İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ARTIK ROBOT KOLLARININ TERS KİNEMATİK PROBLEMLERİNİN ÇÖZÜMÜ VE GERÇEKLENMESİ

YÜKSEK LİSANS TEZİ

Müh. Bülent ÜNVER

Anabilim Dalı: Disiplinlerarası Program

Programı: Mekatronik Mühendisliği

Tez Danışmanı: Doç.Dr. Hakan TEMELTAŞ

ARALIK 2004

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ARTIK ROBOT KOLLARININ TERS KİNEMATİK PROBLEMLERİNİN ÇÖZÜMÜ VE GERÇEKLENMESİ

YÜKSEK LİSANS TEZİ
Müh. Bülent ÜNVER
(518021031)

Tezin Enstitüye Verildiği Tarih : 27 Aralık 2004
Tezin Savunulduğu Tarih : 27 Ocak 2005

Tez Danışmanı : Doç.Dr. Hakan TEMELTAŞ
Diğer Jüri Üyeleri Doç.Dr. Ata MUGAN
Prof.Dr. Leyla GÖREN

ARALIK 2004

ÖNSÖZ

Tüm yaşantım boyunca bana her konuda destek olan ve mutluluğum için hiçbir fedakarlıktan kaçınmayan aileme, bu çalışma sırasında beni yönlendiren ve katkılarını hiçbir zaman esirgemeyen değerli hocam Doç. Dr. Hakan TEMELTAŞ'a, çalışma sırasında karşılaştığım sorunları aşmamda bana büyük yardımda bulunan Arş. Gör. Murat YEŞİLOĞLU'na, çalışma süresince desteklerini esirgemeyen sevgili arkadaşlarım Aykut KILIÇ'a, Hüseyin DEMİR'e, Halis TULUM'a, Erdem BALIK'a, Bilge GÜROL'a ve Koray YILMAZ'a teşekkürü bir borç bilirim.

Aralık 2004 Bülent ÜNVER

İÇİNDEKİLER

	Sayfa No
ÖNSÖZ	**ii**
KISALTMALAR	**vi**
TABLO LİSTESİ	**vii**
ŞEKİL LİSTESİ	**viii**
SEMBOL LİSTESİ	**xiv**
ÖZET	**xvi**
SUMMARY	**xvii**
1. GİRİŞ	**1**
1.1. Giriş	1
2. DİFERANSİYEL KİNEMATİK VE ARTIKLIK	**4**
2.1. Jakobian Matrisinin Hesaplanması	4
2.1.1. Açısal Hız	5
2.1.2. Lineer Hız	6
2.1.2.1. Prizmatik Eklem Durumu	7
2.1.2.2. Dönel Eklem Durumu	8
2.1.3. Açısal ve Lineer Jakobianların Birleştirilmesi	10
2.2. Kinematik Tekillik	11
2.3. Kinematik Artıklık	12
2.3.1. Artıklık Analizi	12
3. TERS KİNEMATİK PROBLEMLERİN SAYISAL ÇÖZÜMLERİ	**16**
3.1. Ters Kinematik Problemi	16
3.2. Sayısal Çözünürlük Yöntemlerine Genel Bir Bakış	17
3.2.1 Analitik Yöntemler	17
3.2.2 Jakobian Devriği Yöntemi	17
3.2.3 “Resolved Motion Rate” Yöntemi	18
3.3 Görev Denkleminin Lineerleştirilmesi	18
3.4 Tekil Değer Ayrıştırması (SVD)	19
3.4.1 Temel Uzaylar	20
3.4.2 SVD’ nin Tanımlanması	20
3.5 Lineer Sistemin Çözümü	21
3.5.1 Çözüm Uzayının Ayrıştırılması	22
3.5.2 En Küçük Kareler Yöntemi	22
3.5.3 Homojen Çözüm	23

3.5.4 Genel Çözüm 23
3.5.5 En Küçük Kareler Çözümünün Sorunu: Düzenlenme Gereksinimi 24
3.5.6 Sönümlendirilmiş En Küçük Kareler 25
3.5.7 SVD ile Sönümlendirilmiş En Küçük Kareler Tersinin Hesaplanması 27
3.5.8 Uygun Sönümleme Çarpanının Hesabı 28
3.6 İteratif İşlemin Çözüme Yakınsaması 29
3.7 Sonlandırma 30
3.8 Sonuç 30

4. TERS KİNEMATİK ÇÖZÜMDEKİ SINIRLAMALAR 31
4.1 Artık Robot Kolları 31
4.2 Görev Ayrıştırması Yaklaşımı 32
4.2.1 Görevi Öncelik Sırası ile Alt Görevlere Ayrıştırma 32
4.2.2 Temel Denklemler 32
4.2.3 İstenilen Yörünge ile Verilen İkinci Alt Görev 34
4.2.4 Ölçüt Fonksiyonu ile Verilen İkinci Alt Görev 36
4.2.5 Anlık Optimizasyon Problemi Olarak Formülize Etme 36
4.3 Tekil Noktalardan ve Mekanik Eklem Limitlerinden Sakınma Uygulamaları 37
4.3.1 Tekil Noktalardan Sakınma 37
4.3.2 Mekanik Eklem Sınırlarından Sakınma 40
4.4 Engellerden Sakınma Uygulamaları 40
4.4.1 Engellerden Sakınma Stratejisi 43
4.4.1.1 Kritik Noktaların Yerini Saptama 44
4.4.1.2 Tam Çözüm 45
4.4.1.3 Yaklaşık Çözüm 50
4.4.2 Sonuç 51

5. BENZETİMLER VE DENEY SONUÇLARI 52
5.1 Mitsubishi PA 10 Robot Kolunun Kinematik Yapısı 52
5.2 Mitsubishi PA 10 Robot Kolu İçin Sınırlamasız Ters Kinematik Çözümler ve Karşılaştırmaları 54
5.2.1 En Küçük Kareler Yöntemi Benzetim Sonuçları 54
5.2.2 Sönümlendirilmiş En Küçük Kareler Yöntemi Benzetim Sonuçları 61
5.2.2.1 Sabit Sönümlendirme Çarpanı Kullanılan Sonuçlar 61
5.2.2.2 Adaptif Sönümlendirme Çarpanı Kullanılan Sonuçlar 68
5.2.3 Jakobian Devriği Yöntemi Benzetim Sonuçları 75
5.2.4 Sınırlamasız Çözüm Karşılaştırmaları 81
5.3 Mitsubishi PA 10 Robot Kolu İçin Sınırlamalı Ters Kinematik Çözümler ve Karşılaştırmaları 83
5.3.1 Mekanik Eklem Sınırlarından Sakınma Uygulanan Benzetim Sonuçları 83

5.3.2 Engellerden Sakınma Uygulanan Benzetim Sonuçları 85
5.4 Genel Değerlendirmeler 92
6. SONUÇLAR VE GELECEK ÖNERİLERİ 93
KAYNAKLAR 95
ÖZGEÇMİŞ 105

