrolörler için gerçekleşen ile referans bacak konumları arasındaki yörünge takip hatalarını göstermektedir. Şekil 6.32’de görüldüğü üzere, PSO tabanlı bulanık FOPID kontrolörü, iyi bir kontrol performansı ile daha ufak hata sinyallerini üretebilmiştir. Şekil incelendiğinde, bacaklardaki en yüksek dinamik yörünge hataları, PID, FOPID, bulanık PID ve bulanık FOPID kontrolörleri için sırasıyla 0,1149 mm, 0,0192 mm, 0,0168 mm ve 0,0113 mm değerlerinde olduğu görülmektedir. Sonuç olarak, bulanık FOPID kontrolörü, tüm bacaklarda düşük hata değişim aralıkları ile daha üstün kontrol performansı göstermiştir. Bu deneyden elde edilen kontrol sinyallerinin RMS ve yörünge takip hatalarının MAE değerleri, Tablo 6.9’da verilmektedir.



Şekil 6.31. Deney 5'e göre kontrol altındaki bacakların yörünge takibi



Şekil 6.32. Deney 5'e göre kontrolörlerin yörünge takip hataları

Tabloda verilen MAE değerleri karşılaştırıldığında PSO tabanlı bulanik FOPID kontrolörü, diğer PSO tabanlı kontrolörlere göre daha başarılı sonuçlar verdiği görülmektedir. Bu deneyde de, PID kontrolörün MAE değerlerine bakıldığında, tüm eklemelerin kontrolünde aynı kararlı yapıyı sağlayamadığı anlaşılmaktadır.

Tablo 6.9. Deney 5'e göre kontrolörlerin performans karşılaştırması

Eklem	PID		FOPID		Bulanik PID		Bulanik FOPID	
	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}
Bacak 1	0.0128	0.4141	0.0118	0.4098	0.0078	0.4041	0.0049	0.4187
Bacak 2	0.0883	0.2273	0.0065	0.2288	0.0041	0.2203	0.0026	0.2140
Bacak 3	0.0371	0.4164	0.0119	0.4145	0.0079	0.4117	0.0049	0.4095
Bacak 4	0.0945	0.4030	0.0116	0.4146	0.0076	0.3948	0.0048	0.4006
Bacak 5	0.0125	0.2444	0.0070	0.2460	0.0044	0.2369	0.0028	0.2368
Bacak 6	0.0436	0.3970	0.0114	0.3976	0.0075	0.3873	0.0049	0.4013

Kontrolörlerin RMS değerlerine gelince, bacakların tümünde her bir kontrolör hemen hemen aynı kontrol sinyallerini üretmişlerdir. Sonuç olarak, hemen hemen aynı kontrol girişlerin üretildiği bu deneyde PSO tabanlı bulanık FOPID kontrolörün daha sağlam ve daha yüksek takip hassasiyetini başarabilmiştir.

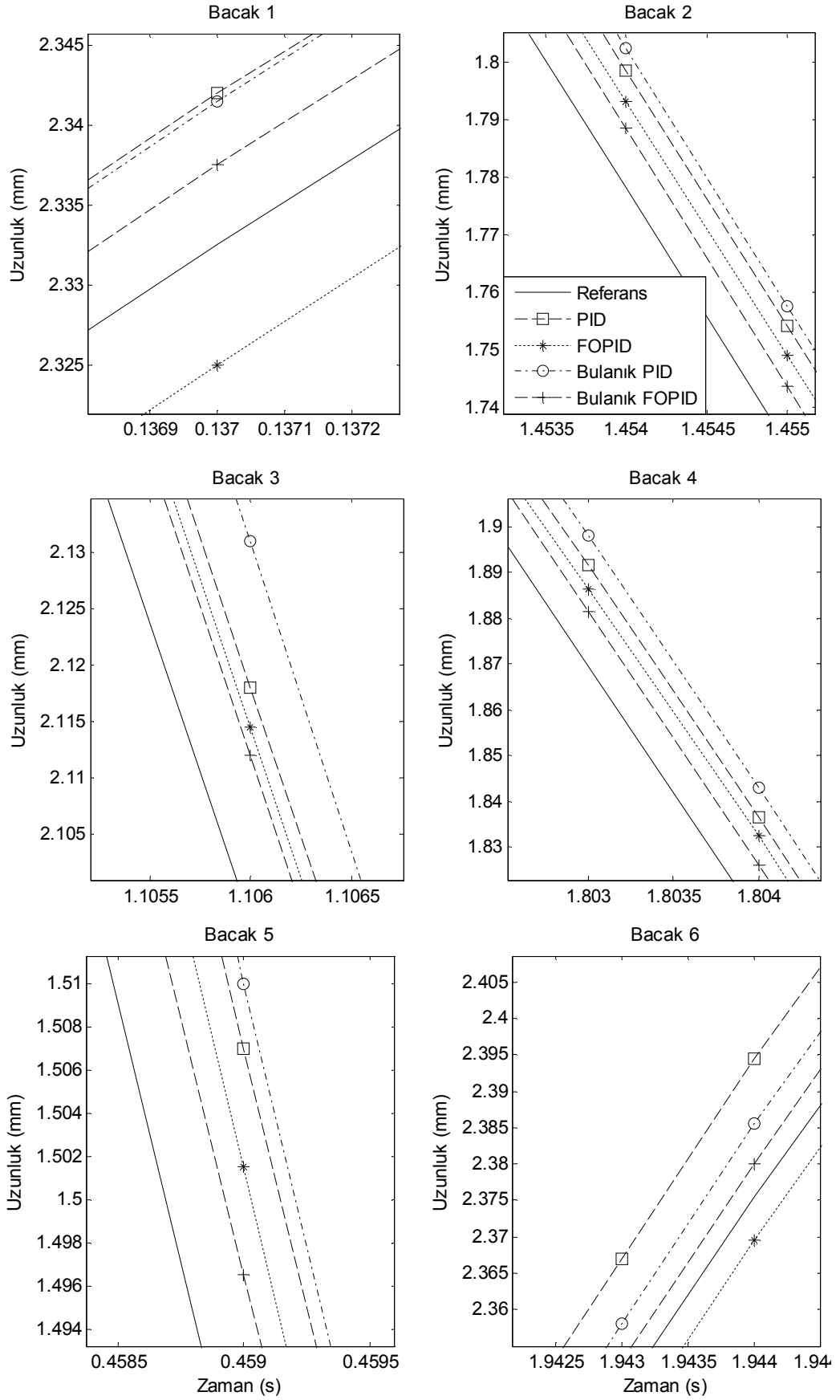
6.7.6. Yörünge takip deneyi 6

Bu deneyde, yüksek frekanslı hareketli platformun 6 serbestlik dereceli hareketi için yörünge denklemi aşağıda verildiği gibi seçilmiştir.

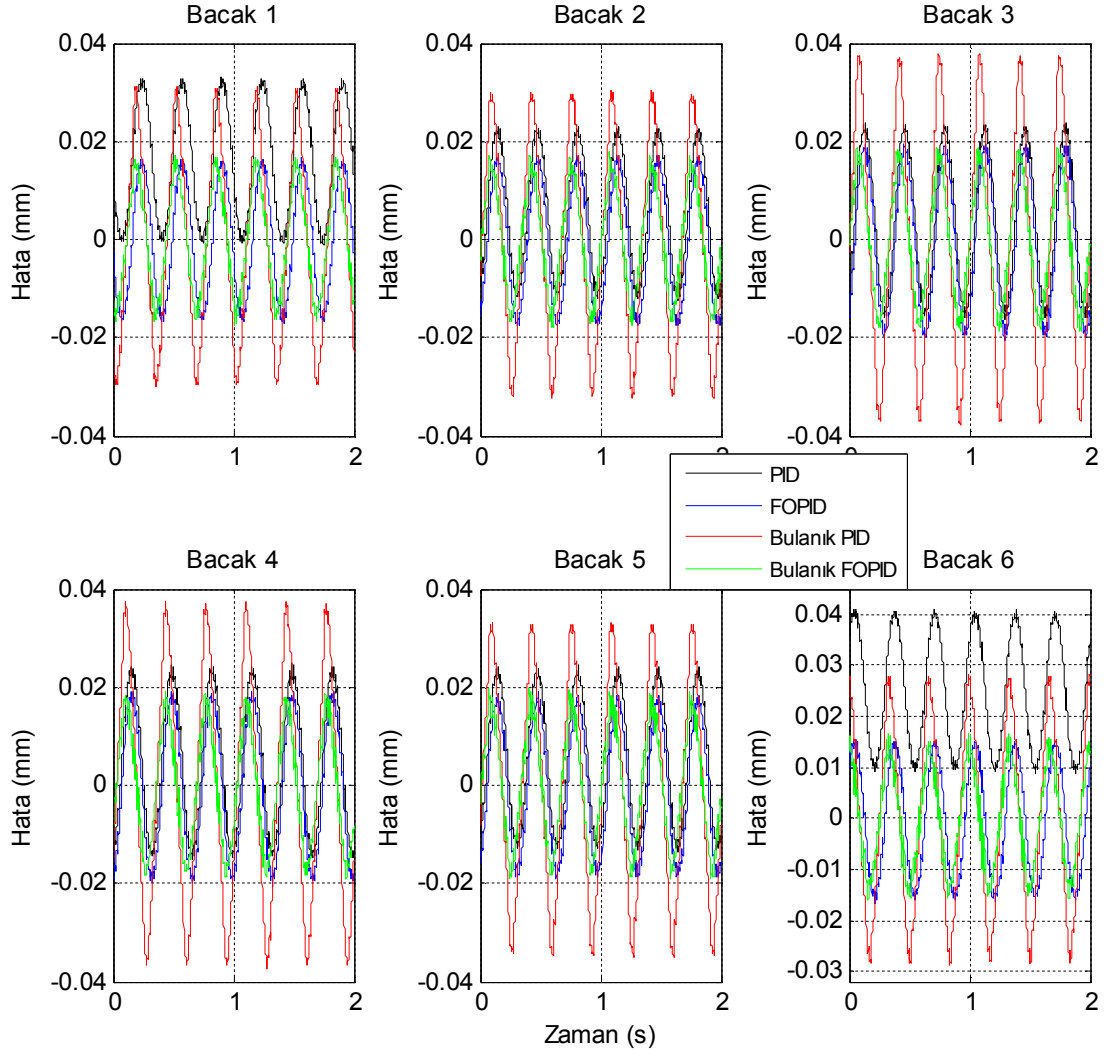
$$\begin{aligned}x(t) &= 5 \sin(2\pi 3t) & \alpha(t) &= \cos(2\pi 3t) \\y(t) &= 5 \cos(2\pi 3t) & \beta(t) &= \sin(2\pi 3t) \\z(t) &= 0.25 + 3 \sin(2\pi 3t) & \gamma(t) &= \cos(2\pi 3t)\end{aligned}\tag{6.14}$$

Şekil 6.33, verilen yörünge denklemine göre ters kinematik sonucu elde edilen referans bacak uzunlukları cevabında, PSO tabanlı PID, FOPID, bulanık PID ve bulanık FOPID kontrolörlerin bacak yörünge takip performansını göstermektedir. Şekil incelendiğinde bulanık FOPID kontrolörü, diğer kontrolörler ile karşılaştırıldığında daha iyi takip performansına sahip olduğu görülmektedir. Diğer taraftan, PSO tabanlı FOPID kontrolörün, PSO tabanlı PID ve bulanık PID kontrolörlerinden daha sağlam olduğu görülmektedir. Sonuç olarak, hem bulanık FOPID kontrolör hem de FOPID kontrolör, yüksek frekanslı bu yörünge için daha doğru yörünge takibini başarabilmiştir.

Yörünge takibi sırasında bacaklarda meydana gelen konum hataları, Şekil 6.34'te gösterilmektedir. Şekil 6.34 incelendiğinde bulanık FOPID kontrolörlü yörünge takip hatası $\pm 0,0195$ mm'den az olduğu, FOPID kontrolörlü yörünge takip hatası ise $\pm 0,0199$ mm'ye eriştiği görülmektedir. Bu sonuç, her iki kontrolöründe hem PID hem de bulanık PID kontrolörlerinden yüksek frekanslı yörüngelerde daha iyi performans sergilediğini göstermektedir. Bu deneyden elde edilen kontrol sinyallerin RMS ve yörünge takip hataların MAE değerleri Tablo 6.10'da verilmektedir.



Şekil 6.33. Deney 6'ya göre kontrol altındaki bacakların yörünge takibi



Şekil 6.34. Deney 6'ya göre kontrolörlerin yörünge takip hataları

Tablo ve grafiklerde verilen bu deneysel sonuçlar, bulanık FOPID kontrolörün, bacak yörünge takip kontrolünde hareket performansını geliştirilebildiğini göstermiştir. Diğer taraftan, Bulanık PID kontrolörün bu yüksek frekanslı yörünge deneyinde, takip performansının oldukça zayıfladığı gözlenmiştir.

Tablo 6.10. Deney 6'ya göre kontrolörlerin performans karşılaştırması

Eklem	PID		FOPID		Bulanık PID		Bulanık FOPID	
	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}
Bacak 1	0.0160	0.3625	0.0104	0.3601	0.0168	0.3532	0.0090	0.3686
Bacak 2	0.0114	0.3713	0.0107	0.3724	0.0174	0.3636	0.0092	0.3556
Bacak 3	0.0125	0.4212	0.0121	0.4220	0.0210	0.4207	0.0103	0.4149
Bacak 4	0.0125	0.4168	0.0120	0.4222	0.0204	0.4130	0.0103	0.4140
Bacak 5	0.0120	0.3951	0.0114	0.3953	0.0188	0.3874	0.0100	0.3862
Bacak 6	0.0250	0.3384	0.0098	0.3399	0.0156	0.3334	0.0084	0.3431

Tabloda verilen kontrolörlerin RMS değerleri incelendiğinde, en uygun kontrol girişlerini üretmelerine rağmen bulanık FOPID kontrolörü, düşük bacak konum hataları ile daha doğru yörünge takibini başarabilmiştir. Sonuç olarak, yüksek frekanslı yörünge deneyinde hem PSO tabanlı bulanık FOPID kontrolörün hem de PSO tabanlı FOPID kontrolörün, başarılı sonuçlar verdikleri gözlemlenmiştir.

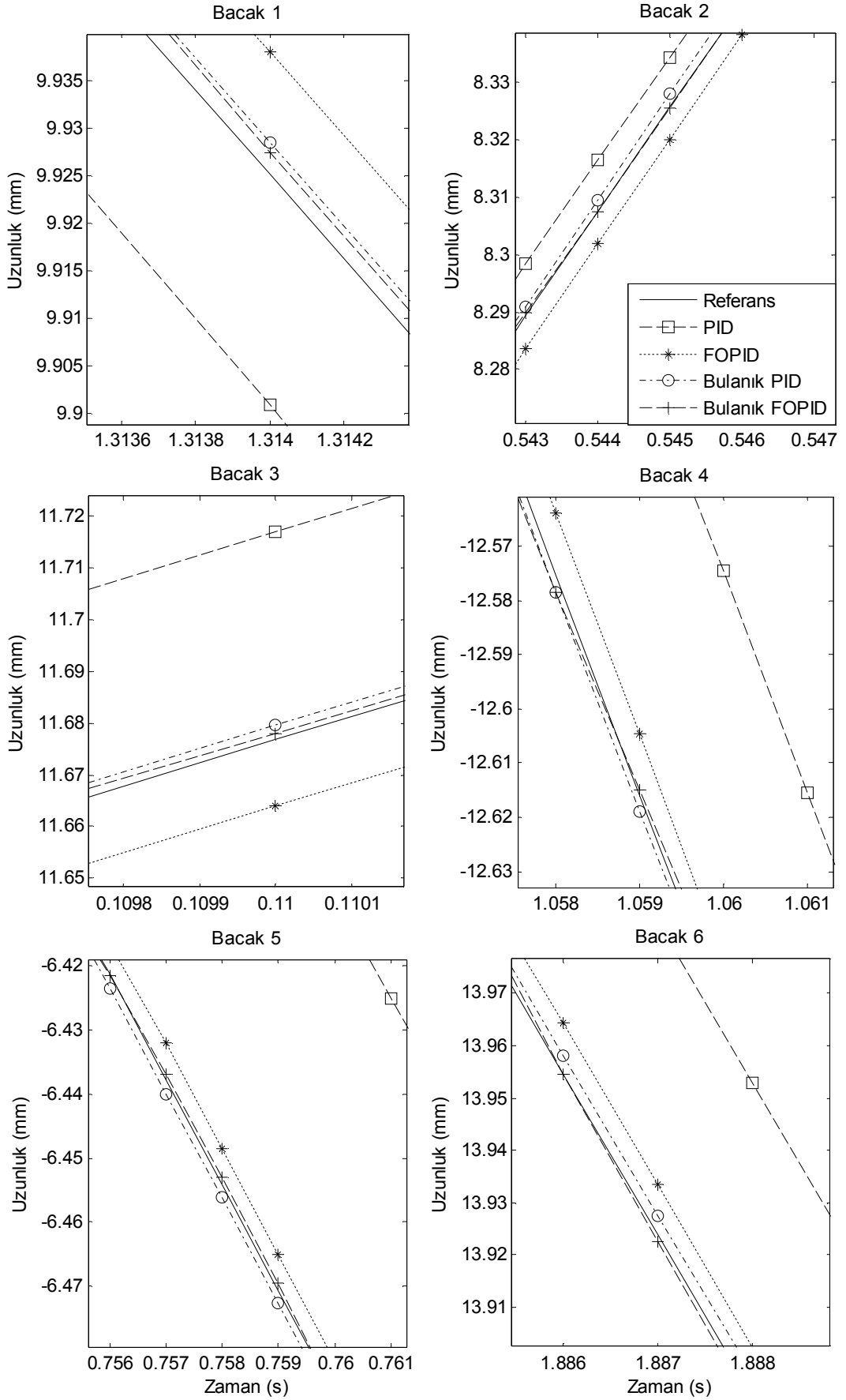
6.7.7. Yörünge takip deneyi 7

Bu deneyde, hareketli platformun 6 serbestlik dereceli hareketi ile yüksek bacak uzunluğunu verecek düşük frekanslı yörünge denklemi, aşağıda verildiği gibi seçilmiştir.