KISALTMALAR

DH: Denavit-Hartenberg Gösterilimi
SVD: Tekil Değer Ayrıştırması

TABLO LİSTESİ

Sayfa No

Tablo 5.1 : Mitsubishi PA 10 Robot Kolunun DH Tablosu……53

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 *i* Linki Nedeniyle Uç Noktasının Hareketi……9

Şekil 2.2 Eklem Hızları Uzayı İle Uç Noktası Hızları Uzayı Arasındaki Eşleme…….14

Şekil 3.1 *C* Kontrol Eksen Takımının *y*-ekseni Boyunca Serbestçe Hareket Eden Uç Noktasının Kısmi Konum Kontrolü……...16

Şekil 3.2 İki Boyutlu Eklem Değişim Uzayında (*n=2*) Gösterilen Bir Boyutlu Görev (*m=1*) için (3.11) eşitliğinin Genel Çözümünün İki Elemanı..24

Şekil 3.3 Tepe Konumu Sınırlamasına Aykırı Hareket Etmeden *h(q)* Skaler Kriterinin Minimizasyonu……...24

Şekil 3.4 En Küçük Kareler Çözümü İle Sönümlendirilmiş En Küçük Kareler Çözümünün σ Skaler Değerinin Fonsiyonu Olarak Karşılaştırılması.26

Şekil 3.5 λ Sönümleme Çarpanının Minimum Tekil Değerin Fonksiyonu Olarak Seçimi……..29

Şekil 4.1 Üç Linkli Robot Kolu……..38

Şekil 4.2 Tekil Noktalardan Kaçınma Benzetimi……...…39

Şekil 4.3 Manipüle Edilebilirlik Ölçüsünün Yörüngesi…….40

Şekil 4.4 Bazı Engellerin Varlığında Robot Kolu Hareketi……...43

Şekil 4.5 *L* Robot Kolu Linkine Göre *O* Engel Noktasının Üç Olası Konumuna Karşılık Gelen *C* Kritik Noktasının Yeri…….44

Şekil 4.6 Bir Doğruyu İzleyen ve J_{d0} Jakobianını Kullanarak Engelden Sakınan 3 Serbestlik Dereceli Düzlemsel Robot Kolu…….46

Şekil 4.7 Engelden Uzaklığa Karşı α_v Engelden Sakınma Kazancı ve α_h Homojen Terim Kazancı…….48

Şekil 4.8 Farklı Yaklaşımlar İçin A_0 Kritik Noktasındaki Kaçınma Hızları Karşılaştırması……50

Şekil 5.1 Mitsubishi PA 10 Robot Kolunun Eksen Takımlarının Gösterimi….52

Şekil 5.2 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Eklem Açı ve Hızları…..…….54

Şekil 5.3 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..55

Şekil 5.4 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Eklem Açı ve Hızları…….……..56

Şekil 5.5 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..56

Şekil 5.6 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Eklem Açı ve Hızları……..…….57

Şekil 5.7 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..58

Şekil 5.8 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Eklem Açı ve Hızları…….……..58

Şekil 5.9 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..59

Şekil 5.10 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Eklem Açı ve Hızları…….……..60

Şekil 5.11 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..60

Şekil 5.12 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Eklem Açı ve Hızları…….61

Şekil 5.13 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları..……62

Şekil 5.14 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Eklem Açı ve Hızları…….63

Şekil 5.15 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi İle Yörünge 2 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları…….63

Şekil 5.16 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Eklem Açı ve Hızları…….64

Şekil 5.17 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..65

Şekil 5.18 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Eklem Açı ve Hızları…….65

Şekil 5.19 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..66

Şekil 5.20 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Eklem Açı ve Hızları…….67

Şekil 5.21 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları..……67

Şekil 5.22 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Eklem Açı ve Hızları…….68

Şekil 5.23 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..69

Şekil 5.24 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Eklem Açı ve Hızları…….70

Şekil 5.25 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları…….70

Şekil 5.26 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Eklem Açı ve Hızları…….71

Şekil 5.27 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..72

Şekil 5.28 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Eklem Açı ve Hızları……72

Şekil 5.29 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları…..…73

Şekil 5.30 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Eklem Açı ve Hızları…....74

Şekil 5.31 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları…..…74

Şekil 5.32 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Eklem Açı ve Hızları……...……75

Şekil 5.33 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..76

Şekil 5.34 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Eklem Açı ve Hızları…….……..76

Şekil 5.35 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..77

Şekil 5.36 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Eklem Açı ve Hızları……...……78

Şekil 5.37 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..78

Şekil 5.38 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Eklem Açı ve Hızları……..…….79

Şekil 5.39 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..80

Şekil 5.40 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Eklem Açı ve Hızları…….……..80

Şekil 5.41 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..81

Şekil 5.42 Farklı Ağırlıklandırma Katsayıları İçin Yörünge 4' deki İkinci Eklem Açılarının Zamanla Değişimi……..83

Şekil 5.43 Yörünge 4 İçin Mekanik Eklem Sınırlarından Sakınıldığında Robotun Eklem Açı ve Hızları……...……84

Şekil 5.44 Yörünge 4 İçin Mekanik Eklem Sınırlarından Sakınıldığında Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..85

Şekil 5.45 Yörünge 6 İçin Engellerden Sakınma Uygulanmayan Halde Robotun Eklem Açı ve Hızları……...……86

Şekil 5.46 Yörünge 6 İçin Engellerden Sakınma Uygulanmayan Halde Robotun Uç Noktasının İstenilen, Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları……..87

Şekil 5.47 Yörünge 6 İçin Yaklaşık Çözüm Modeli Kullanılarak Engellerden Sakınma Uygulandığında Robotun Eklem Açı ve Hızları……..…88

Şekil 5.48 Yörünge 6 İçin Yaklaşık Çözüm Modeli Kullanılarak Engellerden Sakınma Uygulandığında Robotun Uç Noktasının İstenilen, Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları…….…..88

Şekil 5.49 Sakınmasız ve Sakınmalı Çözümlerin Herhangi Bir Anına İlişkin Eklem Konfigürasyonları……...89

Şekil 5.50 Sakınmasız ve Sakınmalı Durumlara İlişkin Robotun Engele Olan Minimum Uzaklıkları……..90

Şekil 5.51 Yörünge 6 İçin Tam Çözüm Modeli Kullanılarak Engellerden Sakınma Uygulandığında Robotun Eklem Açı ve Hızları……..…91

Şekil 5.52 Yörünge 6 İçin Tam Çözüm Modeli Kullanılarak Engellerden Sakınma Uygulandığında Robotun Uç Noktasının İstenilen, Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları…….……..91