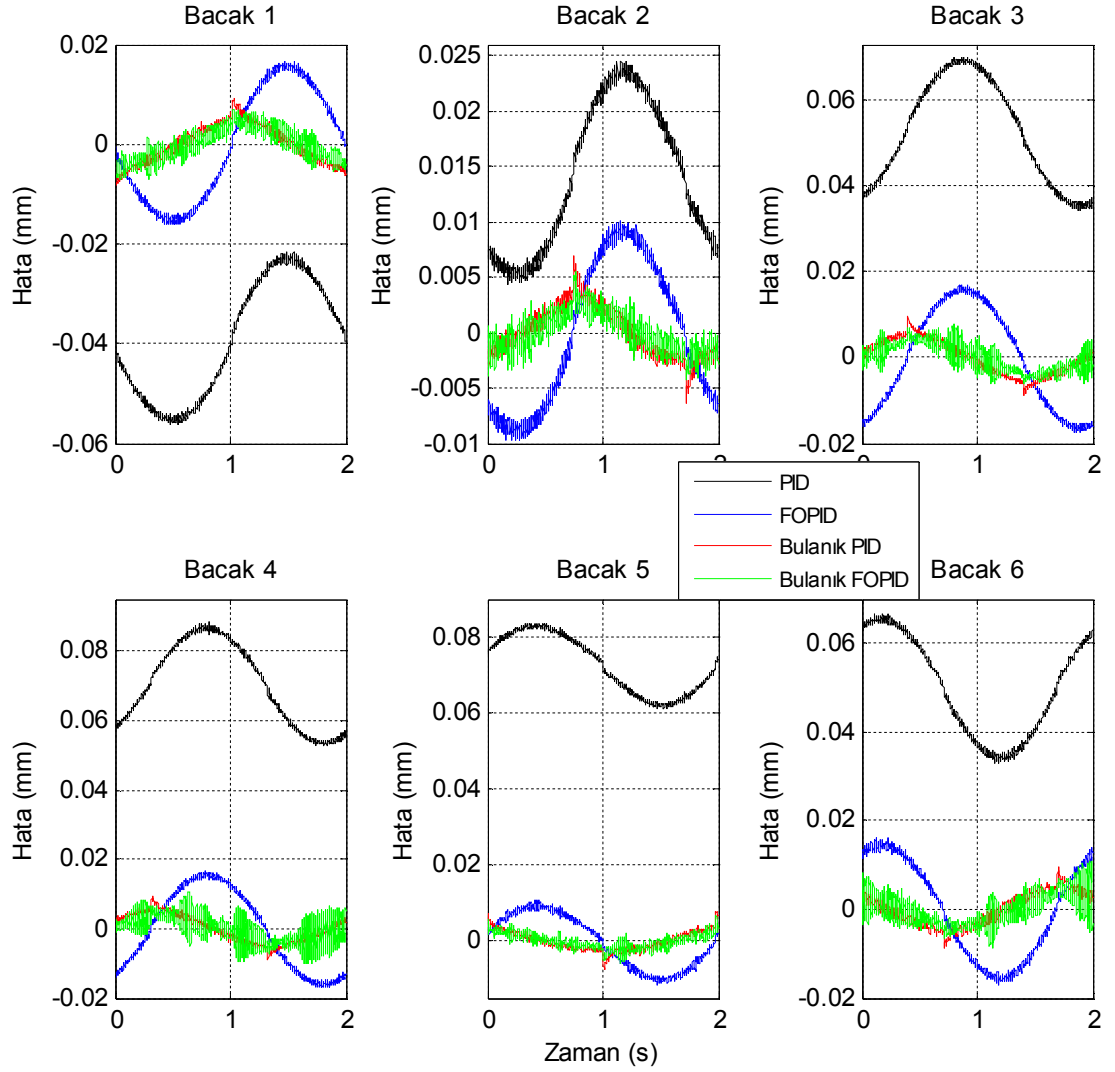
$$\begin{aligned}
 x(t) &= 10 \sin(2\pi 0.5t) & \alpha(t) &= 9 \cos(2\pi 0.5t) \\
 y(t) &= 10 \cos(2\pi 0.5t) & \beta(t) &= 9 \sin(2\pi 0.5t) \\
 z(t) &= 0.25 + \sin(2\pi 0.5t) & \gamma(t) &= 5 \cos(2\pi 0.5t)
 \end{aligned} \tag{6.15}$$

Şekil 6.35, izlenecek yörünge doğrultusunda belirlenen referans bacak uzunlukları ile PID, FOPID, bulanık PID ve bulanık FOPID kontrolörleri altındaki bacakların gerçekleştirdiği bacak uzunluklarını göstermektedir. Şekil incelendiğinde PSO tabanlı bulanık FOPID ve bulanık PID kontrolörlerin yörünge takibi, PSO tabanlı FOPID ve PID kontrolörlerine göre daha iyi olduğu görülmektedir. Ayrıca hem bulanık FOPID hemde bulanık PID kontrolörleri, platformun bacaklarını, referans yörüngeler doğrultusunda en az hata ile hareket ettirmişlerdir.

Yörünge takibi sırasında bacaklarda meydana gelen konum hataları, Şekil 6.36'da gösterilmektedir. Şekil gözlemlendiğinde, yüksek bacak uzunluklu bu yörünge deneyinde PSO tabanlı PID kontrolörün hata değişim aralığı, PSO tabanlı diğer kontrolörlerden daha yüksek olduğu görülmektedir. Bulanık FOPID ve bulanık PID kontrolörlerin hata değişim aralığı ise, daha azdır. Ayrıca, Şekil incelendiğinde, hem bulanık FOPID ve hem de bulanık PID kontrolörleri, $\pm 0,01$ mm hatalar ile referans yörüngelerini çok yakın takip ettikleri görülmektedir. Bu deneyden elde edilen kontrol sinyallerin RMS ve yörünge takip hataların MAE değerleri, Tablo 6.11'de verilmektedir.



Şekil 6.35. Deney 7'ye göre kontrol altındaki bacakların yörünge takibi



Şekil 6.36. Deney 7'ye göre kontrolörlerin yörünge takip hataları

Bulanık FOPID ve bulanık PID kontrolörlerin tabloda verilen MAE değerlerinin çok düşük olduğu görülmektedir. Bu sonuç, hem bulanık FOPID hem de bulanık PID kontrolörlerin daha az hatalı kontrol gerçekleştirdiğini göstermektedir.

Tablo 6.11. Deney 7'ye göre kontrolörlerin performans karşılaştırması

Eklem	PID		FOPID		Bulanık PID		Bulanık FOPID	
	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}	MAE	U_{RMS}
Bacak 1	0.0395	0.3604	0.0102	0.3557	0.0034	0.3481	0.0028	0.3667
Bacak 2	0.0140	0.2080	0.0060	0.2105	0.0020	0.2015	0.0015	0.1962
Bacak 3	0.0526	0.3711	0.0106	0.3707	0.0036	0.3661	0.0030	0.3688
Bacak 4	0.0703	0.3596	0.0103	0.3708	0.0035	0.3533	0.0035	0.3673
Bacak 5	0.0728	0.2234	0.0064	0.2259	0.0021	0.2163	0.0017	0.2164
Bacak 6	0.0500	0.3432	0.0099	0.3448	0.0033	0.3355	0.0030	0.3523

Tabloda verilen kontrolörlerin RMS değerleri karşılaştırıldığında, tüm bacaklarda hemen hemen aynı kontrol sinyallerinin üretildiği anlaşılmaktadır. Sonuç olarak, birbirine yakın üretilen kontrol sinyalleri altında, PSO tabanlı bulanık FOPID kontrolörü, referans yörüngelerini oldukça yakın ve başarılı bir şekilde takip edebilmiştir.

Yapılan 7 deney için PSO tabanlı PID, FOPID, bulanık PID ve bulanık FOPID kontrolörlerin toplam yörünge takip hataları (Denklem (6.2)) ve bacaklarda üretilen kontrol sinyallerin RMS değerlerin ortalaması (U_{MRMS}), Tablo 6.12’de verilmektedir.

Tablo 6.12. Tüm deneyler için kontrolörlerin toplam yörünge takip hataları ve kontrol sinyallerin ortalama RMS değerleri

DENEYLER	PID		FOPID		Bulanık PID		Bulanık FOPID	
	E_{MRSE}	U_{MRMS}	E_{MRSE}	U_{MRMS}	E_{MRSE}	U_{MRMS}	E_{MRSE}	U_{MRMS}
DENEY 1	0.1045	0.3243	0.0260	0.3256	0.0176	0.3141	0.0118	0.3238
DENEY 2	0.1146	0.3119	0.0245	0.3122	0.0163	0.3029	0.0109	0.3077
DENEY 3	0.1337	0.2845	0.0199	0.2850	0.0129	0.2769	0.0092	0.2814
DENEY 4	0.2101	0.4226	0.0296	0.4236	0.0212	0.4131	0.0139	0.4188
DENEY 5	0.1445	0.3504	0.0277	0.3519	0.0188	0.3425	0.0120	0.3468
DENEY 6	0.0433	0.3842	0.0296	0.3853	0.0507	0.3786	0.0262	0.3804
DENEY 7	0.1328	0.3110	0.0244	0.3131	0.0086	0.3035	0.0075	0.3113

Tabloda verilen kontrolörlerin E_{MRSE} (Denklem (6.2)) değerlerine bakıldığında, bulanık FOPID kontrolörün daha üstün sağlamlıkla tüm deneylerde yüksek kontrol performansını başarabildiği görülmektedir.

6.8. Sonuç

Bu bölümde, 6 serbestlik dereceli Stewart Platform manipülatörün bacak yörünge takip kontrolünde klasik PID, kesir dereceli PID (FOPID), bulanık PID ve bulanık FOPID kontrol yapıları tasarlanmıştır. Bu kontrol yapılarına ait parametrelerin en uygun değerleri, üst platformun tüm eksenlerde hareketini sağlayan yörünge denklemi verilerek parçacık sürü optimizasyon (PSO) algoritması ile ayarlanmıştır. Parametreleri optimize edilen kontrolörler ile deneysel Stewart Platform manipülatörü üzerinde 7 yörünge deneyi gerçekleştirilmiştir. Ayrıca, deneyler

sırasında alınan veriler kullanılarak PSO tabanlı bu kontrolörlerin kontrol performansları, grafik ve tablolar şeklinde karşılaştırılmıştır.

Yapılan deneysel sonuçlar incelendiğinde, üyelik fonksiyon parametreleri, türev ve integratörün dereceleri (μ, λ) , giriş ve çıkış kazanç katsayıları ayarlanan bulanık FOPID kontrolörü, $K_p, K_d, K_i, \lambda, \mu$ parametreleri ayarlanan FOPID, K_p, K_d, K_i parametreleri ayarlanan PID ve de yapısal ve yapısal olmayan parametreleri ayarlanan bulanık PID kontrolörlerinden daha başarılı olduğu gözlemlenmiştir. Ayrıca, tablolarda verilen MAE ve RMS değerlerine bakıldığında, hemen hemen aynı üretilen kontrol sinyallerinde PSO tabanlı bulanık FOPID kontrolörün kontrol performansının daha yüksek olduğu görülmektedir. Verilen deneysel grafikler incelendiğinde, PSO tabanlı bulanık FOPID kontrolörü altındaki bacakların verilen referans yörüngelerini başarılı bir şekilde takip ettiği görülmüştür. Diğer taraftan, yüksek frekanslı yörünge deneyinde bulanık PID kontrolörü düşük kontrol performansı gösterirken diğer kontrolörler ise, daha kararlı bir kontrol performansı sergilemiştir.

Sonuç olarak, Stewart Platform yapısının elde edilen ters kinematik ve dinamik modeli kullanılarak PSO tabanlı 4 farklı kontrol yapısı tasarlanmıştır. Deneysel çalışmalar incelendiğinde PSO yöntemi ile parametreleri en iyilenen bulanık FOPID kontrolörün, hareketli platformun belirlenen yörüngeleri için bacak uzayında daha iyi performans vermesi ve bacakları referans yörüngeler doğrultusunda en az hata ile takip ettirebilmesinden dolayı daha etkili bir kontrol yapısı olarak ortaya çıkmıştır.

7. SONUÇLAR VE ÖNERİLER

7.1. Giriş

Bir makine yapısı olarak Stewart Platform mekanizmalarına olan yakın ilgi, bir dizi araştırma konularını ön plana çıkarmaktadır. Bu tezde yapılan çalışmanın ilgi alanları, Stewart Platform'un ters kinematik modelini, dinamik modelini, simülasyonu ve de kontrolör tasarımları ve değerlendirilmesini kapsamaktadır.

Paralel bağlı bir manipülatör olarak Stewart Platform'u robotik bir mekanizmadır. Stewart Platform manipülatörlerin kinematik ve dinamik davranışını modellemek için paralel ve seri bağlı robot manipülatörleri arasındaki benzerlikler göz önünde bulundurularak, seri bağlı manipülatörlerde kullanılan çözümsel yöntemlere başvurulmaktadır.

Stewart Platform manipülatörleri, yüksek sağlamlık ve hassas konumlandırma kabiliyetini gösterebilen makine yapılarına benzerdirler. Geleneksel makine yapılarında hareket kontrol sistemlerin performansını geliştirmek için günümüzde çok çeşitli kontrol algoritmaları kullanılmakta ve gün geçtikçe bu algoritmalar geliştirilmektedir. Ayrıca uygulama başarıları artan ve sürekli geliştirilmeye çalışılan evrimsel algoritmalar, en iyi kontrol parametrelerini ayarlama kullanılmaktadır. Bu çalışmada, makine yapısına dayanan Stewart Platform manipülatörün hareket kontrolü için bu kontrol algoritmalarının geliştirilmesi ve uygulanabilirliği araştırılmıştır.

Bu tezde ele alınan konular ve sonuçlar bir sonraki alt bölümde detaylı olarak verilecektir. Ayrıca, yeni veya benzer bir konuda yapılacak çalışmalara temel oluşturmak için bazı önerilerde bulunulacaktır.

7.2. Sonular

Stewart Platform manipölatöründe kinematik problem, hareketli platformun kinematik deęişkenleri ile prizmatik eklemlerin kinematik deęişkenleri arasındaki ilişki ile ilgilendir. İleri kinematik analizde problem, bacak uzunlukları verildiğinde hareketli platformun aęırlık merkezinin konumunu (ötelenme ve yönelme) bulmaktır. Ters kinematik analizde problem ise hareketli platformun aęırlık merkezinin konumu verildiğinde bacak uzunluklarının bulunmasıdır. Paralel bir manipölatör olan Stewart Platform'un ileri kinematik analizi, daha karmaşık ve beraberinde işlem yükü ve çözümünün tek olmadığı gibi sıkıntılı durumları getirmektedir. Bundan dolayı, bu çalışmada, seri manipölatörlere göre daha kolay olan ters kinematik problemi ele alınmıştır. Sonuç olarak, burada kullanılan deneysel Stewart Platform manipölatöründe referans bacak uzunluklarını elde edebilmek için hareketli platformun konumu ve yönelimi verildiğinde bu bacak uzunlukların ne olacağını belirleyen ters kinematik problem çözümü yapılmıştır.

Bir paralel manipölatöre ait konum, hız, ivme analizi denklemlerinin incelenmesinde, herhangi bir 6 serbestlik dereceli manipölatörün dinamik denklemlerin nasıl elde edileceęi ve manipölatöre ait dinamik denklemler çıkarılırken izlenen yol ve ne gibi yaklaşımların yapıldığı detaylı olarak anlatılmıştır. Verilen literatür çalışmaları ışığı altında, sistemin kinetik ve potansiyel enerjilerinden yola çıkılarak hız ve ivme karakteristiklerin bulunabildięi ve bu deęişkenlerle de sistemin dięer dinamik deęişkenlerinin hesaplanabildięi Lagrange formölasyonu, çalışmada kullanılan Stewart Platform'un katı gövde dinamik denklemlerinin çıkarılmasında kullanılmıştır.

Lagrange formölasyonu kullanılarak önce hareketli platformun dinamięi sonra da bacakların dinamięi çıkarılmıştır. Bu formölasyonda kullanılan Jacobian matrislerinin çıkarımı için iki farklı yöntem kullanılmıştır. Dolayısıyla iki farklı Jacobian matrisi çıkarıldığından, bacaklara ait iki farklı şekilde kütle matrisi, Coriolis ve merkezkaç matrisi ve son olarak yerçekimi kuvvet vektörü elde edilmiştir. Hareketli platformun ve bacakların dinamik denklemlerinde yer alan matris ve

vektörler birleştirilerek iki farklı şekilde katı gövde dinamik denklemleri elde edilmiştir. Son olarak, bacaklarda bulunan eyleyicilerin elektriksel ve mekanik dinamikleri çıkarılarak elde edilen katı gövde dinamiğine eklenmiştir. Böylece, Stewart Platform manipülatörün tüm sistem dinamiği iki farklı (SIM-A ve SIM-B) şekilde çıkarılması gerçekleştirilmiştir.

Elde edilen iki farklı dinamik modelin davranışlarını gözlemlemek amacıyla 7 farklı yörünge denklemini kullanarak doğrulama deneyleri yapılmıştır. Deneysel ve simülasyon sonuçları karşılaştırıldığında, dinamik modellerin (SIM-A ve SIM-B), deneysel robotun gerçekleştirdiği bacak yörüngelerini başarılı bir şekilde takip edebildiği görülmüştür. Ayrıca, tablolarda verilen maliyet değerlerine bakıldığında, dinamik modellerin gerçekleştirdiği hataların, deneysel robotun gerçekleştirdiği hatalara yakın olduğu görülmektedir. Bu sonuçlar, önerilen yöntemle elde edilen dinamik modellerin yüksek doğruluğa sahip olduğunu göstermektedir. Ayrıca, SIM-A ve SIM-B dinamik modelleri, tüm doğrulama deneylerinde birbirine yakın yörünge takip hataları yaparak hemen hemen benzer davranışlar sergilemişlerdir. Sonuç olarak, Stewart Platform yapısının genel bir sınıfı için ters kinematik ve dinamik model elde edilebilmiştir.

Stewart Platform manipülatörlerinde genel olarak, eklem uzayı kontrolü ve kartezyen (görev veya çalışma) uzayı kontrolü olmak üzere iki kontrol yapısı kullanılır. Bu çalışmada elde edilen ters kinematik ve dinamik model kullanılarak eklem uzayında kontrol yapıları tasarlanır. Tasarlanan bu kontrol yapıları, bacaklara uygulanacak gerilim değerlerini üretirler ve bu değerler dinamik modele iletilir. Dinamik modelden bacaklara ait geri besleme konum bilgileri alınarak bir sonraki kontrol adımında kullanılmak üzere referans konum bilgisi ile karşılaştırılır ve tekrar kontrolörlere giriş bilgisi olarak alınır. Bu işlem verilen simülasyon süresi boyunca tekrarlanır.

Bu tezde, deneysel Stewart Platform manipülatörün eklem esaslı referans yörünge takip kontrolünde klasik PID, kesir dereceli PID (FOPID), bulanık PID ve bulanık FOPID olmak üzere toplam 4 kontrol yapısı tek giriş tek çıkış olarak tasarlanmıştır. Her bir kontrol yapısı için bacaklar birbirinden bağımsız olarak düşünülmüş ve her

bir bacak için yapı olarak aynı, çalışma olarak birbirlerinden bağımsız altı adet kontrolör tasarlanmıştır.

Deneysel Stewart Platform manipülatörü için kontrolörler tasarlanırken elde edilen kinematik ve dinamik model sayesinde parçacık sürü optimizasyon (PSO) algoritması kullanılarak bu kontrolörlere ait parametrelerin en iyi değerleri bulunmuştur. Deneysel robota ait dinamik denklemlerin doğru modellenmesi ve bu denklemlerin doğrusal olmayan etkilerini azaltacak PID, FOPID, bulanık PID ve bulanık FOPID kontrol algoritmalarının optimize edilmesi, bacakların yörünge takip kontrolünde başarılı olmasında büyük rol oynamıştır.

Parametreleri optimize edilen kontrolörler ile deneysel Stewart Platform manipülatörü üzerinde 7 farklı yörünge deneyi gerçekleştirilmiştir. Ayrıca, deneyler sırasında alınan veriler kullanılarak PSO tabanlı bu kontrolörlerin kontrol performansları, bacakların gerçekleştirdiği konum ve konum hataları üzerinden hem grafiksel hem de sayısal sonuçlarla karşılaştırılmıştır.

Yapılan deneysel sonuçlar incelendiğinde, PSO tabanlı bulanık FOPID kontrolörünün, PSO tabanlı PID, FOPID ve bulanık PID kontrolörlerinden daha başarılı olduğu gözlemlenmiştir. Tüm deneylerde bacakların konum grafikleri incelendiğinde, bulanık FOPID kontrol sistemi altındaki bacakların gerçekleştirdiği yörüngeler, verilen referans yörüngelerini oldukça yakın takip etmiştir.

Tüm deneylerde verilen MAE ve RMS değerlerine bakıldığında, kontrolörler hemen hemen aynı kontrol sinyallerini üretmelerine rağmen, PSO tabanlı bulanık FOPID kontrolörün kontrol performansının daha yüksek olduğu görülmüştür. Kontrolörlerin uyarlanabilir özelliklerinin sınanması için birbirinden farklı yörünge deneyleri yapılmıştır. Yüksek frekanslı yörünge seçilerek yapılan deneyde, bulanık PID kontrolörü düşük kontrol performansı gösterirken diğer kontrolörler ise, daha kararlı bir kontrol performansı sergilemiştir. Bulanık FOPID kontrol düzeneği ile deneysel robot manipülatörün yüksek frekanslı bu yörünge deneyinde, yüksek doğrulukla yörünge takibi yapabildiği grafiksel olarak gösterilmiştir. Bu yüksek frekanslı

yörünge deneyinde PID ve bulanık PID kontrolörlerine kıyasla, hem bulanık FOPID kontrolörün hem de FOPID kontrolörün, başarılı sonuçlar verdikleri gözlemlenmiştir.

Değişik hareket yörüngeleri için önerilen bulanık FOPID kontrol yapısının yörünge takibi performansı ve yörünge takibi doğruluğunda sağladığı iyileştirme grafiksel ve sayısal olarak gösterilmiştir. Önerilen kontrol düzeneği ile deneysel Stewart Platform manipülatörün oldukça iyi doğrulukla yörünge takibi yapabildiği gözlemlenmiştir. Stewart manipülatör sisteminin eklem esaslı kontrol düzeneğinde önerilen kontrol yapısı ile diğer kontrol yapılarında ortaya çıkan konum hataları etkin bir şekilde giderilmiş, bu sayede manipülatör sistemi iyi bir kararlılık ve dayanıklılık göstermiştir.

Sonuç olarak; Stewart Platform yapısının elde edilen ters kinematik ve dinamik modeli kullanılarak PSO tabanlı 4 farklı kontrol yapısı tasarlanmıştır. Deneysel çalışmalar incelendiğinde PSO algoritması ile parametreleri en iyilenen bulanık FOPID kontrolörü, hareketli platformun belirlenen yörüngeleri için bacak uzayında daha iyi performans vermesi ve bacakları referans yörüngeler doğrultusunda en az hata ile takip ettirebilmesinden dolayı daha etkili bir kontrol yapısı olarak ortaya çıkmıştır. Ayrıca, takibi zor yörüngelerin seçilmesine rağmen bacakların çok küçük hatalarla referans yörüngelerini gayet yakından takip etmesi, hem PSO tabanlı bulanık FOPID kontrol yapısının hem de bu tez çalışmasında önerilen kinematik ve dinamik modelin kullanımının bir başarısı olarak ortaya çıkmaktadır.

7.3. Öneriler

Stewart Platform manipülatörü gibi böyle karmaşık dinamik sistemlerin MATLAB/Simulink ortamında gerçek zamanlı ve etkileşimli blok uygulamaları şeklinde simüle etmek için, işlemci yeteneğine dayanan işlem hızı yüksek bilgisayarlar kullanılabilir. Ayrıca, paralel işlem yeteneği artırılmış işlemciler kullanılarak hesapsal yük dağıtılabilir.

Stewart Platform manipülatörün bacak ataletlerini ve sürtünme etkilerini dinamik modele katarak en genel durumda simülasyon çalışmaların yapılması sağlanabilir.

Stewart Platform manipülatörün eklem yörünge kontrolü için geliştirilen ve bulanık FOPID kontrol yapısının iyileştirildiği bu çalışmalar, referans yörünge için belirlendiği diğer kontrol uygulamaları için de geliştirilebilir. Genetik algoritma, karınca kolonisi, yapay arı kolonisi, diferansiyel gelişim algoritması gibi farklı optimizasyon teknikleri kullanılarak burada tasarlanan kontrolörlerin parametreleri ayarlanabilir. Bu algoritmaların Stewart Platform manipülatörün yörünge takip kontrolünde performansları araştırılabilir. Ayrıca bu konuda yapılacak çalışmalarda çeşitli bozucu etkilerinde hesaplama dahil edilmesi ve kullanılan kontrol algoritmaların değişik bozucular etkisindeki performansı da gözlemlenebilir. Stewart Platform manipülatörün yörünge kontrolü için yapılan optimizasyon çalışmasında kullanılan program algoritması, matris işlemlerinin kolay olması nedeniyle MATLAB' de yazılıp Simulink modeli oluşturulmuştur. Ancak örnekleme zamanı 1 milisaniye olduğundan programın çalışma hızı önemli ölçüde azalmakta ve dinamik hesaplamaların ağır oluşundan dolayı da optimizasyon süreci uzun sürmektedir. İleride bu algoritma üzerinde yapılabilecek değişiklikler ile veya farklı algoritmalar kullanılarak bu süre kısaltılabilir.

KAYNAKLAR

- [1] Stewart D., A Platform with Six Degrees of Freedom, *Proceedings of the Institution of Mechanical Engineers*, 1965, **180**, 371-386.
- [2] Tsai L. W., *Robot Analysis: The Mechanism of Serial and Parallel Manipulators*, John Wiley & Sons, Inc., 1999.
- [3] Gough V. E., Whitehall S. G., Universal Tire Test Machine, *In Proceedings of the International Technical Congress F.I.S.I.T.A.*, London, May 1962.
- [4] Wang Y., Symbolic Kinematics and Dynamics Analysis and Control of a General Stewart Parallel Manipulator, Master of Science, Faculty of the Graduate School of the State University of New York, Buffalo, 2008.
- [5] Raghavan M., The Stewart platform of general geometry and has 40 configurations, *Journal of Mechanical Design*, 1993, **115**, 277-282.
- [6] Goldenberg A. A., Apkarian J. A., Smith H. W., An Approach to Adaptive Control of Robot Manipulators Using the Computed Torque Technique, *Journal of Dynamic Systems, Measurement and Control*, 1989, **111**, 1-8.
- [7] Klir G. J., Fuzzy logic, *Potentials, IEEE* , 1995, **14**, 10-15.
- [8] Concepcion A. M., Blas M. V., Vicente F., YangQuan C., Tuning and auto-tuning of fractional order controllers for industry applications, *Control Engineering Practise*, 2008, **16**, 798-812.
- [9] Åström K. J., Hägglund T., *PID Controllers: Theory, Design, and Tuning*, 2nd Edition, Instrument Society of America, USA, 1995.
- [10] Caponetto R., Dongola G., Fortuna L., Petras I., *Fractional Order Systems: Modeling and Control Applications*, World Scientific, Singapore, 2010.
- [11] Vinagre B. M., Monje C. A., Calderon A. J., Suarez J. I., Fractional PID Controllers for Industry Application. A Brief Introduction, *Journal of Vibration and Control*, 2007, **13**, 1419-1429.
- [12] Calderon A. J., Vinagrea B. M., Feliu V., Fractional order control strategies for power electronic buck converters, *Signal Processing*, 2006, **86**, 2803-2819.
- [13] Schlegel M., Cech M., Fractal system identification for robust control - the moment approach, *in Proceedings of ICCS 2004 Conference*, Zakopane, Poland, 1-6 August 2004.

- [14]Fonseca Ferreira N. M., Tenreiro Machado J. A., Fractional-Order Hybrid Control of Robotic Manipulators, *In 11th International Conference on Advanced Robotics*, Coimbra, Portugal, 30 June - 3 July 2003.
- [15]Feliu-Batlle V., Rivas Pe´rez R., Sanchez Rodriguez R., Fractional robust control of main irrigation canals with variable dynamic parameters, *Control Engineering Practice*, 2007, **15**, 673-686.
- [16]Fan H., Sun Y., Zhang X., Research on Fractional Order Controller in Servo Press Control System, *Proc. of the 2007 IEEE Int. Conf. on Mechatronics and Automation*, Harbin, China, 5-7 August 2007.
- [17]Chengbin Ma., Hori Y., The application of fractional order PID controller for robust two-inertia speed control, *In Proceedings of the 4th International Power Electronics and Motion Control Conference*, Xi’an, China, 14-16 August 2004.
- [18]Isabel S. J., Tenreiro Machado J. A., Fractional control of heat diffusion systems, *Nonlinear Dynamics*, 2008, **54**, 263-282.
- [19]Machado T., Da Graca Marcos D., Duarte F., Fractional Dynamics in the Trajectory Control of Redundant Manipulators, *Communications in Nonlinear Science and Numerical Simulation*, 2008, **13**, 1836-1844.
- [20]Bohannon G. W., Analog Fractional Order Controller in a Temperature Control Application, *in Proc. IFAC Workshop on Fractional Differentiation and its Application*, Porto, Portugal, 19-21 July 2006.
- [21]Podlubny I., Petras I., Vinagre B. M., O’Leary P., Dorcak L., Analogue Realization of Fractional Order Controller, *Nonlinear Dyn.*, 2002, **29**, 281-296.
- [22]Silva M. F., Tenreiro Machado J. A., Fractional order PD Joint Control of Legged Robots, *Journal of Vibration and Control*, 2006, **12**, 1483-1501.
- [23]Podlubny I., Fractional-Order Systems and $PI^\lambda D^\mu$ Controllers, *IEEE Transactions on Automatic Control*, 1999, **44**, 208-214.
- [24]Vinagre B. M., Podlubny I., Dorcak I., Feliu V., On Fractional PID Controllers: A Frequency Domain Approach, *IFAC workshop on Past, present and future of PID control*, Terrasa, Spain, 5-7 April 2000.
- [25]Petras I., Hysiusova M., Design of Fractional Order Controllers Via H_∞ Norm Minimization, *Selected Topics in Modeling and Control*, 2002, **3**, 50-54.
- [26]Chen Y. Q., Moore K. L., Discretization Schemes for Fractional Order Differentiators and Integrators, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 2002, **49**, 363-367.

- [27]Chen Y. Q., Vinagre B. M., Podlubny I., Continued Fraction Expansion Approaches to Discretizing Fractional Order Derivatives - An Expository Review, *Nonlinear Dynamics*, 2004, **38**, 155-170.
- [28]Petras I., Fractional-Order Feedback Control of A DC Motor, *Journal of Electrical Engineering*, 2009, **60**, 117-128.
- [29]Xue D., Zhao C., Chen Y. Q., Fractional Order PID Control of A DC-Motor with Elastic Shaft: A Case Study, *Proceedings of the 2006 American Control Conference Minneapolis*, Minnesota, USA, 14-16 June 2006.
- [30]Chang L. Y., Chen H. C., Tuning of fractional PID Controllers Using Adaptive Genetic Algorithm for Active Magnetic Bearing System, *WSEAS transactions on Systems*, 2009, **8**, 226-236.
- [31]Aldair A. A., Wang W. J., Design of Fractional Order Controller Based on Evolutionary Algorithm for a Full Vehicle Nonlinear Active Suspension Systems, *International Journal of Control and Automation*, 2010, **3**, 33-46.
- [32]Lee C. H., Chang F. K., Fractional-order PID Controller optimization via improved Electromagnetism-Like algorithm, *Expert Systems with Applications*, 2010, **37**, 8871-8878.
- [33]Arijit B., Swagatam D., Ajith A., Sambarta D., Design of Fractional-Order Controllers with an Improved Differential Evolution, *Engineering Applications of Artificial Intelligence*, 2009, **22**, 343-350.
- [34]Cao J. Y., Cao B. G., Design of Fractional Order Controller Based on Particle Swarm Optimization, *International Journal of Control, Automation, and Systems*, 2006, **4**, 775-781.
- [35]Maiti D., Sagnik B., Konar K., Design of a Fractional Order PID Controller Using Particle Swarm Optimization Technique, *2. National Conference on Recent Trends In Information Systems*, Kolkata, India, 7-9 February 2008.
- [36]Majid Z., Masoud K. G., Nasser S., Mostafa P., Design of A Fractional Order PID Controller For An AVR Using Particle Swarm Optimization, *Control Engineering Practice*, 2009, **17**, 1380-1387.
- [37]Oustaloup A., *La Commande CRONE: Commande Robuste d'Ordre Non Entier*, Hermes, Paris, 1991.
- [38]Oustaloup A., Bansard M., First Generation CRONE control, in *Proc. International Conference on Systems, Man and Cybernetics*, LeTouquet, France, 17-20 October 1993.
- [39]Oustaloup A., Lanusse P., Mathieu B., Second generation CRONE control, in *Proc. International Conference on Systems, Man and Cybernetics*, LeTouquet, France, 17-20 October 1993.

- [40] Oustaloup A., Lanusse P., Mathieu B., Third generation CRONE control, in *Proc. International Conference on Systems, Man and Cybernetics*, LeTouquet, France, 17-20 October 1993.
- [41] Valerio D., Ninteger v. 2.3 Fractional Control Toolbox for MATLAB, <http://web.ist.utl.pt/duarte.valerio/ninteger/ninteger.html> (Ziyaret Tarihi: 10 Kasım 2011).
- [42] Şenol B., Kesir Dereceli Kontrol Sistemlerinin Frekans Cevaplarının Analizi için Matlab Ortamında Toolbox Geliştirilmesi, Yüksek Lisans Tezi, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Malatya, 2011, 300978.
- [43] Oldham K. B., Spanier J., *The Fractional Calculus*, Academic Press, New York and London, 1974.
- [44] Podlubny I., *Fractional Differential Equations*, 198 of Mathematics in Science and Engineering, Academic Press, San Diego, Calif, USA, 1999.
- [45] Oldham K. B., Spanier J., *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*, Dover Books on Mathematics, New York, 2006.
- [46] Zadeh L. A., Fuzzy Sets, *Information and Control*, 1965, **8**, 338-353.
- [47] Üstoğlu İ., Bulanık Kontrol Sistemlerinde Kararlılık, Yüksek Lisans Tezi, İ.T.Ü., Fen Bilimleri Enstitüsü, İstanbul, 1999, 100579.
- [48] Mamdani E. H., Assilian S., An experiment in linguistic synthesis with a fuzzy logic controller, *Int. J. Man-Machine Studies*, 1975, **7**, 1-13.
- [49] Mac Vicar-Whelan P. J., Fuzzy sets for man-machine interaction, *Int. J. Man-Machine Studies*, 1976, **8**, 687-697.
- [50] Passino K. M., Yurkovich S., *Fuzzy Control*, Addison-Wesley Longman Inc., California, 1998.
- [51] Sugeno M., *Industrial Applications of Fuzzy Control*, Elsevier Science Inc., Amsterdam, 1985.
- [52] Wang Li-Xin, *A Course in Fuzzy Systems and Control*, Prentice Hall, USA, 1997.
- [53] Hao Ying, *Fuzzy Control and Modeling: Analytical Foundations and Applications*, Wiley-IEEE Press, 2000.
- [54] Yen J., Langari R., *Fuzzy Logic: Intelligence, Control and Information*, Prentice Hall, New Jersey, 1999.

- [55] Tanaka K., Wang H. O., *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, John Wiley & Sons, New York, 2001.
- [56] Mohan B. M., Patel A. V., Analytical structures and analysis of the simplest fuzzy PD controllers, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 2002, **32**, 239-248.
- [57] Güzelkaya M., Eksin I., Gürleyen F., A new methodology for designing a fuzzy logic controller and PI, PD blending mechanism, *Journal of Intelligent and Fuzzy Systems*, 2001, **11**, 85-98.
- [58] Arzen K. E., Johansson M., Babuska R., *Fuzzy algorithms for control*, Kluwer Academic Publishers, Boston, USA, 1999.
- [59] Chen C. L., Chen P. C., Chen C. K., Analysis and design of fuzzy control systems, *Fuzzy Sets and Systems*, 1993, **57**, 125-140.
- [60] Li H. X., Gatland H. B., Conventional fuzzy control and its enhancement, *IEEE Trans. On Systems, Man, Cybernetics-Part B*, 1996, **26**, 791-797.
- [61] Karaođlan Ö. G., Kontrol Sistemleri İçin Bulanık PID Kontrolörlerin Genetik Algoritmalar Yardımıyla Ayarlanması, Yüksek Lisans Tezi, İ.T.Ü., Fen Bilimleri Enstitüsü, İstanbul, 2007, 223197.
- [62] Rezek S. F., Elsodany N. M., Maharem N. A., Fuzzy Gain Scheduling Control of a Stepper Motor Driving a Flexible Rotor Approach, *European Journal of Scientific Research*, 2010, **39**, 50-63.
- [63] Shao J., Chen L., Sun Z., The Application of Fuzzy Control Strategy in Electrohydraulic Servo System, *In Proceedings of the IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada, 29 July - 1 August 2005.
- [64] Akbıyık B., Eksin İ., Güzelkaya M., Yeşil E., Evaluation of the performance of various fuzzy PID controller structures on benchmark systems, *ELECO '2005, 4rd International Conf. on Electrical and Electronics Engineering*, Bursa, Turkey, 5-9 December 2005.
- [65] Maeda M., Murakami S., Fuzzy gain scheduling of PID controllers, *Fuzzy Sets and Systems*, 1992, **51**, 29-40.
- [66] Qiao W. Z., Mizumoto M., PID type fuzzy controller and parameters adaptive method, *Fuzzy Sets and Systems*, 1996, **78**, 23-35.
- [67] Er M. J., Sun Y. L., Hybrid fuzzy proportional integral plus conventional derivative control of linear and nonlinear systems, *IEEE Trans. On Industrial Electronics*, 2001, **48**, 1109-1117.
- [68] Erenođlu I., Akıllı Melez Bulanık PID Kontrol Yapısı, Yüksek Lisans Tezi, İ.T.Ü., Fen Bilimleri Enstitüsü, İstanbul, 2006, 223826.