SEMBOL LİSTESİ

q	: Eklem değişkenleri vektörü
$T_n^0(q)$	: Robot kolunun uç noktası eksen takımından taban eksen takımına olan dönüşümü
$R_n^0(q)$	: Uç noktası yönelimi
$O_n^0(q)$	: Uç noktası konumu
$\dot{q}(t)$	: Eklem hızları vektörü
v_n^0	: Uç noktasının lineer hız vektörü
ω_n^0	: Uç noktasının açısal hız vektörü
J_n^0	: Uç noktasının Jakobian matrisi
$\dot{O}_n^0$	: Uç noktasının lineer hızı
J_{vi}	: J_v matrisinin *i.* nci sütunu
d_i	: *i*.nci prizmatik eklem için eklem değişkeni
J_v	: Lineer jakobian
J_ω	: Açısal jakobian
m	: İşlemsel uzayının boyutu
n	: Eklem uzayı boyutu
$x(q)$	: Görev fonksiyonu, bir uç noktasının konum ve yönelimi
W	: Kartezyen görevler için kontrol eksen takımı
$N(J)$	: Boş uzay
$\Re(J)$	: Değer uzayı
σ_i	: *i*. nci tekil değer
ε	: Kazanç çarpanı
λ	: Sönümlendirme çarpanı
$J^\dagger$	: Sözde jakobian tersi
$e(q)$	: Kalıntı hatası
y_i	: Manipülasyon vektörü
H_i	: Uygun köşegenel kazanç matrisi
K_p	: Sabit kazanç matrisi
p	: İstenilen performans ölçüt fonksiyonu
∇p	: p'nin gradyanı
k_p	: Kazanç
$\dot{x}_e$	: Uç noktasının istenilen hızı
A_0	: Bir engelin komşuluğundaki kritik nokta
v_0	: Nominal hız

α_v : Engelden sakınma kazancı

d_m : Kritik uzaklık

w_i : Ağırlık çarpanı

$\dot{\varphi}$: Keyfi Eklem Hızları

ARTIK ROBOT KOLLARININ TERS KİNEMATİK PROBLEMLERİNİN ÇÖZÜMÜ VE GERÇEKLENMESİ

ÖZET

Bu çalışmada, iteratif yaklaşımla artık robot kollarının ters kinematik probleminin çözümünde kullanılan yöntemler incelenmiş ve bu yöntemlerle elde edilen sonuçlar Mitsubishi PA 10 robot koluna uygulanmıştır. İncelenen yöntemler arasında hata başarımı en iyi olan yöntemin adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi olduğu görülmüştür. Robot kollarının hareket kabiliyetlerini geliştirmede kullanılan artık serbestlik dereceleri sayesinde, Mitsubishi PA 10 robot kolunun mekanik sınırları içinde kalarak veya konumu belirli bir engelden sakınarak farklı eklem konfigürasyonları ile belirlenen yörüngeyi takip etmesi sağlanmıştır. Engellerden sakınma uygulaması literatürde genellikle düzlemsel robotlar üzerinde gerçeklenmektedir. Bu çalışmada elde edilen sonuçlar sayesinde, üç boyutlu uzayda hareket eden robotlarda da gerçeklenmesi oldukça anlamlıdır.

SOLVING AND IMPLEMENTATION INVERSE KINEMATIC PROBLEMS OF REDUNDANT ARMS

SUMMARY

In this study, using iterative algorithms, inverse kinematic solutions of the redundant manipulators are obtained for different methods. The results are applied to Mitsubishi PA 10 manipulator. Damped least squares solution method with adaptive coefficient is seen the most successful method among tested methods because of possesing the smallest error norm in all trajectories. Thanks to the redundant degrees of freedom, redundant manipulators can be improve their motion ability. With this property, Mitsubishi PA 10 manipulator can follow the end-effector trajectories within mechanical joint limits or it can avoid from an obstacle without changing the end-effector trajectory. Results obtained in this study are for 3 dimensional space covering a different aspect than most of the examples given in the literature which are for planar robotic arms.

1 GİRİŞ

Robotik alanında, kinematik artık robotlar gittikçe artan bir şekilde kullanılmaktadır. Bu tip robotların tercih sebebi, uzayda bir noktanın konumlanması için altı serbestlik derecesi gerekirken, bu robotların daha fazla serbestlik derecesine sahip olmalarıdır. Fazla olan bu serbestlik dereceleri sayesinde, artık robotlar tekil noktalardan, mekanik eklem sınırlarından, engellerden sakınma ve belirlenen bir ölçütü optimize etme amacıyla kullanılabilirler. Bu çalışma için kinematik artık bir robot olarak Mitsubishi PA 10 seçilmiştir.

Düz kinematik problemi, herhangi bir robot modelinde eklem konfigürasyonu ile bu robotun uç noktasının kartezyen uzayındaki konum ve yöneliminin tanımlanmasıdır. Verilen uç noktası son konum ve yöneliminden uygun eklem konfigürasyonunu arama işi ise ters kinematik problemi olarak tanımlanır. Ters kinematik probleminin çözümü lineer olmayan yapısından dolayı, düz kinematik problemine göre daha karmaşık bir yapıya sahiptir.

Literatürde, ters kinematik problemi çözümünde bazı yaklaşımlar vardır. Bunlar; analitik yaklaşım [74,99,11,82,43,36,37,1], iteratif yaklaşım [98,64,24,19,13,3,17,31,81,89,71,78,15,70,103,80], optimal yaklaşım [2] ,yapay sinir ağları yaklaşımı [76,60,92,26,28,94,58,104,66,105,106], ve bulanık mantık yaklaşımıdır [4,96]. Bu çalışmada ters kinematik problemi iteratif yaklaşımla irdelenecektir.

İteratif yaklaşımın tarihsel gelişiminde Whitney [100], eklem konfigürasyonunun değişiminin ağırlıklandırılmış normunu minimize etmek için genelleştirilmiş tersi [7] kullanmıştır. Deo [25], eklem değişimlerinin mümkün olan en düşük büyüklüklerini elde etmek için sonsuzluk normunun kullanımını önermiştir. Liegeois [48], jakobian matrisinin boş uzayını kullanarak eklem uzayında ifade edilen bir ölçütün optimizasyonu için bir uzantı önermiştir. Klein ve Huang [44] konuya, sözde ters çözümün, jakobian matrisinin Tekil Değer Ayrıştırması [77] ile gösterilmesi anlamında, daha farklı bir bakış açısı getirmişlerdir. Cleary [21] ve McGhee [57]

yöntemdeki ağırlıklandırılmış optimizasyon ölçütü olan bir integrasyon katsayısından söz etmişlerdir. Hanafusa, Nakamura ve Yoshikawa [35,65] bu yöndeki çalışmaları kartezyen uzayında ifade edilen ölçütlü artıklık kullanımı ile genişletmişlerdir. Bu sayede, ikincil kartezyen görev, birincil görevi etkilemeksizin sağlanabilmiştir. Bu değişik önceliğe sahip iki görevin anlık çözünürlüğü, görev önceliği stratejisi olarak bilinir. Maciejewski ve Klein [51], Hanafusa' nın çözünürlük yöntemini, sözde ters özelliklerini kullanarak geliştirmişlerdir. Yöntem, Sciliano [85] ve Garziera [32] tarafından öncelik seviyelerinin keyfi bir değerine göre genelleştirilmiştir.

Jakobian matrisinin tekil noktalarını yönetmek, diğer bir önemli konudur. Sadece sözde ters kullanımı, tekil konfigürasyonların komşuluğundaki çözünürlük işleminde sorun oluşturan yüksek eklem artışları sonucunu doğurur. Geleneksel olarak, düzenleme yöntemleri bahsedilen kötü konumlanmış problemlerin çözümü için kullanılırlar [72]. Wampler [90] ve Nakamura [64] bağımsız olarak, sözde tersin genelleştirilmiş hali olan sönümlendirilmiş en küçük kareler tersini önermişlerdir. Bundan daha sonra Maciejewski ve Klein [52] tarafından daha ileri yöntemler önerilmiştir.