- [69] Kennedy J., Eberhart R. C., Particle Swarm Optimization, *Proc. IEEE International Conference on Neural Networks*, Perth Australia, IEEE Service Center, Piscataway, NJ, IV, 27 November - 01 December 1995.
- [70] Eberhart R. C., Simpson P. K., Dobbins R. W., *Computational Intelligent PC Tools*, Academic Press Professional, Boston, MA, 1996.
- [71] Schwefel H. P., *Evolution and Optimum Seeking*, Wiley, New York, 1995.
- [72] Eberhart R. C., Shi Y., In Evolutionary Programming VII, Editors: Porto V. W., Saravanan N., Waagen D., Eiben A. E., *Comparison between genetic algorithms and particle swarm optimization*, Springer, 611-616, 1998.
- [73] Daniel P., Xiaodong L., Locating and Tracking Multiple Dynamic Optima by a Particle Swarm Model Using Speciation, *IEEE Transaction on Evolutionary Computation*, 2006, **10**, 440-458.
- [74] Denavit J., Hartenberg R. S., A kinematic Notation for Lower-Pair Mechanisms Based on Matrices, *Journal of Applied Mechanics*, 1955, **1**, 215-221.
- [75] Bingul Z., Küçük S., *Robot Teknigi-I*, Birsen Yayınevi, İstanbul, 2005.
- [76] Craig J. J., *Introduction to Robotics: Mechanics and Control*, 2nd edition, Addison Wesley, USA, 1989.
- [77] Lee C. S. G., Robot Arm Kinematics, Dynamics and Control, *Computer*, 1982, **15**, 62-80.
- [78] Paul R. P. C., *Robot Manipulators: Mathematics, Programming and Control*, MIT Press, USA, 1981.
- [79] Hollerbach J. M., A Recursive Lagrangian Formulation of Manipulator Dynamics and A Comparative Study of Dynamic Formulation Complexity, *IEEE Trans. Systems, Man and Cybernetics.*, 1980, **10**, 730-736.
- [80] Walker J. Y. S., Lu M. W., Paul R. P. C., On-Line Computational Scheme for Mechanical Manipulators, *Trans. ASME, J. Dynamic Systems, Measurement and Control*, 1980, **102**, 69-76.
- [81] Lee C. S. G., Lee B. H., Nigam N., Development of the Generalized D'Alembert Equations of Motion for Robot Manipulators, *Proc. Of 22nd Conf. on Decision and Control*, San Antonio, Texas, 14-16 December 1983.
- [82] Kane T. R., Levinson D. A., "The Use of Kane's Dynamic Equations in Robotics, *International Journal of Robotics Research*, 1983, **2**, 3-20.
- [83] Lewis F. L., Abdallah C. T., Dawson D. M., *Control Of Robot Manipulators*, Macmillan Publishing Company, Oxford, 1993.

- [84] Paul R. P., Shimano B., Mayer G., Kinematic Control Equations for Simple Manipulators, *Transactions on Systems, Man and Cybernetics*, 1981, **11**, 449-455.
- [85] Duran M. A., Ankaralı A., Üç Serbestlik Dereceli PUMA Tipi Bir Manipülörün PID Kontrolü, *Teknik-Online Dergi*, 2008, **9**, 79-98.
- [86] Zhao Z. Y., Tomizuka M., Isaka S., Fuzzy gain scheduling of PID controllers, *IEEE transactions on systems, man and cybernetics*, 1993, **23**, 1392-1398.
- [87] Santos M., Dexter A. L., Control of a cryogenic process using a fuzzy PID scheduler, *Control engineering practice*, 2002, **10**, 1147-1152.
- [88] Zulfatman Rahmat M. F., Application of Self-Tuning Fuzzy PID Controller On Industrial Hydraulic Actuator Using System Identification Approach, *International J. on Smart Sensing and Intelligent Systems*, 2009, **2**, 246-261.
- [89] Khoury G. M., Saad M., Kanaan H. Y., Asmar C., Fuzzy PID Control of a Five DOF Robot Arm, *Journal of Intelligent and Robotic Systems*, 2004, **40**, 299-320.
- [90] Alassar A. Z., Abuhadrous I. M., Elaydi H. A., Comparison Between FLC and PID Controller For 5 DOF Robot Arm, *2nd International Conference on Advanced Computer Control*, Shenyang, China, 29-31 January 2010.
- [91] Ham C., Jhonson Z., Robust Fuzzy Control For Robot Manipulators, *Control Theory and Applications*, 2000, **147**, 212-216.
- [92] Malki H. A., Misir D., Feigenspan D., Chen G., Fuzzy PID Control of a Flexible-Joint Robot Arm with Uncertainties from Time-Varying Loads, *IEEE Transaction On Control Systems Technology*, 1997, **5**, 371-378.
- [93] An T. H., Nguyen H. C., Sok J. H., Lee W. S., Han S. H., A Robust Neural Network Control of Robot Manipulator for Industrial Application, *International Conference on Control, Automation and Systems, KINTEX*, Korea, 27-30 October 2010.
- [94] Hesselroth T., Sarkar K., Smagt P. P., Schulten K., Neural Network Control of a Pneumatic Robot Arm, *Transactions On Systems, Man and Cybernetics*, 1994, **24**, 28-38.
- [95] Zannatha J. M. I., Bassi D. F., Garcia R. A., Position and Differential Kinematic Neural Control of Robot Manipulators: A Comparison Between Two Schemes, *Systems, Man and Cybernetics*, 1993, **4**, 479-484.
- [96] Zhang W., Qi N., Yin H., PD Control of Robot Manipulators with Uncertainties Based on Neural Network, *International Conference on Intelligent Computation Technology and Automation, ICICTA*, Changsha, China, 11-12 May 2010.

- [97] Marichal G. N., Acosta L., Moreno L., Mendez J. A., Rodrigo J. J., Sigut M., Obstacle Avoidance for a Mobile robot: A Neuro-Fuzzy Approach, *Fuzzy Sets and Systems*, 2001, **124**, 171-179.
- [98] Low C. B., Nah K. H., Er M. J., Real-Time Implementation of a Dynamic Fuzzy Neural Networks Controller for a Scara, *Journal of the Institution of Engineers*, 2004, **44**, 449-461.
- [99] Alavandar S., Nigam M. J., Adaptive Neuro-Fuzzy Inference System Based Control of Six DOF Robot Manipulator, *Journal of Engineering Science and Technology Review*, 2008, **1**, 106-111.
- [100] Durmaz U., Bir Robot Kolunun Sinirsel Bulanık Kontrolü, Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya, 2007, 212202.
- [101] Wai R. J., Chen P. C., Robust Neural-Fuzzy-Network Control For Robot Manipulator Including Actuator Dynamics, *IEEE Transactions on Industrial Electronics*, 2006, **53**, 1328-1349.
- [102] Wai R. J., Yang Z. W., Adaptive Fuzzy-Neural-Network Control of Robot Manipulator Using T-S Fuzzy Model Design, *IEEE International Conference on Fuzzy Systems*, Hong Kong, China, 1-6 June 2008.
- [103] Kwok D. P., Leung T. P., Sheng F., Genetic Algorithms for Optimal Control of Robot Arms, *International Conference on Industrial Electronics, Control and Instrumentation, IECON*, Hawaii, USA, 15-19 November 1993.
- [104] Homaifar A., Bikdash M., Gopalan V., Design Using Genetic Algorithms of Hierarchical Hybrid Fuzzy-PID Controllers of Two-Link Robotic Arms, *Journal of Robotic Systems*, 1997, **14**, 449-463.
- [105] Ahn K. K., Anh H. P. H., System Modeling and Identification The Two Link Pneumatic Artificial Muscle (PAM) Manipulator Optimized With Genetic Algorithms, *International Joint Conference, SICE-ICASE*, Korea, 18-21 October 2006.
- [106] Wong C. C., Wang H. H., Li S. A., PSO-based Motion Fuzzy Controller Design For Mobile Robots, *International Journal of Fuzzy Systems*, 2008, **10**, 24-32.
- [107] Bingul Z., Karahan O., Tuning of Fractional PID Controllers Using PSO Algorithm for Robot Trajectory Control, *International Conference on Mechatronics, ICM*, Istanbul, Turkey, 13-15 April 2011.
- [108] Mazhari S. A., Kumar S., Heuristic Search Algorithms for Tuning PUMA 560 Fuzzy PID Controller, *International Journal of Computer Science*, 2008, **3**, 277-286.

- [109] Bingul Z., Karahan O., A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control, *Expert Systems with Applications*, 2011, **38**, 1017-1031.
- [110] Nguyen V. B., Morris A. S., Using a genetic algorithm to fully optimize a fuzzy logic controller for a two-link flexible robot arm, *Robotica*, 2009, **27**, 677-687.
- [111] Fichter E. F., A Stewart Platform Based Manipulator: General Theory and Practical Construction, *The International journal of Robotics Research*, 1986, **5**, 157-182.
- [112] Merlet J. P., Parallel Manipulators, Parts 1: Theory, design, kinematics, dynamics and control, *Institute National de Recherche en Informatique Automatique Rapport de Recherche (INRIA)*, Technical Report 646, 1987.
- [113] Do W. Q. D., Yang D. C. H., Inverse Dynamic Analysis and Simulation of a Platform Type of Robot, *Journal of Robotic System*, 1988, **5**, 209-227.
- [114] Reboulet C., Berthomjeu T., Dynamic Models of A Six Degree of Freedom Parallel Manipulators, in *Proc. ICAR*, Italy, 19-22 June 1991.
- [115] Khalil W., Ibrahim O., General Solution for the Dynamic Modeling of Parallel Robots, *IEEE Inter. Conf. On Robotics and Automation*, New Orleans, LA, 26 April - 1 May 2004.
- [116] Riebe S., Ulbrich H., Modelling and online computation of the dynamics of a parallel kinematic with six degrees-of-freedom, *Archive of Applied Mechanics*, 2003, **72**, 817-829.
- [117] Wang J., Wu J., Wang L., Li T., Simplified strategy of the dynamic model of a 6-UPS parallel kinematic machine for real-time control, *Mechanism and Machine Theory*, 2007, **42**, 1119-1140.
- [118] Dasgupta B., Mruthyunjaya T. S., Closed-Form Dynamic Equations of The General Stewart Platform Through The Newton-Euler Approach, *Mechanism and Machine Theory*, 1998, **33**, 993-1012.
- [119] Khalil W., Guegan S., Inverse and Direct Dynamic Modeling of Gough–Stewart Robots, *IEEE Transactions on Robotics*, 2004, **20**, 754-762.
- [120] Harib K., Srinivasan K., Kinematic and dynamic analysis of Stewart platform-based machine tool structures, *Robotica*, 2003, **21**, 541-554.
- [121] Khalil W., Ibrahim O., General solution for the dynamic modelling of parallel robots, *J. Intelligent Robot Systems*, 2007, **49**, 19-37.
- [122] Fried G., Djouni K., Borojeni D., Iqbal S., An Inverse Dynamic Model of the Gough-Stewart Platform, *Wseas Transactions on Systems*, 2008, **7**, 88-99.

- [123] Carvalho J., Ceccarelli M., A Closed-Form Formulation for the Inverse Dynamics of a Cassino Parallel Manipulator, *Multibody System Dynamics*, 2001, **5**, 185-210.
- [124] Ibrahim O., Khalil W., Inverse and direct dynamic models of hybrid robots, *Mechanism and Machine Theory*, 2010, **45**, 627-640.
- [125] Gui X., Wang W., Inverse Dynamic Model and Simulation for the Planar Parallel Manipulator, *International Conference on Computer Application and System Modeling (ICCASM)*, Taiyuan, China, 22-24 October 2010.
- [126] Zanganeh K. E., Sinatra R., Angeles J., Kinematics and dynamics of a six-degree-of-freedom parallel manipulator with revolute legs, *Robotica*, 1997, **15**, 385-394.
- [127] Abdellatif H., Heimann B., Computational efficient inverse dynamics of 6-DOF fully parallel manipulators by using the Lagrangian formalism, *Mechanism and Machine Theory*, 2009, **44**, 192-207.
- [128] Guo H., Li H., Dynamic analysis and simulation of a six degree of freedom Stewart platform manipulator, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2006, **220**, 61-72.
- [129] Harib K. H., Dynamic modeling, identification and control of Stewart Platform based machine tools, Ph.D. Dissertation, Department of Mechanical Engineering, The Ohio State University, USA, 1997.
- [130] Bhattacharya S., Hatwal H., Ghosh A., An on line parameter estimation scheme for generalized Stewart Platform type parallel manipulators, *Mechanism Machine Theory*, 1997, **32**, 79-89.
- [131] Lin J., Chen C. W., Computer-aided-symbolic dynamic modeling for Stewart-platform manipulator, *Robotica*, 2009, **27**, 331-341.
- [132] Hajimirzaalian H., Moosavi H., Massah M., Dynamics analysis and simulation of parallel robot Stewart platform, *Computer and Automation Engineering (ICCAE)*, 2010, **5**, 472-477.
- [133] Beji L., Pascal M., The Kinematics and the Full Minimal Dynamic Model of a 6-DOF Parallel Robot Manipulator, *Nonlinear Dynamics*, 1999, **18**, 339-356.
- [134] Nguyen C. C., Pooran F. J., Dynamic Analysis of 6 DOF CKCM Robot End Effectors for Dual Arm Tele robot Systems, *Journal of Robotics and Autonomous Systems*, 1989, **5**, 377-394.
- [135] Pang H., Shahingpoor M., Inverse Dynamics of a parallel manipulator, *Journal of Robotic System*, 1994, **11**, 693-702.

- [136] Lebert G., Liu K., Lewis F. L., Dynamic Analysis and Control of a Stewart Platform Manipulator, *Journal of Robotic systems*, 1993, **10**, 629-655.
- [137] Wang J., Gosselin C. M., A New Approach for the Dynamic Analysis of Parallel Manipulators, *Multi body System Dynamics*, 1998, **2**, 317-334.
- [138] Li Y., Xu Q., Dynamic modeling and robust control of a 3-PRC translational parallel kinematic machine, *Robotics and Computer Integrated Manufacturing*, 2009, **25**, 630-640.
- [139] Staicu S., Dynamics of the 6-6 Stewart parallel manipulator, *Robotics and Computer-Integrated Manufacturing*, 2011, **27**, 212-220.
- [140] Tsai L. W., Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work, *Journal of Mechanical Design*, 2000, **122**, 3-9.
- [141] Staicu S., Zhang D., A novel dynamic modelling approach for parallel mechanisms analysis, *Robotics and Computer-Integrated Manufacturing*, 2008, **24**, 167-172.
- [142] Staicu S., Dynamics analysis of the Star parallel manipulator, *Robotics and Autonomous Systems*, 2009, **57**, 1057-1064.
- [143] Tsai L. W., *Robot Analysis: The Mechanics of Serial and Parallel Manipulator*, John Wiley & Sons, Inc, New York, 1999.
- [144] Zhang C. D., Song S. M., An efficient method for inverse dynamics of manipulators based on virtual work principle, *Journal of Robotic Systems*, 1993, **10**, 605-627.
- [145] Liu M. J., Li C. X., Li C. N., Dynamics analysis of the Gough-Stewart Platform manipulator, *IEEE Trans. Robot. Autom.*, 2000, **16**, 94-98.
- [146] Mengi Q., Zhang T., He J. F., Song J. Y., Han J. W., Dynamic Modeling of a 6 Degree of Freedom Stewart Platform Driven By a Permanent Magnet Synchronous Motor, *Journal of Zhejiang Univesity Science*, 2010, **11**, 751-761.
- [147] Jianming M., Jingfeng H., Xiong H., Han J. W., Simulation of Coupling Characteristic of Hydraulically Driven Stewart Platform Based on Dynamics Model, *Modelling, Simulation and Optimization WMSO '08*, Hong Kong, China, 27-28 December 2008.
- [148] Yang C., Huang Q., Ye Z., Han J., Dynamic Modeling of Spatial 6-DOF Parallel Robots Using Kane Method for Control Purposes, *Intelligent Human-Machine Systems and Cybernetics*, 2010, **2**, 180-183.
- [149] Gallardo J., Rico J. M., Checcacci D., Bergamasco M., Dynamics of parallel manipulators by means of screw theory, *Mechanism Machine Theory*, 2003, **38**, 1113-1131.

- [150] Lopes A. M., Dynamic modeling of a Stewart platform using the generalized momentum approach, *Commun Nonlinear Sci Numer Simulat*, 2009, **14**, 3389-3401.
- [151] Nguyen C. C., Antraz S. S., Zhou Z. L., Campbell C. E., Adaptive control of a Stewart platform-based manipulators, *Journal of Robotic Systems*, 1993, **10**, 57-687.
- [152] Davliakos I., Papadopoulos E., Model-based control of a 6-dof electrohydraulic Stewart–Gough platform, *Mechanism and Machine Theory*, 2008, **43**, 1385-1400.
- [153] Fu S., Yao Y., Non-linear robust control with partial inverse dynamic compensation for a Stewart platform manipulator, *Int. J. Modelling, Identification and Control*, 2006, **1**, 44-51.
- [154] Lee S. H., Song J. B., Choi W. C., Hong D., Position control of a Stewart platform using inverse dynamics control with approximate dynamics, *Mechatronics*, 2003, **13**, 605-619.
- [155] Ghobakhloo A., Eghtesad M., Azadi M., Position Control of a Stewart-Gough Platform Using Inverse Dynamics Method with Full Dynamics, *9th IEEE International Workshop on Advanced Motion Control*, İstanbul, Turkey, 27-29 March 2006.
- [156] Zubizarreta A., Marcos M., Cabanes I., Pinto C., A procedure to evaluate Extended Computed Torque Control configurations in the Stewart–Gough platform, *Robotics and Autonomous Systems*, 2011, **59**, 770-781.
- [157] Su Y. X., Duan B. Y., Zheng C. H., Zhang Y. F., Chen G. D., Mi J. W., Disturbance-Rejection High-Precision Motion Control of a Stewart Platform, *IEEE Transaction on Control Systems Technology*, 2004, **12**, 364-374.
- [158] Yang C., Huang Q., Jiang H., Peter O. O., Han J., PD control with gravity compensation for hydraulic 6-DOF parallel manipulator, *Mechanism and Machine Theory*, 2010, **45**, 666-677.
- [159] Su Y. X., Duan B. Y., Zheng C. H., Nonlinear PID control of a six-DOF parallel manipulator, *IEEE Proceedings - Control Theory and Applications*, 2004, **151**, 95-102.
- [160] Omran A., El-Bayiumi G., Bayoumi M., Kassem A., Genetic Algorithm Based Optimal Control for a 6-DOF Non Redundant Stewart Manipulator, *International Journal of Mechanical Systems Science and Engineering*, 2008, **2**, 73-79.
- [161] Kassem A., Omran A., Optimal task space control design of a Stewart manipulator for aircraft stall recovery, *Aerospace Science and Technology*, 2011, **15**, 353-365.

- [162] Sun J., Ding Y. S., Hoa K. R., Particle Swarm Optimizing Clonal Algorithm to Design an Intelligent PID Controller with Application to 3-2-1 Stewart Platform, *Second International Symposium on Intelligent Information Technology Application*, Shanghai, China, 20-22 December 2008.
- [163] Kim N. I., Lee C. W., High Speed Tracking Control of Stewart Platform Manipulator via Enhanced Sliding Mode Control, *Robotics and Automation, IEEE International Conference on*, Belgium, 16-20 May 1998.
- [164] Huang C. I., Chang C. F., Yu M. Y., Fu L. C., Sliding-Mode Tracking Control of the Stewart Platform, *IEEE, 5th Asian Control Conference*, Australia, 20-23 July 2004.
- [165] Wu B., Dong Y., Wu S., Xu D., Zhao K., An Integral Variable Structure Controller with Fuzzy Tuning Design for Electro-hydraulic Driving Stewart Platform, *IEEE, 1th International Symposium on Systems and Control In Aerospace and Astronautics*, Harbin, China, 19-21 January 2006.
- [166] Dongsu W., Hongbin G., Adaptive Sliding Control of Six-DOF Flight Simulator Motion Platform, *Chinese Journal of Aeronautics*, 2007, **20**, 425-433.
- [167] Guo H. B., Liu Y. G., Liu G. R., Li H. R., Cascade control of a hydraulically driven 6-DOF parallel robot manipulator based on a sliding mode, *Control Engineering Practice*, 2008, **16**, 1055-1068.
- [168] Negash D. S., Mitra R., Integral Sliding Mode Controller for Trajectory Tracking Control of Stewart Platform Manipulator, *5th International Conference on Industrial and Information Systems*, India, 29 July - 1 August 2010.
- [169] Pi Y., Wang X., Trajectory tracking control of a 6-DOF hydraulic parallel robot manipulator with uncertain load disturbances, *Control Engineering Practice*, 2011, **19**, 185-193.
- [170] Fraguera L., Fridman L., Alexandrov V. V., Position stabilization of a Stewart platform: High-order sliding mode observers based approach, *Journal of The Franklin Institute*, 2012, **349**, 441-455.
- [171] Chung F., Chang H. H., Lin C. T., Fuzzy control of a six-degree motion platform with stability analysis, *Systems, Man and Cybernetics*, 1999, **1**, 325-330.
- [172] Mann G. K., Surgenor B. W., Model-free intelligent control of a 6-DOF Stewart-Gough based parallel manipulator, *Proceedings of the 2002 International Conference on Control Applications*, Glasgow, Scotland, United Kingdom, 18-20 September 2002.

- [173] Remillard V., Boukas E. K., Gough-Stewart Platform Control: A Fuzzy control approach, *Annual meeting of the North American Fuzzy Information Processing Society*, Montreal, Quebec, Canada, 3-6 June 2006.
- [174] Wu D., Gu H., Li P., Adaptive Fuzzy Control of Stewart Platform under Actuator, *World Academy of Science, Engineering and Technology*, 2009, **60**, 680-684.
- [175] Hsu C. H., Liang S. F., Lin C. J., Wu S. Y., An Implementation of Functional Neural Fuzzy Controller for the Electrical 6-DOF Stewart Platform, *International Conference on System Science and Engineering*, Macau, China, 8-10 June 2011.
- [176] Zhu D., Mei T., Sun L., Fuzzy Support Vector Machines Control for 6-DOF Parallel Robot, *Journal of Computers*, 2011, **1**, 1926-1934.
- [177] Bo Y., Zhongcai P., Zhiyong T., Fuzzy PID control of Stewart platform, *International Conference on Fluid Power and Mechatronics*, Beijing, China, 17-20 August 2011.
- [178] Ömürlü V. E., Yıldız İ., Self-tuning fuzzy PD-based stiffness controller of a 3×3 Stewart platform as a man-machine interface, *Turk J Elec Eng & Comp Sci*, 2011, **19**, 743-752.
- [179] Ma J., Yang T., Hou Z. G., Tan M., Neural Network Disturbance Observer Based Controller of an Electrically Driven Stewart Platform Using Backstepping for Active Vibration Isolation, *International Joint Conference on Neural Networks*, Atlanta, 14-19 June 2009.
- [180] Yıldız İ., Ömürlü V. E., Engin Ş. N., Bulanık Mantık ve Yapay Sinir Ağları ile bir 3-3 Stewart Platformu'nun Pozisyon Kontrolü, *TOK'2008*, İstanbul, Türkiye, 13-15 Kasım 2008.

EKLER

EK-A

Dosya: Jacobian_A.m

```
syms m1 m2 GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
syms GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
syms alpha beta gamma h1 h2 h3 h4 h5 h6 k1 k2 k3 k4 k5 k6
syms ux1 uy1 uz1 ux2 uy2 uz2 ux3 uy3 uz3 ux4 uy4 uz4 ux5 uy5 uz5 ux6 uy6 uz6
J1 J2 J
```

% Jacobian J1 Matrisi

```
J1=[ux1 uy1 uz1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 ux2 uy2 uz2 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 ux3 uy3 uz3 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 ux4 uy4 uz4 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 ux5 uy5 uz5 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ux6 uy6 uz6];
```

% Jacobian J2 Matrisinin Elemanlarının Hesaplanması

```
RxAlpha=[1 0 0;0 cos(alpha) -sin(alpha);0 sin(alpha) cos(alpha)];
RyBeta= [cos(beta) 0 sin(beta);0 1 0 ;-sin(beta) 0 cos(beta)];
RzGamma= [cos(gamma) -sin(gamma) 0;sin(gamma) cos(gamma)0;0 0 1];
SX=[0 0 0;0 0 -1;0 1 0];SY=[0 0 1;0 0 0;-1 0 0];SZ=[0 -1 0;1 0 0;0 0 0];
```

% 4. stun

```
GT41=[GTx1 GTy1 GTz1];R41=RyBeta*SX*RxAlpha*RzGamma*GT41.';
R1GTx1=R41(1);R1GTy1=R41(2);R1GTz1=R41(3);
GT42=[GTx2 GTy2 GTz2];R42=RyBeta*SX*RxAlpha*RzGamma*GT42.';
R1GTx2=R42(1);R1GTy2=R42(2);R1GTz2=R42(3);
GT43=[GTx3 GTy3 GTz3];R43=RyBeta*SX*RxAlpha*RzGamma*GT43.';
R1GTx3=R43(1);R1GTy3=R43(2);R1GTz3=R43(3);
GT44=[GTx4 GTy4 GTz4];R44=RyBeta*SX*RxAlpha*RzGamma*GT44.';
R1GTx4=R44(1);R1GTy4=R44(2);R1GTz4=R44(3);
GT45=[GTx5 GTy5 GTz5];R45=RyBeta*SX*RxAlpha*RzGamma*GT45.';
R1GTx5=R45(1);R1GTy5=R45(2);R1GTz5=R45(3);
GT46=[GTx6 GTy6 GTz6];R46=RyBeta*SX*RxAlpha*RzGamma*GT46.';
R1GTx6=R46(1);R1GTy6=R46(2);R1GTz6=R46(3);
```

%5. stun

```
GT51=[GTx1 GTy1 GTz1];R51=SY*RyBeta*RxAlpha*RzGamma*GT51.';
R2GTx1=R51(1);R2GTy1=R51(2);R2GTz1=R51(3);
GT52=[GTx2 GTy2 GTz2];R52=SY*RyBeta*RxAlpha*RzGamma*GT52.';
R2GTx2=R52(1);R2GTy2=R52(2);R2GTz2=R52(3);
GT53=[GTx3 GTy3 GTz3];R53=SY*RyBeta*RxAlpha*RzGamma*GT53.';
R2GTx3=R53(1);R2GTy3=R53(2);R2GTz3=R53(3);
GT54=[GTx4 GTy4 GTz4];R54=SY*RyBeta*RxAlpha*RzGamma*GT54.';
```

```

R2GTx4=R54(1);R2GTy4=R54(2);R2GTz4=R54(3);
GT55=[GTx5 GTy5 GTz5];R55=SY*RyBeta*RxAlpha*RzGamma*GT55.!'
R2GTx5=R55(1);R2GTy5=R55(2);R2GTz5=R55(3);
GT56=[GTx6 GTy6 GTz6];R56=SY*RyBeta*RxAlpha*RzGamma*GT56.!'
R2GTx6=R56(1);R2GTy6=R56(2);R2GTz6=R56(3);

```

%6. stun

```

GT61=[GTx1 GTy1 GTz1];R61=RyBeta*RxAlpha*SZ*RzGamma*GT61.!'
R3GTx1=R61(1);R3GTy1=R61(2);R3GTz1=R61(3);
GT62=[GTx2 GTy2 GTz2];R62=RyBeta*RxAlpha*SZ*RzGamma*GT62.!'
R3GTx2=R62(1);R3GTy2=R62(2);R3GTz2=R62(3);
GT63=[GTx3 GTy3 GTz3];R63=RyBeta*RxAlpha*SZ*RzGamma*GT63.!'
R3GTx3=R63(1);R3GTy3=R63(2);R3GTz3=R63(3);
GT64=[GTx4 GTy4 GTz4];R64=RyBeta*RxAlpha*SZ*RzGamma*GT64.!'
R3GTx4=R64(1);R3GTy4=R64(2);R3GTz4=R64(3);
GT65=[GTx5 GTy5 GTz5];R65=RyBeta*RxAlpha*SZ*RzGamma*GT65.!'
R3GTx5=R65(1);R3GTy5=R65(2);R3GTz5=R65(3);
GT66=[GTx6 GTy6 GTz6];R66=RyBeta*RxAlpha*SZ*RzGamma*GT66.!'
R3GTx6=R66(1);R3GTy6=R66(2);R3GTz6=R66(3);

```

```

J2= [1  0  0  R1GTx1  R2GTx1  R3GTx1;
      0  1  0  R1GTy1  R2GTy1  R3GTy1;
      0  0  1  R1GTz1  R2GTz1  R3GTz1;
      1  0  0  R1GTx2  R2GTx2  R3GTx2;
      0  1  0  R1GTy2  R2GTy2  R3GTy2;
      0  0  1  R1GTz2  R2GTz2  R3GTz2;
      1  0  0  R1GTx3  R2GTx3  R3GTx3;
      0  1  0  R1GTy3  R2GTy3  R3GTy3;
      0  0  1  R1GTz3  R2GTz3  R3GTz3;
      1  0  0  R1GTx4  R2GTx4  R3GTx4;
      0  1  0  R1GTy4  R2GTy4  R3GTy4;
      0  0  1  R1GTz4  R2GTz4  R3GTz4;
      1  0  0  R1GTx5  R2GTx5  R3GTx5;
      0  1  0  R1GTy5  R2GTy5  R3GTy5;
      0  0  1  R1GTz5  R2GTz5  R3GTz5;
      1  0  0  R1GTx6  R2GTx6  R3GTx6;
      0  1  0  R1GTy6  R2GTy6  R3GTy6;
      0  0  1  R1GTz6  R2GTz6  R3GTz6];

```

Dosya: Jacobian_B.m

```

syms m1 m2 GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
syms GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
syms alpha beta gamma ux1 uy1 uz1 ux2 uy2 uz2
syms ux3 uy3 uz3 ux4 uy4 uz4 ux5 uy5 uz5 ux6 uy6 uz6

```

```

ax1=GTx1;ay1=GTy1;az1=GTz1;ax2=GTx2;ay2=GTy2;az2=GTz2;
ax3=GTx3;ay3=GTy3;az3=GTz3;ax4=GTx4;ay4=GTy4;az4=GTz4;
ax5=GTx5;ay5=GTy5;az5=GTz5;ax6=GTx6;ay6=GTy6;az6=GTz6;

```

```

n1=[ux1 uy1 uz1]';n2=[ux2 uy2 uz2]';n3=[ux3 uy3 uz3]';
n4=[ux4 uy4 uz4]';n5=[ux5 uy5 uz5]';n6=[ux6 uy6 uz6]';
a1=[ax1 ay1 az1]';a2=[ax2 ay2 az2]';a3=[ax3 ay3 az3]';
a4=[ax4 ay4 az4]';a5=[ax5 ay5 az5]';a6=[ax6 ay6 az6]';

ROT=[cos(gamma)*cos(beta)
sin(gamma)*cos(alpha)+cos(gamma)*sin(beta)*sin(alpha)
sin(gamma)*sin(alpha)+cos(gamma)*sin(beta)*cos(alpha);sin(gamma)*cos(beta)
cos(gamma)*cos(alpha)+sin(gamma)*sin(beta)*sin(alpha)
cos(gamma)*sin(alpha)+sin(gamma)*sin(beta)*cos(alpha);-sin(beta)
cos(beta)*sin(alpha) cos(beta)*cos(alpha)];

J12=cross((ROT*a1),n1);
j11=n1(1);j12=n1(2);j13=n1(3);j14=J12(1);j15=J12(2);j16=J12(3);
J22=cross((ROT*a2),n2);
j21=n2(1);j22=n2(2);j23=n2(3);j24=J22(1);j25=J22(2);j26=J22(3);
J32=cross((ROT*a3),n3);
j31=n3(1);j32=n3(2);j33=n3(3);j34=J32(1);j35=J32(2);j36=J32(3);
J42=cross((ROT*a4),n4);
j41=n4(1);j42=n4(2);j43=n4(3);j44=J42(1);j45=J42(2);j46=J42(3);
J52=cross((ROT*a5),n5);
j51=n5(1);j52=n5(2);j53=n5(3);j54=J52(1);j55=J52(2);j56=J52(3);
J62=cross((ROT*a6),n6);
j61=n6(1);j62=n6(2);j63=n6(3);j64=J62(1);j65=J62(2);j66=J62(3);

% Jacobian J1 Matrisi

J1=[j11 j12 j13 j14 j15 j16;
j21 j22 j23 j24 j25 j26;
j31 j32 j33 j34 j35 j36;
j41 j42 j43 j44 j45 j46;
j51 j52 j53 j54 j55 j56;
j61 j62 j63 j64 j65 j66];

% Jacobian J2 Matrisi

J2=[1 0 0 0 0 0;
0 1 0 0 0 0;
0 0 1 0 0 0;
0 0 0 cos(beta) 0 0;
0 0 0 0 1 -sin(alpha);
0 0 0 -sin(beta) 0 cos(alpha)];

Dosya: UstPlatformKutleMatrisi.m

clc;
clear all;
syms alpha beta gamma
syms Ix Iy Iz
syms dalpha dbeta dgamma

```

```

Xpo=[alpha beta gamma].';dXpo=[dalpha dbeta dgamma].';
RxAlpha=[1 0 0;0 cos(alpha) -sin(alpha);0 sin(alpha) cos(alpha)];
RyBeta= [cos(beta) 0 sin(beta);0 1 0 ;-sin(beta) 0 cos(beta)];
RzGamma= [cos(gamma) -sin(gamma) 0;sin(gamma) cos(gamma)0;0 0 1];

rot=RzGamma*RyBeta*RxAlpha;
I=diag([Ix Iy Iz]);
Omega_B=RyBeta*diag([1 0 0])+ diag([0 1 0])+RxAlpha*RzGamma*diag([0 0 1]);
Omega_U=simplify(transpose(RzGamma)*transpose(RxAlpha)*transpose(RyBeta)*
Omega_B);
Mup=(transpose(Omega_U)*I*Omega_U);
Massup=[Mup(1,1) Mup(1,2) Mup(1,3)
         Mup(2,1) Mup(2,2) Mup(2,3)
         Mup(3,1) Mup(3,2) Mup(3,3)];

```

Dosya: BacakKutleMatrisi_A.m

```

syms m1 m2 GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
syms GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
syms alpha beta gamma h1 h2 h3 h4 h5 h6 k1 k2 k3 k4 k5 k6
syms ux1 uy1 uz1 ux2 uy2 uz2 ux3 uy3 uz3 ux4 uy4 uz4 ux5 uy5 uz5 ux6 uy6 uz6
J1 J2 J

```

```

J1=[ux1 uy1 uz1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 ux2 uy2 uz2 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 ux3 uy3 uz3 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 ux4 uy4 uz4 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 ux5 uy5 uz5 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ux6 uy6 uz6];

```

```

RxAlpha=[1 0 0;0 cos(alpha) -sin(alpha);0 sin(alpha) cos(alpha)];
RyBeta= [cos(beta) 0 sin(beta);0 1 0 ;-sin(beta) 0 cos(beta)];
RzGamma= [cos(gamma) -sin(gamma) 0;sin(gamma) cos(gamma)0;0 0 1];
SX=[0 0 0;0 0 -1;0 1 0];SY=[0 0 1;0 0 0;-1 0 0];SZ=[0 -1 0;1 0 0;0 0 0];

```

% 4. stun

```

GT41=[GTx1 GTy1 GTz1];
R41=RyBeta*SX*RxAlpha*RzGamma*GT41.';
R1GTx1=R41(1);R1GTy1=R41(2);R1GTz1=R41(3);
GT42=[GTx2 GTy2 GTz2];
R42=RyBeta*SX*RxAlpha*RzGamma*GT42.';
R1GTx2=R42(1);R1GTy2=R42(2);R1GTz2=R42(3);
GT43=[GTx3 GTy3 GTz3];
R43=RyBeta*SX*RxAlpha*RzGamma*GT43.';
R1GTx3=R43(1);R1GTy3=R43(2);R1GTz3=R43(3);
GT44=[GTx4 GTy4 GTz4];
R44=RyBeta*SX*RxAlpha*RzGamma*GT44.';
R1GTx4=R44(1);R1GTy4=R44(2);R1GTz4=R44(3);
GT45=[GTx5 GTy5 GTz5];

```



```

R45=RyBeta*SX*RxAlpha*RzGamma*GT45.';
R1GTx5=R45(1);R1GTy5=R45(2);R1GTz5=R45(3);
GT46=[GTx6 GTy6 GTz6];
R46=RyBeta*SX*RxAlpha*RzGamma*GT46.';
R1GTx6=R46(1);R1GTy6=R46(2);R1GTz6=R46(3);

```

%5. stun

```

GT51=[GTx1 GTy1 GTz1];
R51=SY*RyBeta*RxAlpha*RzGamma*GT51.';
R2GTx1=R51(1);R2GTy1=R51(2);R2GTz1=R51(3);
GT52=[GTx2 GTy2 GTz2];
R52=SY*RyBeta*RxAlpha*RzGamma*GT52.';
R2GTx2=R52(1);R2GTy2=R52(2);R2GTz2=R52(3);
GT53=[GTx3 GTy3 GTz3];
R53=SY*RyBeta*RxAlpha*RzGamma*GT53.';
R2GTx3=R53(1);R2GTy3=R53(2);R2GTz3=R53(3);
GT54=[GTx4 GTy4 GTz4];
R54=SY*RyBeta*RxAlpha*RzGamma*GT54.';
R2GTx4=R54(1);R2GTy4=R54(2);R2GTz4=R54(3);
GT55=[GTx5 GTy5 GTz5];
R55=SY*RyBeta*RxAlpha*RzGamma*GT55.';
R2GTx5=R55(1);R2GTy5=R55(2);R2GTz5=R55(3);
GT56=[GTx6 GTy6 GTz6];
R56=SY*RyBeta*RxAlpha*RzGamma*GT56.';
R2GTx6=R56(1);R2GTy6=R56(2);R2GTz6=R56(3);

```

%6. stun

```

GT61=[GTx1 GTy1 GTz1];
R61=RyBeta*RxAlpha*SZ*RzGamma*GT61.';
R3GTx1=R61(1);R3GTy1=R61(2);R3GTz1=R61(3);
GT62=[GTx2 GTy2 GTz2];
R62=RyBeta*RxAlpha*SZ*RzGamma*GT62.';
R3GTx2=R62(1);R3GTy2=R62(2);R3GTz2=R62(3);
GT63=[GTx3 GTy3 GTz3];
R63=RyBeta*RxAlpha*SZ*RzGamma*GT63.';
R3GTx3=R63(1);R3GTy3=R63(2);R3GTz3=R63(3);
GT64=[GTx4 GTy4 GTz4];
R64=RyBeta*RxAlpha*SZ*RzGamma*GT64.';
R3GTx4=R64(1);R3GTy4=R64(2);R3GTz4=R64(3);
GT65=[GTx5 GTy5 GTz5];
R65=RyBeta*RxAlpha*SZ*RzGamma*GT65.';
R3GTx5=R65(1);R3GTy5=R65(2);R3GTz5=R65(3);
GT66=[GTx6 GTy6 GTz6];
R66=RyBeta*RxAlpha*SZ*RzGamma*GT66.';
R3GTx6=R66(1);R3GTy6=R66(2);R3GTz6=R66(3);