Artık robotların ters kinematiğindeki temel problem, tersi alınacak jakobian matrisinin karesel olmamasıdır. Karesel olmayan matris tersi alma, karesel matrislerden daha zor bir işlemdir ve sözde ters olarak adlandırılır. Sözde ters, tekil değer ayrıştırması (SVD) ile hesaplanabilir. Bu yöntem, matrisin tekil noktalarını ortaya çıkarabilir. Tekil noktalardan kaynaklanan süreksizliklerden kaçınmak ve sözde tersi dayanıklı bir şekilde hesaplamak için sönümlendirilmiş en küçük kareler yöntemi kullanılır [54]. Bazı durumlarda hesaplama kolaylığı açısından jakobian matrisinin tersini almak yerine devriğini almak da işe yarayabilir. Ancak bu yöntem, her zaman geçerli değildir. Bu çalışma sırasında, Mitsubishi PA 10 robot kolunun ters kinematik probleminin çözümünde sırasıyla en küçük kareler, sabit çarpanlı sönümlendirilmiş en küçük kareler, adaptif çarpanlı sönümlendirilmiş en küçük kareler ve jakobian devriği yöntemleri kullanılmış ve sonuçları karşılaştırılmıştır.

Artıklık, robot kollarının kinematik ve dinamik bağlamında, hareket kabiliyetlerini geliştirmede kullanılabilir [63,65,69,59,39,10,55]. Bahsedilen kabiliyetler, engellerden sakınma [51,22,83,91,42,30,34,44], eklem sınırlarından sakınma [47], tekil noktalardan sakınma [56], tork optimizasyonu [20,38] ve beceriklilik

optimizasyonu [45] olabilir. Bu çalışma sırasında, tekil noktalardan sakınma, eklem sınırlarından sakınma ve engellerden sakınma üzerinde durulacaktır.

Bu çalışmanın ikinci bölümünde, diferansiyel kinematikte kullanılan jakobian matrisinin nasıl hesaplanacağı, kinematik tekillik ve artıklık hakkında bilgi verilecektir. Üçüncü bölümde, ters kinematik problemlerinin çözümünde kullanılan yöntemler incelenecek; dördüncü bölümde, artıklık kullanımıyla ters kinematik çözümdeki sınırlamalardan bahsedilip, tekil noktalardan, mekanik engel sınırlarından ve engellerden sakınmada kullanılan ölçütler verilecektir. Beşinci bölümde, Mitsubishi PA 10 robot kolu için çeşitli yöntemler kullanılarak benzetimle elde edilen ters kinematik çözümleri ve karşılaştırmaları, ayrıca adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi için mekanik eklem sınırlarından ve engellerden sakınma benzetim sonuçları verilecektir. Altıncı bölümde ise bu tezde yapılan çalışmalar sonuçlandırılacak ve ileriki çalışmalara ilişkin önerilere yer verilecektir.

2 DİFERANSİYEL KİNEMATİK VE ARTIKLIK

2.1 Jakobian Matrisinin Hesaplanması

q_1'den q_n'e kadar eklem değişkenlerine sahip *n* linkli bir robot kolu düşünelim ve

$$T_n^0(q) = \begin{bmatrix} R_n^0(q) & O_n^0(q) \\ 0 & 1 \end{bmatrix} \tag{2.1}$$

$q = (q1,...,qn)^T$ eklem değişkenleri vektörünü göstermek üzere, (2.1) eşitliği robot kolunun uç noktası eksen takımından taban eksen takımına olan dönüşümünü göstersin. Robot zamanla hareket ettiğinden hem q_i eklem değişkenleri, hem de O_n^0 uç noktası konumu ve R_n^0 yönelimi zamanın fonksiyonlarıdır. Bu bölümdeki amacımız, uç noktasının lineer ve açısal hızı ile $\dot{q}(t)$ eklem hızları vektörünü ilişkilendirmektir.

$$S(\omega_n^0) = \dot{R}_n^0 (R_n^0)^T \tag{2.2}$$

eşitliği uç noktasının ω_n^0 açısal hız vektörünü tanımlasın ve

$$v_n^0 = \dot{O}_n^0 \tag{2.3}$$

uç noktasının lineer hızını göstersin. Bu noktada amacımız aşağıdaki iki eşitliği araştırmaktır.

$$v_n^0 = J_v \dot{q} \tag{2.4}$$

$$\omega_n^0 = J_\omega \dot{q} \tag{2.5}$$

Bu ifadelerdeki J_v ve J_ω matrisleri *3xn* boyutundadır. (2.4) ve (2.5) eşitliklerini birleştirecek olursak,

$$\begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = J_n^0 \dot{q} \tag{2.6}$$

eşitliğini elde ederiz. Buradaki J_n^0 terimi,

$$J_n^0 = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \tag{2.7}$$

olarak ifade edilir.

Yukarıdaki ifadedeki, J_n^0 matrisi Jakobian olarak adlandırılır. J_n^0 matrisi, *n* link sayısını göstermek üzere *6xn* boyutundadır. Şimdi, herhangi bir robot kolu için Jakobian ifadesini türetelim. Bunun için önce açısal ve lineer hızları inceleyelim.

2.1.1 Açısal Hız

$$\omega_n^0 = \omega_1^0 + R_1^0\omega_2^1 + R_2^0\omega_3^2 + R_3^0\omega_4^3 + ... + R_{n-1}^0\omega_n^{n-1} \tag{2.8}$$

eşitliğinde görüldüğü gibi, ortak bir koordinat eksen takımına göre ifade edilmiş açısal hızların vektörel olarak toplanmasından oluşur. Böylelikle uç noktasının tabana göre açısal hızını, taban eksen takımının yönelimindeki her eklemin etkidiği açısal hızları tanımlayıp bunları toplayarak, belirleyebiliriz.

Eğer *i.* nci eklem dönel ise *i*. nci eklem değişkeni q_i, θ_i 'ye eşittir ve dönüş ekseni z_{i-1} dir. ω_i^{i-1}, *i*. nci eklemin dönüşüyle oluşan, *i*.nci linkin $o_{i-1}x_{i-1}y_{i-1}z_{i-1}$ eksen takımına göre ifade edilmiş açısal hızını göstersin. Bu açısal hız (*i*-1). nci eksen takımında

$$\omega_i^{i-1} = \dot{\theta}_i z_{i-1}^{i-1} = \dot{\theta}_i k \tag{2.9}$$

eşitliği ile ifade edilir. Buradaki *k* terimi $(0,0,1)^T$ birim koordinat vektörüdür.

Eğer *i*. nci eklem prizmatik ise *i*. nci eksen takımının (*i*-1). nci eksen takımına göre hareketi bir dönüşümdür ve

$$\omega_i^{i-1} = 0 \tag{2.10}$$

dır. Bu sebepten dolayı *i*. nci eklem prizmatikse uç noktasının açısal hızı, prizmatik eklem durumunda d_i ye eşit olan q_i ye bağlı değildir.