```

```

J2= [1 0 0 R1GTx1 R2GTx1 R3GTx1;
      0 1 0 R1GTy1 R2GTy1 R3GTy1;

```

```

0 0 1 R1GTz1 R2GTz1 R3GTz1;
1 0 0 R1GTx2 R2GTx2 R3GTx2;
0 1 0 R1GTy2 R2GTy2 R3GTy2;
0 0 1 R1GTz2 R2GTz2 R3GTz2;
1 0 0 R1GTx3 R2GTx3 R3GTx3;
0 1 0 R1GTy3 R2GTy3 R3GTy3;
0 0 1 R1GTz3 R2GTz3 R3GTz3;
1 0 0 R1GTx4 R2GTx4 R3GTx4;
0 1 0 R1GTy4 R2GTy4 R3GTy4;
0 0 1 R1GTz4 R2GTz4 R3GTz4;
1 0 0 R1GTx5 R2GTx5 R3GTx5;
0 1 0 R1GTy5 R2GTy5 R3GTy5;
0 0 1 R1GTz5 R2GTz5 R3GTz5;
1 0 0 R1GTx6 R2GTx6 R3GTx6;
0 1 0 R1GTy6 R2GTy6 R3GTy6;
0 0 1 R1GTz6 R2GTz6 R3GTz6];

```

```

H=diag([h1 h1 h1 h2 h2 h2 h3 h3 h3 h4 h4 h4 h5 h5 h5 h6 h6 h6]);
K=diag([k1 k2 k3 k4 k5 k6]);
MassLegs=(m1+m2)*(J2*(H-J1'*K*J1)*J2);

```

Dosya: BacakKutleMatrisi_B.m

```

syms m1 m2 GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
syms GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
syms alpha beta gamma h1 h2 h3 h4 h5 h6 k1 k2 k3 k4 k5 k6
syms ux1 uy1 uz1 ux2 uy2 uz2 ux3 uy3 uz3 ux4 uy4 uz4 ux5 uy5 uz5 ux6 uy6 uz6

```

```

ax1=GTx1;ay1=GTy1;az1=GTz1;ax2=GTx2;ay2=GTy2;az2=GTz2;
ax3=GTx3;ay3=GTy3;az3=GTz3;ax4=GTx4;ay4=GTy4;az4=GTz4;
ax5=GTx5;ay5=GTy5;az5=GTz5;ax6=GTx6;ay6=GTy6;az6=GTz6;

```

```

n1=[ux1 uy1 uz1]';n2=[ux2 uy2 uz2]';n3=[ux3 uy3 uz3]';
n4=[ux4 uy4 uz4]';n5=[ux5 uy5 uz5]';n6=[ux6 uy6 uz6]';
a1=[ax1 ay1 az1]';a2=[ax2 ay2 az2]';a3=[ax3 ay3 az3]';
a4=[ax4 ay4 az4]';a5=[ax5 ay5 az5]';a6=[ax6 ay6 az6]';

```

```

ROT=[cos(gamma)*cos(beta)
sin(gamma)*cos(alpha)+cos(gamma)*sin(beta)*sin(alpha)
sin(gamma)*sin(alpha)+cos(gamma)*sin(beta)*cos(alpha) ;
sin(gamma)*cos(beta) cos(gamma)*cos(alpha)+sin(gamma)*sin(beta)*sin(alpha) -
cos(gamma)*sin(alpha)+sin(gamma)*sin(beta)*cos(alpha);
-sin(beta) cos(beta)*sin(alpha) cos(beta)*cos(alpha)];

```

```

J12=cross((ROT*a1),n1);
j11=n1(1);j12=n1(2);j13=n1(3);j14=J12(1);j15=J12(2);j16=J12(3);
J22=cross((ROT*a2),n2);
j21=n2(1);j22=n2(2);j23=n2(3);j24=J22(1);j25=J22(2);j26=J22(3);
J32=cross((ROT*a3),n3);
j31=n3(1);j32=n3(2);j33=n3(3);j34=J32(1);j35=J32(2);j36=J32(3);

```

```
J42=cross((ROT*a4),n4);
j41=n4(1);j42=n4(2);j43=n4(3);j44=J42(1);j45=J42(2);j46=J42(3);
J52=cross((ROT*a5),n5);
j51=n5(1);j52=n5(2);j53=n5(3);j54=J52(1);j55=J52(2);j56=J52(3);
J62=cross((ROT*a6),n6);
j61=n6(1);j62=n6(2);j63=n6(3);j64=J62(1);j65=J62(2);j66=J62(3);
```

```
J1=[j11 j12 j13 j14 j15 j16;
    j21 j22 j23 j24 j25 j26;
    j31 j32 j33 j34 j35 j36;
    j41 j42 j43 j44 j45 j46;
    j51 j52 j53 j54 j55 j56;
    j61 j62 j63 j64 j65 j66];
```

```
J2=[1 0 0 0 0 0;
    0 1 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 cos(beta) 0 0;
    0 0 0 1 -sin(alpha);
    0 0 0 -sin(beta) 0 cos(alpha)];
```

```
H=diag([h1 h2 h3 h4 h5 h6]);
K=diag([k1 k2 k3 k4 k5 k6]);
MassLegs=(m1+m2)*(J2*(H-J1'*K*J1)*J2);
```

Dosya: BacakCoriloisMatrisi_A.m

```
syms m1 m2 GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
syms GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
syms px py pz alpha beta gamma dpx dpy dpz dalpha dbeta dgamma
syms h1 h2 h3 h4 h5 h6 k1 k2 k3 k4 k5 k6
syms ux1 uy1 uz1 ux2 uy2 uz2 ux3 uy3 uz3 ux4 uy4 uz4 ux5 uy5 uz5 ux6 uy6 uz6
```

```
Xpo=[px py pz alpha beta gamma].';
dXpo=[dpx dpy dpz dalpha dbeta dgamma].';
BacakKutleMatrisi_A.m;%Yontem A'ya göre bacakların kütle matrisi m-file
MassLegs=[ Massleg11 Massleg12 Massleg13 Massleg14 Massleg15 Massleg16;
    Massleg21 Massleg22 Massleg23 Massleg24 Massleg25 Massleg26;
    Massleg31 Massleg32 Massleg33 Massleg34 Massleg35 Massleg36;
    Massleg41 Massleg42 Massleg43 Massleg44 Massleg45 Massleg46;
    Massleg51 Massleg52 Massleg53 Massleg54 Massleg55 Massleg56;
    Massleg61 Massleg62 Massleg63 Massleg64 Massleg65 Massleg66];
```

%Yontem A'ya göre bacakların Corilois matrisi

```
for i=1:6
    for j=1:6
        for k=1:6
```

```

        a(k)=1/2*(diff(MassLegs(k,j),Xpo(i))+diff(MassLegs(k,i),Xpo(j))-
diff(MassLegs(i,j),Xpo(k)))*dXpo(k);
    end
    Vmasslegs(i,j)=a(1)+a(2)+a(3)+a(4)+a(5)+a(6);
end
end

```

Dosya: BacakCoriloisMatrisi_B.m

```

syms m1 m2 GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
syms GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
syms px py pz alpha beta gamma dpx dpy dpz dalpha dbeta dgamma
syms h1 h2 h3 h4 h5 h6 k1 k2 k3 k4 k5 k6
syms ux1 uy1 uz1 ux2 uy2 uz2 ux3 uy3 uz3 ux4 uy4 uz4 ux5 uy5 uz5 ux6 uy6 uz6

```

```

Xpo=[px py pz alpha beta gamma].';
dXpo=[dpx dpy dpz dalpha dbeta dgamma].';
BacakKutleMatrisi_B.m;%Yontem B'ye göre bacakların kütle matrisi m-file
MassLegs=[ Massleg11 Massleg12 Massleg13 Massleg14 Massleg15 Massleg16;
    Massleg21 Massleg22 Massleg23 Massleg24 Massleg25 Massleg26;
    Massleg31 Massleg32 Massleg33 Massleg34 Massleg35 Massleg36;
    Massleg41 Massleg42 Massleg43 Massleg44 Massleg45 Massleg46;
    Massleg51 Massleg52 Massleg53 Massleg54 Massleg55 Massleg56;
    Massleg61 Massleg62 Massleg63 Massleg64 Massleg65 Massleg66];

```

%Yontem B'ye göre bacakların Corilois matrisi

```

for i=1:6
    for j=1:6
        for k=1:6
            a(k)=1/2*(diff(MassLegs(k,j),Xpo(i))+diff(MassLegs(k,i),Xpo(j))-
diff(MassLegs(i,j),Xpo(k)))*dXpo(k);
        end
        Vmasslegs(i,j)=a(1)+a(2)+a(3)+a(4)+a(5)+a(6);
    end
end
end

```

Dosya: BacakYerCekimiVektor_A.m

```

syms I m1 m2 pz g L1 L2 L3 L4 L5 L6
syms GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
syms GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
syms px py pz alpha beta gamma dpx dpy dpz dalpha dbeta dgamma
syms ux1 uy1 uz1 ux2 uy2 uz2 ux3 uy3 uz3 ux4 uy4 uz4 ux5 uy5 uz5
syms ux6 uy6 uz6 J1 J2 J

```

% Jacobian matrisi J1 Yöntem A

```

J1=[ux1 uy1 uz1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 ux2 uy2 uz2 0 0 0 0 0 0 0 0 0 0 0 0 0;

```

```

0 0 0 0 0 0 ux3 uy3 uz3 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 ux4 uy4 uz4 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 ux5 uy5 uz5 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ux6 uy6 uz6];

```

```

Xpo=[px py pz alpha beta gamma].';
dXpo=[dpx dpy dpz dalpha dbeta dgamma].';
GT1=[GTx1 GTy1 GTz1].';GT2=[GTx2 GTy2 GTz2].';
GT3=[GTx3 GTy3 GTz3].';GT4=[GTx4 GTy4 GTz4].';
GT5=[GTx5 GTy5 GTz5].';GT6=[GTx6 GTy6 GTz6].';

```

% Jacobian matrisi J2 Yöntem A

```

J2=[ 1 0 0 R1GTx1 R2GTx1 R3GTx1;
0 1 0 R1GTy1 R2GTy1 R3GTy1;
0 0 1 R1GTz1 R2GTz1 R3GTz1;
1 0 0 R1GTx2 R2GTx2 R3GTx2;
0 1 0 R1GTy2 R2GTy2 R3GTy2;
0 0 1 R1GTz2 R2GTz2 R3GTz2;
1 0 0 R1GTx3 R2GTx3 R3GTx3;
0 1 0 R1GTy3 R2GTy3 R3GTy3;
0 0 1 R1GTz3 R2GTz3 R3GTz3;
1 0 0 R1GTx4 R2GTx4 R3GTx4;
0 1 0 R1GTy4 R2GTy4 R3GTy4;
0 0 1 R1GTz4 R2GTz4 R3GTz4;
1 0 0 R1GTx5 R2GTx5 R3GTx5;
0 1 0 R1GTy5 R2GTy5 R3GTy5;
0 0 1 R1GTz5 R2GTz5 R3GTz5;
1 0 0 R1GTx6 R2GTx6 R3GTx6;
0 1 0 R1GTy6 R2GTy6 R3GTy6;
0 0 1 R1GTz6 R2GTz6 R3GTz6];

```

```

ZT1=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT1);
ZT2=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT2);
ZT3=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT3);
ZT4=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT4);
ZT5=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT5);
ZT6=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT6);
ZT=[ZT1;ZT2;ZT3;ZT4;ZT5;ZT6];L=[L1;L2;L3;L4;L5;L6];

```

```

% for k=1:6
for m=1:9
leg1(m)=J1(1,m)*J2(m,k);leg2(m)=J1(2,m)*J2(m,k);leg3(m)=J1(3,m)*J2(m,k);
leg4(m)=J1(4,m)*J2(m,k);leg5(m)=J1(5,m)*J2(m,k);leg6(m)=J1(6,m)*J2(m,k);
end
L1_diff(k)=leg1(1)+leg1(2)+leg1(3)+leg1(4)+leg1(5)+leg1(6)+leg1(7)+leg1(8)+leg1(9);
L2_diff(k)=leg2(1)+leg2(2)+leg2(3)+leg2(4)+leg2(5)+leg2(6)+leg2(7)+leg2(8)+leg2(9);

```

```

L3_diff(k)=leg3(1)+leg3(2)+leg3(3)+leg3(4)+leg3(5)+leg3(6)+leg3(7)+leg3(8)+leg3
(9);
L4_diff(k)=leg4(1)+leg4(2)+leg4(3)+leg4(4)+leg4(5)+leg4(6)+leg4(7)+leg4(8)+leg4
(9);
L5_diff(k)=leg5(1)+leg5(2)+leg5(3)+leg5(4)+leg5(5)+leg5(6)+leg5(7)+leg5(8)+leg5
(9);
L6_diff(k)=leg6(1)+leg6(2)+leg6(3)+leg6(4)+leg6(5)+leg6(6)+leg6(7)+leg6(8)+leg6
(9);
End
L_diff=[L1_diff;L2_diff;L3_diff;L4_diff;L5_diff;L6_diff];

```

% Yer Çekimi Vektörünün Elde Edilmesi

```

for k=1:6
    for i=1:3
        a(i)=(m1+m2)*g*(I*((1/L(2*i))^2*(L_diff(2*i,k)))+(1/L(2*i-
1))^2*(L_diff(2*i,k)))*(pz+ZT(i))...
        +(m1+m2)*g*(I*((1/L(2*i)))+(1/L(2*i-
1)))+2*m2/(m1+m2))*diff((pz+ZT(i)),Xpo(k));
    end
    Glegs(k)=a(1)+a(2)+a(3);
end

```

Dosya: BacakYerCekimiVektor_B.m

```

syms I m1 m2 pz g
syms L1 L2 L3 L4 L5 L6
syms GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
syms GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
syms px py pz alpha beta gamma
syms dpx dpy dpz dalpha dbeta dgamma
syms ux1 uy1 uz1 ux2 uy2 uz2 ux3 uy3 uz3 ux4 uy4 uz4 ux5 uy5 uz5 ux6 uy6 uz6
J1 J2 J

```

% Jacobian matrisi J1 Yöntem B

```

J1=[j11 j12 j13 j14 j15 j16;
    j21 j22 j23 j24 j25 j26;
    j31 j32 j33 j34 j35 j36;
    j41 j42 j43 j44 j45 j46;
    j51 j52 j53 j54 j55 j56;
    j61 j62 j63 j64 j65 j66];

```

% Jacobian matrisi J2 Yöntem B

```

J2=[1 0 0 0 0 0;
    0 1 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 cos(beta) 0 0;

```

```

0 0 0 1 -sin(alpha);
0 0 0 -sin(beta) 0 cos(alpha)];

RxAlpha=[1 0 0;0 cos(alpha) -sin(alpha);0 sin(alpha) cos(alpha)];
RyBeta= [cos(beta) 0 sin(beta);0 1 0;-sin(beta) 0 cos(beta)];
RzGamma= [cos(gamma) -sin(gamma) 0;sin(gamma) cos(gamma) 0;0 0 1];

Xpo=[px py pz alpha beta gamma].';
dXpo=[dpx dpy dpz dalpha dbeta dgamma].';
GT1=[GTx1 GTy1 GTz1].';
GT2=[GTx2 GTy2 GTz2].';
GT3=[GTx3 GTy3 GTz3].';
GT4=[GTx4 GTy4 GTz4].';
GT5=[GTx5 GTy5 GTz5].';
GT6=[GTx6 GTy6 GTz6].';

ZT1=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT1);
ZT2=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT2);
ZT3=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT3);
ZT4=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT4);
ZT5=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT5);
ZT6=[0 0 1]*(RzGamma'*RxAlpha'*RyBeta'*GT6);
ZT=[ZT1;ZT2;ZT3;ZT4;ZT5;ZT6];
L=[L1;L2;L3;L4;L5;L6];

for k=1:6
    for m=1:6
        leg1(m)=J1(1,m)*J2(m,k);leg2(m)=J1(2,m)*J2(m,k);leg3(m)=J1(3,m)*J2(m,k);
        leg4(m)=J1(4,m)*J2(m,k);leg5(m)=J1(5,m)*J2(m,k);leg6(m)=J1(6,m)*J2(m,k);
    end
    L1_diff(k)=leg1(1)+leg1(2)+leg1(3)+leg1(4)+leg1(5)+leg1(6);
    L2_diff(k)=leg2(1)+leg2(2)+leg2(3)+leg2(4)+leg2(5)+leg2(6);
    L3_diff(k)=leg3(1)+leg3(2)+leg3(3)+leg3(4)+leg3(5)+leg3(6);
    L4_diff(k)=leg4(1)+leg4(2)+leg4(3)+leg4(4)+leg4(5)+leg4(6);
    L5_diff(k)=leg5(1)+leg5(2)+leg5(3)+leg5(4)+leg5(5)+leg5(6);
    L6_diff(k)=leg6(1)+leg6(2)+leg6(3)+leg6(4)+leg6(5)+leg6(6);
end
L_diff=[L1_diff;L2_diff;L3_diff;L4_diff;L5_diff;L6_diff];

for k=1:6
    for i=1:3
        a(i)=(m1+m2)*g*(I*((1/L(2*i))^2*(L_diff(2*i,k)))+(1/L(2*i-
1))^2*(L_diff(2*i,k))))*(pz+ZT(i))...
        +(m1+m2)*g*(I*((1/L(2*i)))+(1/L(2*i-
1)))+2*m2/(m1+m2))*diff((pz+ZT(i)),Xpo(k));
    end
    Glegs(k)=a(1)+a(2)+a(3);
end

```

Dosya: RobotParametersSetup.m

```
clear all;close all;clc;
start_time=0;samp_time=0.001;sim_time=2;
deg2rad = pi/180;x_axis = [1 0 0];y_axis = [0 1 0];z_axis = [0 0 1];

% Hareketli ve sabit platformun bağlantı noktalarının Hesabı

pos_base = [];
pos_top = [];
alpha_b = 16.41456206/2*deg2rad; % Sabit platform için offset açısı alpha_t =
25.68141134/2*deg2rad; % Üst platform için offset açısı height = 0.25; % Home
konumundaki yükseklik
radius_b = 0.29326616983/2; % Sabit platformun yarıçapı
radius_t = 0.18867975191/2; % Hareketli platformun yarıçapı

for i = 1:3,
    % Sabit platform noktaları
    angle_m_b = (2*pi/3)*(i-1) - alpha_b;
    angle_p_b = (2*pi/3)*(i-1) + alpha_b;
    pos_base(2*i-1,:) = radius_b*[cos(angle_m_b), sin(angle_m_b), 0.0];
    pos_base(2*i,:) = radius_b*[cos(angle_p_b), sin(angle_p_b), 0.0];

    % Hareketli platform noktaları
    angle_m_t = (2*pi/3)*(i-1) - alpha_t + 2*pi/6;
    angle_p_t = (2*pi/3)*(i-1) + alpha_t + 2*pi/6;
    pos_top(2*i-1,:) = [radius_t*cos(angle_m_t), radius_t*sin(angle_m_t), height];
    pos_top(2*i,:) = [radius_t*cos(angle_p_t), radius_t*sin(angle_p_t), height];
end

pos_top = [pos_top(6,:); pos_top(1:5,:)];
% GTi'lerin hesaplanması
body_pts = pos_top' - height*[zeros(2,6);ones(1,6)];
% Bacak vektörleri
legs = pos_top - pos_base;
leg_length = [ ];
leg_vectors = [ ];

for i = 1:6,
    leg_length(i) = norm(legs(i,:));
    leg_vectors(i,:) = legs(i,:) / leg_length(i);
end

% Dönel ve silindirik eksenlerin hesaplanması

for i = 1:6,
    rev1(i,:) = cross(leg_vectors(i,:), z_axis);
    rev1(i,:) = rev1(i,:) / norm(rev1(i,:));
    rev2(i,:) = - cross(rev1(i,:), leg_vectors(i,:));
    rev2(i,:) = rev2(i,:) / norm(rev2(i,:));
end
```



```

cyl1(i,:) = leg_vectors(i,:);
rev3(i,:) = rev1(i,:);
rev4(i,:) = rev2(i,:);
end

```

% Koordinat sistemleri

```

lower_leg = struct('origin', [0 0 0], 'rotation', eye(3), 'end_point', [0 0 0]);
upper_leg = struct('origin', [0 0 0], 'rotation', eye(3), 'end_point', [0 0 0]);

```

```

for i = 1:6,
    lower_leg(i).origin = pos_base(i,:) + (3/8)*legs(i,:);
    lower_leg(i).end_point = pos_base(i,:) + (3/4)*legs(i,:);
    lower_leg(i).rotation = [rev1(i:)', rev2(i:)', cyl1(i:)]';
    upper_leg(i).origin = pos_base(i,:) + (1-3/8)*legs(i,:);
    upper_leg(i).end_point = pos_base(i,:) + (1/4)*legs(i,:);
    upper_leg(i).rotation = [rev1(i:)', rev2(i:)', cyl1(i:)]';
end

```

% Atalet ve kütle hesabı

```

top_thickness = 0.015;
base_thickness = 0.035;
inner_radius = 0.017/2;
outer_radius = 0.036/2;
density = 2700;

```

%Bacak atalet ve kütle hesabı

```

[lower_leg_mass, lower_leg_inertia] = inertiaCylinder(density, ...
    0.75*leg_length(1),outer_radius, inner_radius);
[upper_leg_mass, upper_leg_inertia] = inertiaCylinder(density, ...
    0.75*leg_length(1),inner_radius, 0);

```

% Hareketli ve sabit plakaların kütle atalet hesabı

```

[top_mass, top_inertia] = inertiaCylinder(density, ...
    top_thickness, radius_t, 0);
[base_mass, base_inertia] = inertiaCylinder(density, ...
    base_thickness,radius_b, 0);

```

Dosya: InitialParameters_with_motor.m

```

global mu Ix Iy Iz
global m1 m2
global GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3
global GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6
global g l1 l2
global I
global n p Js Jm Bs Bm

```

```

Kb=2.730e-3;%back EMF sabiti V/rpm
Kt=26.10e-3;%Tork sabiti Nm/A
L=265e-6;% indüktans H
R=7.10;% Armatür direnci ohm
n=1;%Bağlama oranı
p=1e-3; % ball-screw pitch m – Vidali bilye dönüş oranı
Js= 0.00028573e-3; %Kg.m^2 %Vidali bilye atalet momenti
Jm=0.58e-6; % Kg.m^2 %Mass moment of inertia of the motor shaft
Bs= 0.123e-3; % N.s/rad % Vidali bilyenin Viskoz sürtünme katsayısı
Bm= 0.00196e-3; %N.s/rad Motor milinin Viskoz sürtünme katsayısı

```

```

mu=top_mass;%Kg %Hareketli platform kütlesi
Ix=top_inertia(1); %Kg.m^2 %Hareketli platformun atalet momenti
Iy=top_inertia(5);%Kg.m^2 % Hareketli platformun atalet momenti
Iz=top_inertia(9);%Kg.m^2 % Hareketli platformun atalet momenti
g=9.81;
m1=lower_leg_mass;%Bacakların hareketli kısmın kütlesi
m2=upper_leg_mass;%Bacakların sabit kısmın kütlesi
l1=220e-3;%m %Bacakların sabit kısmın uzunluğu
l2=50e-3; %m %Bacakların hareketli kısmın uzunluğu
ksi=80e-3;%m % BGi ile alt platform arasındaki uzunluk
I=(1/(m1+m2))*(ksi*l1-(1/2)*m2*l2);

```

```

GTx1=body_pts(1);GTy1=body_pts(2);GTz1=body_pts(3);
GTx2=body_pts(4);GTy2=body_pts(5);GTz2=body_pts(6);
GTx3=body_pts(7);GTy3=body_pts(8);GTz3=body_pts(9);
GTx4=body_pts(10);GTy4=body_pts(11);GTz4=body_pts(12);
GTx5=body_pts(13);GTy5=body_pts(14);GTz5=body_pts(15);
GTx6=body_pts(16);GTy6=body_pts(17);GTz6=body_pts(18);%0;
UMLegs=[m1 m2 GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3 GTx4
GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6];
UMForward=[m1 m2 mu g I Ix Iy Iz GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3
GTy3 GTz3 GTx4 GTy4 GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6];
UMJacobian=[GTx1 GTy1 GTz1 GTx2 GTy2 GTz2 GTx3 GTy3 GTz3 GTx4 GTy4
GTz4 GTx5 GTy5 GTz5 GTx6 GTy6 GTz6];

```

EK-B

Dosya: PIDTuningPSO.m

```
clc;
```

```
%PSO Algoritmasına ait parametreler
```

```
inertia=0.76;fi1=2.05;fi2=2.05;r1=rand;r2=rand;
```

```
%Optimizasyon parametrelerin belirlenmesi
```

```
swarm_size=10;stpmax=30;stp=1;load swarmmax;load swarmvalues;
```

```
%Simülasyon için Simülink dosyasının ismi
```

```
model_name = 'StewartDynamics_Control_PID';
```

```
%-----Algoritmanın Balatılması -----
```

```
open(model_name); % Simülink Modelini çağırma
```

```
%V, sfirlardan oluşan 10x18'lik başlangıç hız matrisi
```

```
for i=1:swarm_size
```

```
    V(i,1:18,1)=0;
```

```
end
```

```
% Sürüdeki parçacıkların(her bir eklem için optimize edilecek PID kontrolörlere ait parametreler)simülink modeline sırayla gönderilmesi
```

```
while(stp<=stpmax)
```

```
    % Parçacıkların sisteme gönderilmesi
```

```
    for sw=1:swarm_size
```

```
        P(sw,:)=swarm(sw,:,stp);
```

```
    % PID'lerin Kp parametre değerlerinin değiştirilmesi
```

```
set_param([model_name'/PID/PID_CONTROLLER/First_PID_Parameters/Kp'],'Gain', num2str(swarm(sw,1,stp)) );
```

```
set_param([model_name'/PID/PID_CONTROLLER/Second_PID_Parameters/Kp'],'Gain', num2str(swarm(sw,4,stp)) );
```

```
set_param([model_name'/PID/PID_CONTROLLER/Third_PID_Parameters/Kp'],'Gain', num2str(swarm(sw,7,stp)) );
```

```
set_param([model_name'/PID/PID_CONTROLLER/Fourth_PID_Parameters/Kp'],'Gain', num2str(swarm(sw,10,stp)) );
```

```
set_param([model_name'/PID/PID_CONTROLLER/Fifth_PID_Parameters/Kp'],'Gain', num2str(swarm(sw,13,stp)) );
```

```
set_param([model_name'/PID/PID_CONTROLLER/Sixth_PID_Parameters/Kp'],'Gain', num2str(swarm(sw,16,stp)) );
```

% PID'lerin Ki parametre değerlerinin değiştirilmesi

```
set_param([model_name'/PID/PID_CONTROLLER/First_PID_Parameters/Ki'],'Gain',num2str(swarm(sw,2,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Second_PID_Parameters/Ki'],'Gain', num2str(swarm(sw,5,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Third_PID_Parameters/Ki'],'Gain', num2str(swarm(sw,8,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Fourth_PID_Parameters/Ki'],'Gain', num2str(swarm(sw,11,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Fifth_PID_Parameters/Ki'],'Gain', num2str(swarm(sw,14,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Sixth_PID_Parameters/Ki'],'Gain', num2str(swarm(sw,17,stp)) );
```

% PID'lerin Kd parametre değerlerinin değiştirilmesi

```
set_param([model_name'/PID/PID_CONTROLLER/First_PID_Parameters/Kd'],'Gain', num2str(swarm(sw,3,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Second_PID_Parameters/Kd'],'Gain', num2str(swarm(sw,6,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Third_PID_Parameters/Kd'],'Gain', num2str(swarm(sw,9,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Fourth_PID_Parameters/Kd'],'Gain', num2str(swarm(sw,12,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Fifth_PID_Parameters/Kd'],'Gain', num2str(swarm(sw,15,stp)) );
set_param([model_name'/PID/PID_CONTROLLER/Sixth_PID_Parameters/Kd'],'Gain', num2str(swarm(sw,18,stp)) );
```

```
sim(model_name); % Simülink modelini çalıştırma
pause(0.02);
```

%Maliyet fonksiyon değerini hesaplama (o anki parçacığın)

```
t=start_time:samp_time:sim_time;
CF=0; %Maliyet Fonksiyon Değeri
err_tot=0; %Toplam Hata
nn=length(t);

for i=1:nn
err_tot=err_tot+sqrt(errors(1,(i))^2+errors(2,(i))^2+errors(3,(i))^2+errors(4,(i))^2+errors(5,(i))^2+errors(6,(i))^2);
end

CF=(1/nn)*err_tot ;CFSW(sw,stp)=CF;
end
```

```

% G, Sürüde bulunan en iyi konumu hesaplama (Global Best)

[a,ind]=min(CFSW(:,:));
[ss,stps]=min(a);
G=swarm(ind(stps),:,stps);

% pbl(j), j.ci Parçacığın en iyi konumunu hesaplama(Personal Best)

for j=1:swarm_size
[b,indx(j)]=min(CFSW(j,:));
end

for i=1:swarm_size
plb(i,:)=swarm(i,:,indx(i));
end

% Tüm parçacıkların hız ve konumunu güncelleme

for i=1:swarm_size
V(i,:,stp+1)=inertia*rand*(V(i,:,stp))+fi1*rand*(plb(i,:)-
swarm(i,:,stp))+fi2*rand*(G-swarm(i,:,stp));
end

for i=1:swarm_size
swarm(i,:,stp+1)=swarm(i,:,stp)+V(i,:,stp+1);
% Parçacıkların alt-üst sınır kontrolü
swarm(i,:,stp+1)=min(swarmmax(1,:,1),swarm(i,:,stp+1));
swarm(i,:,stp+1)=max(swarmmin(1,:,1),swarm(i,:,stp+1));
end

fprintf('Iteration: %d, Best Cost: %f\n',stp,a(stp))
CFSW
stp=stp+1;
end%---end while ((iter < itermax) ..

%Optimizasyon sonucu en iyi parçacık ve buna ait maliyet değerini yazdırılması

fprintf(1,'The Number Of The Best Particle is %d in swarm at %dth
step\n',ind(stps),stps)
fprintf(1,'The Best Cost so far is %f at %dth step\n',ss,stps)

Dosya: FOPIDTuningPSO.m

clc;

%PSO Algoritmasına ait parametreler

inertia=0.76;fi1=2.05;fi2=2.05;r1=rand;r2=rand;

%Optimizasyon parametrelerinin belirlenmesi

```

```

swarm_size=10;stpmax=30;stp=1;load swarmmax;load swarmvalues;

%Simülasyon için Simülink dosyasının ismi

model_name = 'StewartDynamics_Control_FOPID';

%-----Algoritmanın Balatılması -----

open(model_name); % Simülink Modelini çağırma

%V, sfirlardan oluşan 10x30'luk başlangıç hız matrisi

for i=1:swarm_size
    V(i,1:30,1)=0;
end

% Sürüdeki parçacıkların(her bir eklem için optimize edilecek FOPID kontrolörlere
ait parametreler)simülink modeline sırayla gönderilmesi

While(stp<=stpmax)

% Parçacıkların sisteme gönderilmesi

for sw=1:swarm_size
P(sw,:)=swarm(sw,.,stp);

%Kontrolörlerin yaklaşık transfer fonksiyonların hesaplanması Kontrolör-1 için
FOPID parametreleri

tfpid1=nipid(P(sw,1),P(sw,2),P(sw,3),P(sw,4),P(sw,5),[0.01
600],5,'crone','mcltime','all');[num1,denum1]=tfdata(tfpid1,'v');

%Kontrolör-2 için FOPID parametreleri

tfpid2=nipid(P(sw,6),P(sw,7),P(sw,8),P(sw,9),P(sw,10),[0.01
600],5,'crone','mcltime','all');[num2,denum2]=tfdata(tfpid2,'v');

%Kontrolör-3 için FOPID parametreleri

tfpid3=nipid(P(sw,11),P(sw,12),P(sw,13),P(sw,14),P(sw,15),[0.01
600],5,'crone','mcltime','all');[num3,denum3]=tfdata(tfpid3,'v');

%Kontrolör-4 için FOPID parametreleri

tfpid4=nipid(P(sw,16),P(sw,17),P(sw,18),P(sw,19),P(sw,20),[0.01
600],5,'crone','mcltime','all');[num4,denum4]=tfdata(tfpid4,'v');

%Kontrolör-5 için FOPID parametreleri

tfpid5=nipid(P(sw,21),P(sw,22),P(sw,23),P(sw,24),P(sw,25),[0.01
600],5,'crone','mcltime','all');[num5,denum5]=tfdata(tfpid5,'v');

```

%Kontrolör-6 için FOPID parametreleri

```
tfpid6=nipid(P(sw,26),P(sw,27),P(sw,28),P(sw,29),P(sw,30),[0.01  
600],5,'crone','mcltime','all');[num6,denum6]=tfdata(tfpid6,'v');  
sim(model_name); % Simülink modelini çalıştırma  
pause(0.02);
```

%Maliyet fonksiyon değerini hesaplama (o anki parçacığın)

```
t=start_time:samp_time:sim_time;  
CF=0; %Maliyet Fonksiyon Değeri  
err_tot=0; %Toplam Hata  
nn=length(t);  
for i=1:nn  
err_tot=err_tot+sqrt(errors(1,(i))^2+errors(2,(i))^2+errors(3,(i))^2+errors(4,(i))^2+er  
rors(5,(i))^2+errors(6,(i))^2);  
end  
CF=(1/nn)*err_tot ;  
CFSW(sw,stp)=CF;  
end
```

% G, Sürüde bulunan en iyi konumu hesaplama (Global Best)

```
[a,ind]=min(CFSW(:,:));  
[ss,stps]=min(a);  
G=swarm(ind(stps),:,stps);
```

% pbl(j), j.ci Parçacığın en iyi konumunu hesaplama(Personal Best)

```
for j=1:swarm_size  
[b,indx(j)]=min(CFSW(j,:));  
end  
for i=1:swarm_size  
plb(i,:)=swarm(i,:,indx(i));  
end
```

% Tüm parçacıkların hız ve konumunu güncelleme

```
for i=1:swarm_size  
V(i,:,stp+1)=inertia*rand*(V(i,:,stp))+f1*rand*(plb(i,:)-  
swarm(i,:,stp))+f2*rand*(G-swarm(i,:,stp));  
end  
for i=1:swarm_size  
swarm(i,:,stp+1)=swarm(i,:,stp)+V(i,:,stp+1);
```

% Parçacıkların alt-üst sınır kontrolü

```
swarm(i,:,stp+1)=min(swarmmax(1,:,1),swarm(i,:,stp+1));  
swarm(i,:,stp+1)=max(swarmmin(1,:,1),swarm(i,:,stp+1));  
end  
fprintf('Iteration: %d, Best Cost: %f\n',stp,a(stp))
```

```

CFSW
stp=stp+1;
end%---end while ((iter < itemax) ..

%Optimizasyon sonucu en iyi parçacığa ait maliyet değeri ve parametrelerin
yazdırılması

fprintf(1,'The Number Of The Best Particle is %d in swarm at %dth
step\n',ind(stps),stps)
fprintf(1,'The Best Cost so far is %f at %dth step\n',ss,stps)
fprintf(1,'The Optimal Parameters of Fractional PID\n');

Dosya: BulanıkPIDTuningPSO.m

clc;

%PSO Algoritmasına ait parametreler

inertia=0.76;fi1=2.05;fi2=2.05;r1=rand;r2=rand;

%Optimizasyon parametrelerin belirlenmesi

swarm_size=10;stpmax=30;stp=1;load swarmmax;load swarmvalues;

%Simülasyon için Simülink dosyasının ismi

model_name = 'StewartDynamics_Control_FUZZYPID';

%-----Algoritmanın Balatılması -----

open(model_name); % Simülink Modelini çağırma

%V, sfirlardan oluşan 10x204'luk başlangıç hız matrisi

for i=1:swarm_size
    V(i,1:204,1)=0;
end

% Sürüdeki parçacıkların(her bir eklem için optimize edilecek Bulanık PID
kontrolörlere ait parametreler)simülink modeline sırayla gönderilmesi

while(stp<=stpmax)

% Parçacıkların sisteme gönderilmesi

for sw=1:swarm_size
P(sw,:)=swarm(sw,.,stp);

%Bulanık kontrolörlerin Giriş ve Çıkış değişkenlerine ait üyelik fonksiyon
parametrelerin değiştirilmesi

%Bulanık PID-1

```



```

for i=1:5
PIDcont1=setfis(PIDcont1,'input',1,'mf',i,'params',[P(sw,2*i-1) P(sw,2*i)]);
PIDcont1=setfis(PIDcont1,'input',2,'mf',i,'params',[P(sw,10+2*i-1) P(sw,10+2*i)]);
PIDcont1=setfis(PIDcont1,'output',1,'mf',i,'params',[P(sw,20+2*i-1) P(sw,20+2*i)]);
end

```

%Bulanik PID-2

```

for i=1:5
PIDcont2=setfis(PIDcont2,'input',1,'mf',i,'params',[P(sw,30+2*i-1) P(sw,30+2*i)]);
PIDcont2=setfis(PIDcont2,'input',2,'mf',i,'params',[P(sw,40+2*i-1) P(sw,40+2*i)]);
PIDcont2=setfis(PIDcont2,'output',1,'mf',i,'params',[P(sw,50+2*i-1) P(sw,50+2*i)]);
end

```

%Bulanik PID-3

```

for i=1:5
PIDcont3=setfis(PIDcont3,'input',1,'mf',i,'params',[P(sw,60+2*i-1) P(sw,60+2*i)]);
PIDcont3=setfis(PIDcont3,'input',2,'mf',i,'params',[P(sw,70+2*i-1) P(sw,70+2*i)]);
PIDcont3=setfis(PIDcont3,'output',1,'mf',i,'params',[P(sw,80+2*i-1) P(sw,80+2*i)]);
end