Uç noktasının toplam açısal hızı ω_n^0, taban eksen takımında (2.8) eşitliği ile belirlenir ve

$$\omega_n^0 = \rho_1 \dot{q}_1 k + \rho_2 \dot{q}_2 R_1^0 k + \ldots + \rho_n \dot{q}_n R_{n-1}^0 k \tag{2.11}$$

$$= \sum_{i=1}^{n} \rho_i \dot{q}_i z_{i-1}^0$$

şeklinde ifade edilir. Burada ρ_i terimi *i.* nci eklem dönel ise 1 prizmatik ise 0 olur. Bu sebeple,

$$z_{i-1}^0 = R_{i-1}^0 k \tag{2.12}$$

eşitliğini elde ederiz. Bu ifadenin ilk terimi $z_0^0 = k = (0,0,1)^T$ dır.

Bu denklemde z ekseni boyunca birim vektörler için üst indisler ihmal edilmiştir. Böylelikle tamamı dünya eksen takımına göre referanslandırılmıştır. Bundan sonraki bölümlerde referans eksen takımı ile ilgili bir belirsizlik olmadığı takdirde bu kabul esas alınacaktır.

2.1.2 Lineer Hız

Uç noktasının lineer hızı $\dot{O}_n$ ile ifade edilir. Türev alma kurallarından zincir kuralı yardımıyla,

$$\dot{O}_n = \sum_{i=1}^{n} \frac{\partial O_n}{\partial q_i} \dot{q}_i \tag{2.13}$$

bulunur. J_v matrisinin *i.* nci sütununu gösteren J_{vi} terimini aşağıdaki şekilde ifade edebiliriz.

$$J_{vi} = \frac{\partial O_n^0}{\partial q_i} \tag{2.14}$$

Bu denklem, $\dot{q}_i$'nin bire, diğer $\dot{q}_j$'lerin ise sıfıra eşit olması durumunda uç noktasının lineer hızını gösterir. Başka bir deyişle, *i.* nci eklem birim hızda hareket

ettirilip diğer tüm eklemler sabit tutularak, jakobianın *i*. nci sütunu elde edilebilir. Bu aşamada, prizmatik ve dönel eklemleri ayrı ayrı inceleyelim.

2.1.2.1 Prizmatik Eklem Durumu

i. nci eklemin prizmatik olması durumunda, eklem uç noktasına sadece bir çevrim iletir. DH kuralı kullanılarak ileri yöndeki kinematik ifadesi T_n^0'i aşağıdaki gibi üç dönüşümün çarpımı şeklinde yazabiliriz.

$$\begin{bmatrix} R_n^0(q) & O_n^0(q) \\ 0 & 1 \end{bmatrix} = T_n^0 \tag{2.15}$$

$$= T_{i-1}^0 T_i^{i-1} T_n^i \tag{2.16}$$

$$= \begin{bmatrix} R_{i-1}^0 & O_{i-1}^0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_i^{i-1} & O_i^{i-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_n^i & O_n^i \\ 0 & 1 \end{bmatrix} \tag{2.17}$$

$$= \begin{bmatrix} R_n^0 & R_i^0 O_n^i + R_{i-1}^0 O_i^{i-1} + O_{i-1}^0 \\ 0 & 1 \end{bmatrix} \tag{2.18}$$

Bu eşitliklerden ise,

$$O_n^0 = R_i^0 O_n^i + R_{i-1}^0 O_i^{i-1} + O_{i-1}^0 \tag{2.19}$$

denklemini elde ederiz.

Eğer sadece *i*.nci eklemin hareket etmesine izin verilseydi, hem O_n^i hem de O_{i-1}^0 sabit olurdu. Ayrıca, *i*.nci eklem prizmatikse R_{i-1}^0 çevrim matrisi de sabittir (yine, sadece *i*.nci eklemin hareket etmesine izin verildiği varsayımıyla). Son olarak, DH kuralı yardımıyla, $O_i^{i-1} = (a_i c_i, a_i s_i, d_i)^T$ olur ve O_n^0 teriminin türevi,

$$\frac{\partial O_n^0}{\partial q_i} = \frac{\partial}{\partial d_i} R_{i-1}^0 O_i^{i-1} \tag{2.20}$$

$$= R_{i-1}^0 \frac{\partial}{\partial d_i} \begin{bmatrix} a_i c_i \\ a_i s_i \\ d_i \end{bmatrix} \tag{2.21}$$

$$= \dot{d}_i R_{i-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.22}$$

$$= \dot{d}_i z_{i-1}^0 \tag{2.23}$$

eşitliklerini verir. Bu ifadelerdeki d_i terimi, i.nci prizmatik eklem için eklem değişkenidir. Böylelikle, (yine, z eksenindeki 0 üst indisini ihmal ederek) prizmatik eklem durumu için

$$J_{vi} = z_{i-1} \tag{2.24}$$

eşitliğini elde ederiz.

2.1.2.2 Dönel Eklem Durumu

i. nci eklemin dönel olması durumunda, $q_i = \theta_i$ olur. (2.19) denkleminden başlayalım ve $q_i = \theta_i$ olsun. R_i^0 terimi θ_i' ye göre sabit olmadığı için, aşağıdaki eşitlikleri elde ederiz.

$$\frac{\partial}{\partial \theta_i} O_n^0 = \frac{\partial}{\partial \theta_i} \left[R_i^0 O_n^i + R_{i-1}^0 O_i^{i-1} \right] \tag{2.25}$$

$$= \frac{\partial}{\partial \theta_i} R_i^0 O_n^i + R_{i-1}^0 \frac{\partial}{\partial \theta_i} O_i^{i-1} \tag{2.26}$$

$$= \dot{\theta}_i S(z_{i-1}^0) R_i^0 O_n^i + \dot{\theta}_i S(z_{i-1}^0) R_{i-1}^0 O_i^{i-1} \tag{2.27}$$

$$= \dot{\theta}_i S(z_{i-1}^0) \left[R_i^0 O_n^i + R_{i-1}^0 O_i^{i-1} \right] \tag{2.28}$$

$$= \dot{\theta}_i S(z_{i-1}^0)(O_n^0 - O_{i-1}^0) \tag{2.29}$$

$$= \dot{\theta}_i z_{i-1}^0 \times (O_n^0 - O_{i-1}^0) \tag{2.30}$$

(2.27)' deki ikinci terim ise şu şekilde türetilir:

$$R_{i-1}^{0}\frac{\partial}{\partial\theta_i}\begin{bmatrix}a_i c_i\\ a_i s_i\\ d_i\end{bmatrix}=R_{i-1}^{0}\begin{bmatrix}-a_i s_i\\ a_i c_i\\ 0\end{bmatrix}\dot{\theta}_i \tag{2.31}$$

$$=R_{i-1}^{0}S(k\dot{\theta}_i)O_i^{i-1} \tag{2.32}$$

$$=R_{i-1}^{0}S(k\dot{\theta}_i)(R_{i-1}^{0})^{T}R_{i-1}^{0}O_i^{i-1} \tag{2.33}$$