```

%Bulanik PID-4

```

for i=1:5
PIDcont4=setfis(PIDcont4,'input',1,'mf',i,'params',[P(sw,90+2*i-1) P(sw,90+2*i)]);
PIDcont4=setfis(PIDcont4,'input',2,'mf',i,'params',[P(sw,100+2*i-1)
P(sw,100+2*i)]);
PIDcont4=setfis(PIDcont4,'output',1,'mf',i,'params',[P(sw,110+2*i-1)
P(sw,110+2*i)]);
end

```

%Bulanik PID-5

```

for i=1:5
PIDcont5=setfis(PIDcont5,'input',1,'mf',i,'params',[P(sw,120+2*i-1)
P(sw,120+2*i)]);
PIDcont5=setfis(PIDcont5,'input',2,'mf',i,'params',[P(sw,130+2*i-1)
P(sw,130+2*i)]);
PIDcont5=setfis(PIDcont5,'output',1,'mf',i,'params',[P(sw,140+2*i-1)
P(sw,140+2*i)]);
end

```

%Bulanik PID-6

```

for i=1:5
PIDcont6=setfis(PIDcont6,'input',1,'mf',i,'params',[P(sw,150+2*i-1)
P(sw,150+2*i)]);
PIDcont6=setfis(PIDcont6,'input',2,'mf',i,'params',[P(sw,160+2*i-1)
P(sw,160+2*i)]);
end

```

```

PIDcont6=setfis(PIDcont6,'output',1,'mf',i,'params',[P(sw,170+2*i-1)
P(sw,170+2*i)]);
end
writefis(PIDcont1,'PIDcont1');writefis(PIDcont2,'PIDcont2');writefis(PIDcont3,'PID
cont3');writefis(PIDcont4,'PIDcont4');writefis(PIDcont5,'PIDcont5');writefis(PIDcon
t6,'PIDcont6');

```

%Bulanık PID'lerin giriş ve çıkış kazanç katsayıların değiştirilmesi

```

Ge1=P(sw,181);Gce1=P(sw,182); Gi1=P(sw,183); Gu1=P(sw,184);Ge2=P(sw,185);
Gce2=P(sw,186); Gi2=P(sw,187); Gu2=P(sw,188);Ge3=P(sw,189);Gce3=P(sw,190);
Gi3=P(sw,191); Gu3=P(sw,192); Ge4=P(sw,193); Gce4=P(sw,194); Gi4=P(sw,195);
Gu4=P(sw,196);Ge5=P(sw,197); Gce5=P(sw,198); Gi5=P(sw,199); Gu5=P(sw,200);
Ge6=P(sw,201); Gce6=P(sw,202); Gi6=P(sw,203); Gu6=P(sw,204);
sim(model_name); % Simülink modelini çalıştırma
pause(0.02);

```

%Maliyet fonksiyon değerini hesaplama (o anki parçacığın)

```

t=start_time:samp_time:sim_time;
CF=0; %Maliyet Fonksiyon Değeri
err_tot=0; %Toplam Hata
nn=length(t);
for i=1:nn
err_tot=err_tot+sqrt(errors(1,i)^2+errors(2,i)^2+errors(3,i)^2+errors(4,i)^2+er
rors(5,i)^2+errors(6,i)^2);
end
CF=(1/nn)*err_tot ;CFSW(sw,stp)=CF;
end

```

% G, Sürüde bulunan en iyi konumu hesaplama (Global Best)

```

[a,ind]=min(CFSW(:,:));[ss,stps]=min(a);G=swarm(ind(stps),:,stps);

```

% pbl(j), j.ci Parçacığın en iyi konumunu hesaplama(Personal Best)

```

for j=1:swarm_size
[b,indx(j)]=min(CFSW(j,:));
end
for i=1:swarm_size
plb(i,:)=swarm(i,:,indx(i));
end

```

% Tüm parçacıkların hız ve konumunu güncelleme

```

for i=1:swarm_size
V(i,:,stp+1)=inertia*rand*(V(i,:,stp))+f1*rand*(plb(i,:)-
swarm(i,:,stp))+f2*rand*(G-swarm(i,:,stp));
end
for i=1:swarm_size
swarm(i,:,stp+1)=swarm(i,:,stp)+V(i,:,stp+1);

```

```

% Parçacıkların alt-üst sınır kontrolü

swarm(i,:,stp+1)=min(swarmmax(1,:,1),swarm(i,:,stp+1));
swarm(i,:,stp+1)=max(swarmmin(1,:,1),swarm(i,:,stp+1));
end
fprintf('Iteration: %d, Best Cost: %f\n',stp,a(stp))
CFSW
stp=stp+1;
end%---end while ((iter < itermax) ..

%Optimizasyon sonucu en iyi parçacık ve buna ait maliyet değerinin yazdırılması

fprintf(1,'The Number Of The Best Particle is %d in swarm at %dth
step\n',ind(stps),stps);fprintf(1,'The Best Cost so far is %f at %dth step\n',ss,stps);

Dosya: BulanıkFOPIDTuningPSO.m

clc;

%PSO Algoritmasına ait parametreler

inertia=0.76;f1=2.05;f2=2.05;r1=rand;r2=rand;

%Optimizasyon parametrelerin belirlenmesi

swarm_size=10;stpmax=30;stp=1;load swarmmax;load swarmvalues;

%Simülasyon için Simülink dosyasının ismi

model_name = 'Control_FO_FUZZY_PID';

%-----Algoritmanın Başlatılması -----

open(model_name); % Simülink Modelini çağırma

%V, sfirlardan oluşan 10x216'luk başlangıç hız matrisi

for i=1:swarm_size
V(i,1:216,1)=0;
end

% Sürüdeki parçacıkların(her bir eklem için optimize edilecek Bulanık FOPID
kontrolörlere ait parametreler)simülink modeline sırayla gönderilmesi

while(stp<=stpmax)

% Parçacıkların sisteme gönderilmesi

for sw=1:swarm_size
P(sw,:)=swarm(sw,:,stp);

```

%Bulanık kontrolörlerin Giriş ve Çıkış değişkenlerine ait üyelik fonksiyon parametrelerin değiştirilmesi

%Bulanık FOPID-1

for i=1:5

```
PIDcont1=setfis(PIDcont1,'input',1,'mf',i,'params',[P(sw,2*i-1) P(sw,2*i)]);
PIDcont1=setfis(PIDcont1,'input',2,'mf',i,'params',[P(sw,10+2*i-1) P(sw,10+2*i)]);
PIDcont1=setfis(PIDcont1,'output',1,'mf',i,'params',[P(sw,20+2*i-1) P(sw,20+2*i)]);
end
```

%Bulanık FOPID-2

for i=1:5

```
PIDcont2=setfis(PIDcont2,'input',1,'mf',i,'params',[P(sw,30+2*i-1) P(sw,30+2*i)]);
PIDcont2=setfis(PIDcont2,'input',2,'mf',i,'params',[P(sw,40+2*i-1) P(sw,40+2*i)]);
PIDcont2=setfis(PIDcont2,'output',1,'mf',i,'params',[P(sw,50+2*i-1) P(sw,50+2*i)]);
end
```

%Bulanık FOPID-3

for i=1:5

```
PIDcont3=setfis(PIDcont3,'input',1,'mf',i,'params',[P(sw,60+2*i-1) P(sw,60+2*i)]);
PIDcont3=setfis(PIDcont3,'input',2,'mf',i,'params',[P(sw,70+2*i-1) P(sw,70+2*i)]);
PIDcont3=setfis(PIDcont3,'output',1,'mf',i,'params',[P(sw,80+2*i-1) P(sw,80+2*i)]);
end
```

%Bulanık FOPID-4

for i=1:5

```
PIDcont4=setfis(PIDcont4,'input',1,'mf',i,'params',[P(sw,90+2*i-1) P(sw,90+2*i)]);
PIDcont4=setfis(PIDcont4,'input',2,'mf',i,'params',[P(sw,100+2*i-1)
P(sw,100+2*i)]);
PIDcont4=setfis(PIDcont4,'output',1,'mf',i,'params',[P(sw,110+2*i-1)
P(sw,110+2*i)]);
end
```

%Bulanık FOPID-5

for i=1:5

```
PIDcont5=setfis(PIDcont5,'input',1,'mf',i,'params',[P(sw,120+2*i-1)
P(sw,120+2*i)]);
PIDcont5=setfis(PIDcont5,'input',2,'mf',i,'params',[P(sw,130+2*i-1)
P(sw,130+2*i)]);
PIDcont5=setfis(PIDcont5,'output',1,'mf',i,'params',[P(sw,140+2*i-1)
P(sw,140+2*i)]);
end
```

%Bulanık FOPID-6

```
for i=1:5
PIDcont6=setfis(PIDcont6,'input',1,'mf',i,'params',[P(sw,150+2*i-1)
P(sw,150+2*i)]);
PIDcont6=setfis(PIDcont6,'input',2,'mf',i,'params',[P(sw,160+2*i-1)
P(sw,160+2*i)]);
PIDcont6=setfis(PIDcont6,'output',1,'mf',i,'params',[P(sw,170+2*i-1)
P(sw,170+2*i)]);
end
writefis(PIDcont1,'PIDcont1');writefis(PIDcont2,'PIDcont2');writefis(PIDcont3,'PID
cont3');
writefis(PIDcont4,'PIDcont4');writefis(PIDcont5,'PIDcont5');writefis(PIDcont6,'PID
cont6');
```

%Bulanık FOPID'lerin hem giriş ve çıkış kazanç katsayıları değiştirilmesi hem de türev ve integratörün derecesinin değiştirilmesi

```
Ge1=P(sw,181); Gce1=P(sw,182); Gi1=P(sw,183); Gu1=P(sw,184);lnd1=P(sw,185);
mud1=P(sw,186);Ge2=P(sw,187);Gce2=P(sw,188);Gi2=P(sw,189);Gu2=P(sw,190);
lnd2=P(sw,191);mud2=P(sw,192);Ge3=P(sw,193);Gce3=P(sw,194); Gi3=P(sw,195);
Gu3=P(sw,196);lnd3=P(sw,197);mud3=P(sw,198);Ge4=P(sw,199);Gce4=P(sw,200);
Gi4=P(sw,201);Gu4=P(sw,202);lnd4=P(sw,203);mud4=P(sw,204);Ge5=P(sw,205);
Gce5=P(sw,206);Gi5=P(sw,207);Gu5=P(sw,208);lnd5=P(sw,209);mud5=P(sw,210);
Ge6=P(sw,211); Gce6=P(sw,212); Gi6=P(sw,213); Gu6=P(sw,214);lnd6=P(sw,215);
mud6=P(sw,216);
```

% Bulanık kontrolörlerin girişteki kesir dereceli türev ve türevin önündeki giriş katsayılı çarpımından ve aynı şekilde çıkış için yaklaşık transfer fonksiyonların elde edilmesi

%Bulanık FOPID-1 için giriş ve çıkışın yaklaşık transfer fonksiyonları

```
tfpid1D=nid(Gce1,mud1,[0.01 600],5,'crone','mcltime','all');
[num1D,denum1D]=tfdata(tfpid1D,'v');
tfpid1I=nid(Gi1,-lnd1,[0.01 600],5,'crone','mcltime','all');
[num1I,denum1I]=tfdata(tfpid1I,'v');
```

%Bulanık FOPID-2 için giriş ve çıkışın yaklaşık transfer fonksiyonları

```
tfpid2D=nid(Gce2,mud2,[0.01 600],5,'crone','mcltime','all');
[num2D,denum2D]=tfdata(tfpid2D,'v');
tfpid2I=nid(Gi2,-lnd2,[0.01 600],5,'crone','mcltime','all');
[num2I,denum2I]=tfdata(tfpid2I,'v');
```

%Bulanık FOPID-3 için giriş ve çıkışın yaklaşık transfer fonksiyonları

```
tfpid3D=nid(Gce3,mud3,[0.01 600],5,'crone','mcltime','all');
```

```
[num3D,denum3D]=tfdata(tfpid3D,'v');
tfpid3I=nid(Gi3,-lnd3,[0.01 600],5,'crone','mcltime','all');
[num3I,denum3I]=tfdata(tfpid3I,'v');
```

%Bulanık FOPID-4 için giriş ve çıkışın yaklaşık transfer fonksiyonları

```
tfpid4D=nid(Gce4,mud4,[0.01 600],5,'crone','mcltime','all');
[num4D,denum4D]=tfdata(tfpid4D,'v');
tfpid4I=nid(Gi4,-lnd4,[0.01 600],5,'crone','mcltime','all');
[num4I,denum4I]=tfdata(tfpid4I,'v');
```

%Bulanık FOPID-5 için giriş ve çıkışın yaklaşık transfer fonksiyonları

```
tfpid5D=nid(Gce5,mud5,[0.01 600],5,'crone','mcltime','all');
[num5D,denum5D]=tfdata(tfpid5D,'v');
tfpid5I=nid(Gi5,-lnd5,[0.01 600],5,'crone','mcltime','all');
[num5I,denum5I]=tfdata(tfpid5I,'v');
```

%Bulanık FOPID-6 için giriş ve çıkışın yaklaşık transfer fonksiyonları

```
tfpid6D=nid(Gce6,mud6,[0.01 600],5,'crone','mcltime','all');
[num6D,denum6D]=tfdata(tfpid6D,'v');
tfpid6I=nid(Gi6,-lnd6,[0.01 600],5,'crone','mcltime','all');
[num6I,denum6I]=tfdata(tfpid6I,'v');
sim(model_name); % Simülink modelini çalıştırma
pause(0.02);
```

%Maliyet fonksiyon değerini hesaplama (o anki parçacığın)

```
t=start_time:samp_time:sim_time;
CF=0; %Maliyet Fonksiyon Değeri
err_tot=0; %Toplam Hata
nn=length(t);
for i=1:nn
err_tot=err_tot+sqrt(errors(1,(i))^2+errors(2,(i))^2+errors(3,(i))^2+errors(4,(i))^2+errors(5,(i))^2+errors(6,(i))^2);
end
CF=(1/nn)*err_tot ;
CFSW(sw,stp)=CF;
end
```

% G, Sürüde bulunan en iyi konumu hesaplama (Global Best)

```
[a,ind]=min(CFSW(:,:));[ss,stps]=min(a);G=swarm(ind(stps),:,stps);
```

% pbl(j), j.ci Parçacığın en iyi konumunu hesaplama(Personal Best)

```
for j=1:swarm_size
[b,indx(j)]=min(CFSW(j,:));
end
for i=1:swarm_size
```

```

plb(i,:)=swarm(i,:,indx(i));
end

% Tüm parçacıkların hız ve konumunu güncelleme

for i=1:swarm_size
V(i,:,stp+1)=inertia*rand*(V(i,:,stp))+fi1*rand*(plb(i,:)-
swarm(i,:,stp))+fi2*rand*(G-swarm(i,:,stp));
end

for i=1:swarm_size
swarm(i,:,stp+1)=swarm(i,:,stp)+V(i,:,stp+1);
% Parçacıkların alt-üst sınır kontrolü
swarm(i,:,stp+1)=min(swarmmax(1,:,1),swarm(i,:,stp+1));
swarm(i,:,stp+1)=max(swarmmin(1,:,1),swarm(i,:,stp+1));
end

fprintf('Iteration: %d, Best Cost: %f\n',stp,a(stp));CFSW;stp=stp+1;

end%---end while ((iter < itermax)

%Optimizasyon sonucu en iyi parçacık ve buna ait maliyet değerinin yazdırılması

fprintf(1,'The Number Of The Best Particle is %d in swarm at %dth
step\n',ind(stps),stps);

fprintf(1,'The Best Cost so far is %f at %dth step\n',ss,stps)

```

KİŞİSEL YAYIN VE ESERLER

- [1] Bingül Z., **Karahan O.**, Fractional PID controllers tuned by evolutionary algorithms for robot trajectory control, *Turkish Journal of Electrical Engineering and Computer Sciences*, DOI:10.3906/elk-1102-1011.
- [2] Bingül Z., **Karahan O.**, Dynamic Modeling and Simulation of Stewart Platform, Editor: Küçük S., *Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization*, InTech, Rijeka, Croatia, 19-41, 2012.
- [3] Bingül Z., **Karahan O.**, A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control, *Expert Systems with Applications*, 2011, **38**, 1017-1031.
- [4] Bingül Z., **Karahan O.**, Dynamic identification of Staubli RX-60 robot using PSO and LS methods, *Expert Systems with Applications*, 2011, **38**, 4136-4149.
- [5] Bingül Z., **Karahan O.**, Tuning of Fractional PID Controllers Using PSO Algorithm for Robot Trajectory Control, *IEEE International Conference on Mechatronics, ICM*, Istanbul, Turkey, 13-15 April 2011.
- [6] **Karahan O.**, Kizir S., Bingül Z., Yüksek Hassasiyetli Stewart Platformunun Kesir Dereceli PID Yöntemiyle Konum Kontrolü ve Yörünge Takibi, *Ulusal Otomatik Kontrol Toplantısı*, Kocaeli, Türkiye, 21-23 Eylül 2010.
- [7] **Karahan O.**, Bingül Z., Stewart Platformunun Parçacık Sürü Optimizasyonlu Bulanık Mantık Kontrolü, *Tok'09, Otomatik Kontrol Ulusal Toplantısı*, İstanbul, Türkiye, 13-16 Ekim 2009.
- [8] **Karahan O.**, Bingül Z., PD-Fuzzy Controller Tuned with PSO for Robot Trajectory Control, *The 2nd IFAC International Conference on Intelligent Control Systems and Signal Processing*, İstanbul, Tukey, 21-23 September 2009.
- [9] **Karahan O.**, Bingül Z., Güllü M. K., STAUBLI RX-60 Robot İleri Kinematiğinin Yapay Sinir Ağları Kullanılarak Modellenmesi, *Otomatik kontrol ulusal Toplantısı, TOK'08*, İstanbul, Türkiye, 13-15 Kasım 2008.
- [10] **Karahan O.**, Bingül Z., Modelling and Identification of STAUBLI RX-60 Robot, *IEEE CIS-RAM 2008*, Chengdu, China, 21-24 September 2008.
- [11] **Karahan O.**, Bingül Z., STAUBLI RX-60 Robotunun Parametre Kestirimi, *Otomatik kontrol ulusal Toplantısı, TOK'07*, İstanbul, Türkiye, 5-7 Eylül 2007.

ÖZGEÇMİŞ

1976 yılında İstanbul'da doğdu. İlk ve orta öğrenimini İstanbul'da tamamladı. 1996 yılında Kocaeli Üniversitesi, Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü'nde eğitime başladı. 2001 yılında Teknik Öğretmen unvanı ile mezun oldu. 2002 ile 2004 yılları arasında Kocaeli'nde özel bir şirkette Electronic Data Processing Manager olarak görev aldı ve aynı zaman da diğer bir özel şirkette Software and Newtwork Constultant olarak görev aldı. 2007 yılında Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Bölümü'nde Yüksek Lisans öğrenimini tamamladı. Aynı yıl içerisinde doktora eğitimine, Kocaeli Üniversitesi, Elektronik ve Haberleşme Mühendisliği Bölümü'nde başladı. Halen C ve C++, Visual Programlama, Web Programlama ve Robotik uygulamaları konusun da çalışmalarda bulunmaktadır.