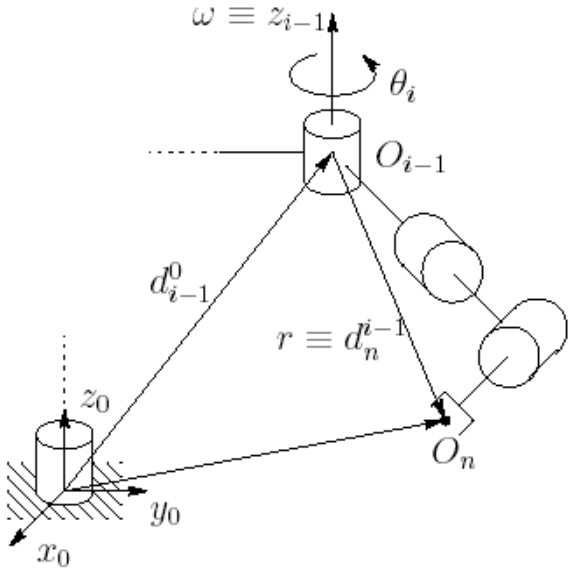
$$=S(R_{i-1}^{0}k\dot{\theta}_i)R_{i-1}^{0}O_i^{i-1} \tag{2.34}$$

$$=\dot{\theta}_i S(z_{i-1}^{0})R_{i-1}^{0}O_i^{i-1} \tag{2.35}$$

(2.32) eşitliğini aşağıdaki basit hesaplama takip eder. Sıfır üst indislerini ihmal edersek,

$$J_{vi}=z_{i-1}\times(O_n-O_{i-1}) \tag{2.36}$$

eşitliğini elde ederiz. Şekil 2.1, (2.36) eşitliğinin ikinci bir yorumunu göstermektedir. Şekilden de görüleceği gibi $O_i - O_{i-1} = r$ ve $z_{i-1}=\omega$ eşitlikleri $v=\omega\times r$ ifadesine benzerdir.



Şekil 2.1 *i* Linki Nedeniyle Uç Noktasının Hareketi

2.1.3 Açısal ve Lineer Jakobianların Birleştirilmesi

Önceki bölümde anlatıldığı gibi Jakobianın üst kısmı J_v,

$$J_v = \begin{bmatrix} J_{v1} & \dots & J_{vn} \end{bmatrix} \tag{2.37}$$

şeklinde verilmişti. Burada, J_v'nin *i*.nci sütunu J_{vi}, *i*.nci eklem dönel ise,

$$J_{vi} = z_{i-1} \times (O_n - O_{i-1}) \tag{2.38}$$

i.nci eklem prizmatik ise,

$$J_{vi} = z_{i-1} \qquad \text{dir.} \tag{2.39}$$

Jakobianın alt kısmı ise,

$$J_\omega = \begin{bmatrix} J_{\omega 1} & \dots & J_{\omega n} \end{bmatrix} \tag{2.40}$$

şeklinde verilmişti. Burada J_ω'nin i.nci sütunu $J_{\omega i}$, *i*.nci eklem dönel ise,

$$J_{\omega i} = z_{i-1} \tag{2.41}$$

i.nci eklem prizmatik ise,

$$J_{\omega i} = 0 \tag{2.42}$$

dır. Jakobianın alt ve üst yarılarını biraraya getirirsek, *n* linkli bir robot kolu için jakobian,

$$J = \begin{bmatrix} J_1 & J_2 & \dots & J_n \end{bmatrix} \tag{2.43}$$

olarak elde edilir. Burada *J*'nin *i*.nci sütunu J_i, *i*.nci eklem dönel ise,

$$J_i = \begin{bmatrix} z_{i-1} \times (O_n - O_{i-1}) \\ z_{i-1} \end{bmatrix} \tag{2.44}$$

i.nci eklem prizmatik ise,

$$J_i = \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} \tag{2.45}$$

dır. Tüm gerekli değerler ileri kinematikten elde edilebileceği için, yukarıdaki formüller herhangi bir robot kolu için jakobian matrisini basitçe tanımlar. Jakobian hesabı için gerekli olan nicelikler sadece z_i birim vektörleri ve $O_1,...,O_n$ koordinat orijinleridir. O_i T_i^0 'nin dördüncü sütununun ilk üç elemanı ile verilirken, taban eksen takımına w.r.t olan z_i için koordinatlar, T_i^0 'ın üçüncü sütununun ilk üç elemanı ile verilir. Bu yüzden, yukarıdaki formüllere göre jakobianı hesaplamak için T matrislerinin sadece üçüncü ve dördüncü sütunları yeterlidir.

Yukarıdaki yöntem, sadece uç noktasının hızını hesaplamada değil robot kolunun herhangi bir noktasının hızını hesaplamada da kullanılır.

2.2 Kinematik Tekillik

Robot kolunun, $\dot{q}$ eklem hızları vektörü ile $v = \begin{bmatrix} \dot{p}^T & \omega^T \end{bmatrix}^T$ uç noktası hızları arasında lineer ilişkiyi aşağıdaki diferansiyel kinematik eşitliğindeki jakobian terimi gösterir. (Eşitlik 2.6)

$$v = J(q)\dot{q} \tag{2.46}$$

Jakobian terimi genel durumda, q konfigürasyonunun bir fonksiyonudur. J teriminin rankının eksik hale geldiği konfigürasyonlar kinematik tekil noktalar olarak adlandırılır. Aşağıdaki nedenlerden dolayı bir robot kolunun tekil noktalarını bulmak önemlidir:

a) Tekil noktalar, yapının hareket kabiliyetinin azaldığı konfigürasyonları gösterir. Bu durumda, uç noktasına keyfi bir hareket yaptırmak mümkün değildir.

b) Yapı tekil noktada iken, ters kinematik problemi için sonsuz çözüm mevcuttur.

c) Tekil noktalara komşu bölgelerde, işlemsel uzaydaki küçük hızlar eklem uzayında büyük hızlara neden olabilir.

Tekil noktalar aşağıdaki gibi sınıflandırılabilir:

- Robot kolu gergin ya da içeri çekilmiş durumda ise, sınır tekil noktaları oluşur. Robot kolu ulaşılabilir çalışma uzayının sınırlarında çalıştırılmazsa, bu tekil noktalar gerçek bir sorun oluşturmaz.

Hareketin, ulaşılabilir çalışma uzayının içinde, iki ya da daha fazla eksenin aynı hizada olmasını gerektiren veya özel uç noktası konfigürasyonları ile erişildiği durumlarda dahili tekil noktalar oluşur. Sınır tekil noktalarından farklı olarak işlemsel uzayın içinde planlanan bir yol için ulaşılabilir çalışma uzayının herhangi bir yerinde karşılaşılabilen bu tip tekil noktalar önemli bir sorun oluşturur.

2.3 Kinematik Artıklık

Bir robot kolunun devingenlik derecesi verilen görevi tanımlamaya yetecek değişken sayısından fazla ise, bu robot koluna kinematik olarak artık denir. Esas olarak robot kolu, işlemsel uzayının boyutu eklem uzayı boyutundan küçükse, ($m < n$) artıkdır. Buna rağmen, artıklık kavramı robot koluna verilen göreve bağlıdır. Bir robot kolu bir göreve göre artık iken bir diğerine göre olmayabilir. Hatta $m=n$ durumunda, $r < n$ olmak üzere özel bir görev için işlemsel uzayın sadece r elemanı ile ilgileniliyorsa, robot kolu işlevsel artıkdır.

Bu noktada, artık robotların neden kullanıldığı akla gelebilir. Bunun nedeni, artıklığın robot kolunun hareketine beceri ve çok yönlülük sağlamasıdır. Bu duruma tipik bir örnek olarak, yedi serbestlik derecesi ile insan kolu verilebilir. Bunların, parmaklardaki değişkenlik derecesi düşünülmezse, üç tanesi omuzda, bir tanesi dirsekte ve üç tanesi de bilektedir. Bu robot kolu esasında artıkdır. Taban, el konum ve yönelimi sabit tutulursa, var olan ilave hareket kabiliyeti derecesi sayesinde, dirsek hareket edebilir. Bu sayede, örneğin çalışma uzayındaki engellerden sakınması mümkün olabilir, ya da artık robot kolundaki eklemlerden biri mekanik limitlerine ulaşsa bile, diğer eklemler verilen uç noktası hareketinin uygulanmasına izin verebilirler.

2.3.1 Artıklık Analizi

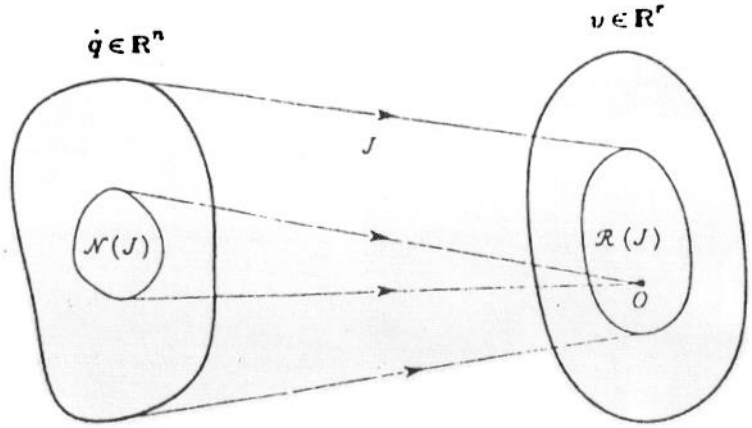
Artıklık, yapının n hareket kabiliyeti derecesi, m işlemsel uzay değişkeni sayısı ve verilen görevi belirlemek için gereken r işlemsel uzay değişkeni sayısı ile ilişkilidir.

Sistematik bir artıklık analizi yapmak istersek, düz kinematik yerine diferansiyel kinematik üzerinde durmalıyız. Bu amaçla (2.46) eşitliğini yorumlayacak olursak; diferansiyel kinematik, eklem hız vektörünün *n* elemanı ile ilgilenilen özel görevin *v* hız vektörünün $r < m$ elemanını ilişkilendirir diyebiliriz. Bu noktayı aydınlatmak için üç linkli düzlemsel robot kolunu düşünelim. Bu robot, esasında artık değildir ($n=m=3$) ve jakobian terimi üç adet boş satıra sahiptir. Eğer görev ω_z' yi ($r=2$) belirlemiyorsa, robot kolu işlevsel artık hale dönüşür.

Farklı olarak, sadece konum değişkenlerinin dikkate alındığı ($n=m=3$) antromorfik robot kolunu ele alalım. Robot kolu için düzlemsel bir görev belirlenmişse, bu robot kolunun jakobianının 6 satırından sadece 3'ü lineer bağımsız olduğu için, robot kolu ne esas olarak artıkdır ne de işlevsel artık hale dönüşür. Bu durumda, görevde uç noktasının lineer hızının üç elamanı üzerinde sınırlamalar vardır.

Bu durumdaki diferansiyel kinematik eşitliği düşünürsek (2.46)'daki gibi $v = J(q)\dot{q}$ yazılabilir. Bu yeni durumda, *v* ilgilenilen özel görev için (rx*1*) boyutunda uç noktası hız vektörünü, *J* (rxn) boyutundaki jakobian matrisini, $\dot{q}$ ise (nx*1*) boyutunda eklem hızları vektörünü gösterir. $r<n$ ise robot kolu kinematik olarak artıktır ve (n-r) adet artık devingenlik derecesi vardır.

Jakobian, eklem hızları uzayından uç noktası hızları uzayına lineer bir eşleme tanımlar. Genel durumda, bu ilişki konfigürasyonun bir fonksiyonudur. Buna rağmen, diferansiyel kinematik bağlamında jakobian sabit bir matris kabul edilir. Çünkü verilen bir duruş için anlık hız eşlemesi ile ilgilenilmektedir. Eşleme aşağıdaki Şekil 2.2'de görülebilir.



Şekil 2.2 Eklem Hızları Uzayı İle Uç Noktası Hızları Uzayı Arasındaki Eşleme

(2.46) eşitliğindeki ilişki eşlemenin değer uzayı ve boş uzayları açısından karakterize edilebilir.

- *J*'nin değer uzayı, verilen robot kolu duruşunda eklem hızları tarafından üretilebilen uç noktası hızlarının $\Re^r$ uzayında, $\Re(J)$'nin bir alt uzayıdır.
- *J*'nin boş uzayı, verilen robot kolu duruşunda herhangi bir uç noktası hızı üretmeyen eklem hızlarının $\Re^n$ uzayında, $\mathrm{N}(J)$'nin bir alt uzayıdır.

Jakobian tam ranklı ise,

$$\dim(\Re(J)) = r \qquad\qquad \dim(\mathrm{N}(J)) = n - r$$

dır ve *J*' nin değer uzayı tüm $\Re^r$ uzayını kapsar. Jakobian tekil noktalara düşerse, boş uzayının boyutu artmasına rağmen değer uzayının boyutu azalır. Bu durum, *J* matrisinin rankından bağımsız olarak aşağıdaki ilişkiye dayanır.

$$\dim(\Re(J)) + \dim(\mathrm{N}(J)) = n$$

Artık bir robot kolu için $\mathrm{N}(J) \neq 0$ alt uzayının varlığı, artık devingenlik derecelerini yönetmek için sistematik tekniklerin belirlenmesine müsaade eder. Bu amaçla, $\dot{q}^*$ (2.46) eşitliği için bir çözüm göstersin ve *P (nxn)* lik bir matris olsun. Böylelikle,

$$\Re(P) \equiv \mathrm{N}(J)$$

olur. Bu durumda $\dot{q}_0$ (2.46) için keyfi bir çözüm olmak üzere, eklem hızları vektörü

$$\dot{q} = \dot{q}^{*} + P\dot{q}_0 \tag{2.47}$$

dır.

Bu eşitliğin her iki tarafını soldan J ile çarparsak, herhangi bir $\dot{q}_0$ için $JP\dot{q}_0 = 0$ olduğundan,

$$J\dot{q} = J\dot{q}^{*} + JP\dot{q}_0 = J\dot{q}^{*} = v$$

olur. Bu sonuç, artıklık çözünürlüğü için temel bir öneme sahiptir. (2.47) tipindeki bir çözüm, artık devingenlik derecelerini avantajlı bir şekilde kullanmak için $\dot{q}_0$ eklem hızları vektörünün keyfi seçilebileceğini gösterir. Aslında $\dot{q}_0$etkisi, yapının uç noktasının konum ve yönelimini değiştirmeyecek şekilde içsel hareketler üretmesini sağlar. Böylelikle, verilen görevin yerine getirilmesi için robot kolu daha becerikli duruşlara göre yeniden düzenlenir.

3 TERS KİNEMATİK PROBLEMLERİNİN SAYISAL ÇÖZÜMLERİ

3.1 Ters Kinematik Problemi

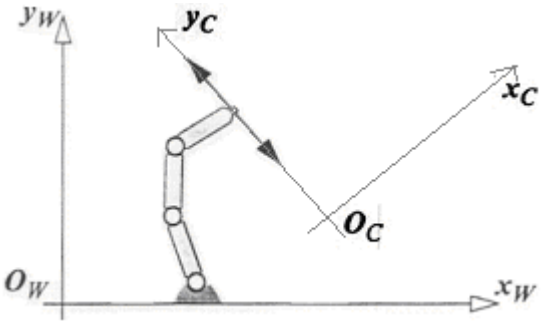
Genelleştirilmiş koordinatları *q* durum vektörünün *n* elemanı olan bir yapının ters kinematik problemi aşağıdaki non lineer denkleme bir çözüm bulmaktır.

$$x(q) = g \tag{3.1}$$

burada *x(q)* ve *g m*-boyutlu vektörler sözde görev uzayında ifade edilmiştir. *m* terimi görevin değerliği olarak adlandırılır. Bu denklem, birçok bağımsız skaler veya vektörel denklemin kümelenmesinden türetilebilir.

Pratikte, kullanıcının gerçekte neyi kontrol etmek istediğini belirlemesine göre, anlamlı görev niteliklerini ifade etmek için daha yüksek seviyede kavramlar gerekmektedir. Tipik olarak, *g* ulaşılan amaç iken *x(q)* görev fonksiyonu bir uç noktasının konum ve yönelimini gösterir.

Görev verilen referans eksen takımına göre ifade edilir. Buradaki referans eksen takımı, *C* kontrol eksen takımı olarak adlandırılır. Kartezyen görevler için kontrol eksen takımı *W* dünya referans eksen takımı ile çakışabilir. Buna rağmen, uç noktası konumu görev tarafından sadece kısmen sınırlanmışsa, genel kontrol eksen takımı kullanımı genel doğru üzerindeki nokta veya düzlem üzerindeki nokta sınırlamalarının belirlenmesine Şekil 3.1 de görülebileceği gibi izin verir.



Şekil 3.1 *C* Kontrol Eksen Takımının *y*-ekseni Boyunca Serbestçe Hareket Eden Uç Noktasının Kısmi Konum Kontrolü

(3.1) denkleminin tam çözümü her zaman bulunmaz. Tam çözümün olmadığı durumda, denklemi sağlayacak yaklaşık sonuç "en iyi" olarak adlandırılır. Bu halde problemimiz aşağıdaki kalıntı hatasını minimize eden bir optimizasyon problemine dönüşür.

$$e(q) = \|x(q) - g\| \tag{3.2}$$

Bu eşitlikte, kullanılan norm Öklid normudur.

3.2 Sayısal Çözünürlük Yöntemlerine Genel Bir Bakış

3.2.1 Analitik Yöntemler

Az sayıda serbestlik derecesine sahip basit robotik kolları için, lineer olmayan eşitliklerin çözünürlüğü ile analitik (veya kapalı formda) çözümler bulunabilir [24,74]. Robot kolları, sıklıkla ters kinematik problemlerine analitik çözümlerin bulunabildiği biçimde tasarlanırlar. Bu çözümler çok hızlı hesaplandıklarından, gerçek zaman uygulamaları için önemlidirler ve çözünürlük yöntemleri dayanıklıdır.

Buna rağmen, genel mafsallı yapılar için analitik çözüm yoktur. Bu durumda ters kinematik problemi çözmek için iteratif, sayısal yöntemler kullanılabilir.

3.2.2 Jakobian Devriği Yöntemi

Welman [93] tarafından sözü edilen jakobian devriği yönteminin, *"resolved motion rate"* yönteminden tek farkı jakobian matrisinin tersi yerine onun devriğini kullanmasıdır. Bu yöntem Wolovich ve Elliot [101] tarafından ortaya atılmış, Sciavicco ve Sciliano [79] tarafından artık robot kolları için, Das [23] tarafından ise ikincil kriterlerin sağlanması için genişletilmiştir. Bu yöntem ile matris tersi almak gerekmediği için iterasyon çok hızlı şekilde gerçekleştirilebilir. Buna rağmen, yakınsama özelliklerinin zayıflığından dolayı, özellikle çözüm yakınında hedefe ulaşmak için gerekli basamak sayısı sözde ters tabanlı yöntemlere göre çok yüksek olabilir. Bu nedenle, jakobian devriği yöntemi sözde ters tabanlı yöntemlere göre daha verimli değildir.

3.2.3 *"Resolved Motion Rate"* Yöntemi

Whitney, öncü çalışmasıyla bilim dünyasını, fazla karmaşık çözünürlük yöntemlerinin temeli olan *"resolved motion rate"* kontrolle tanıştırmıştır [100]. Whitney verilen başlangıç konfigürasyonu için, lineer olmayan denklemleri lineerleştirmeyi, eklem koordinatlarının diferansiyel değişimleri ile görev koordinatlarının diferansiyel değişimlerini ilişkilendiren jakobian matrisi ile gerçekleştirmiştir. Lineer sistem, hedefe daha yakın bir konfigürasyon elde etmek amacıyla çözülür. Sistem genellikle, başlangıç konfigürasyonu yeterli yakınlıkta ise bu işlemin tekrarlanan çözünürlüğüyle, sınırlamayı sağlayan bir çözüme yakınsar. Bu yöntem lineer olmayan denklemlerin çözünürlüğü için iyi bilinen bir iteratif yöntem olan Newton Raphson yönteminden esinlenmiştir.

Tekrar (3.1) eşitliğine dönersek, bu eşitlik genelde lineer olmayan olduğu için, denklemin çözümü *x(q)* fonksiyonunun tersi basitçe alınarak bulunamaz. Bu sorunu çözmek için, Newton Raphson yöntemine [73] oldukça benzer bir *"resolved motion rate"* yöntemi kullanılabilir. Pek çok sayısal yöntemde, sınırlamalı bir denklemin q^0 başlangıç noktası yakınında lineerleştirilmesi, sonucunda jakobian matrisinin elde edildiği iteratif bir işlemdir. Jakobian matrisinin tersinin bulunması ile elde edilen lineer denklemler kümesi Δq artımları için çözülebilir. Bu işlem, Jakobian matrisinin tekil noktaları ile bağlantılıdır. Bir sonraki aşamada Δq artışları, görev denkleminin bir çözümü olan q^i konfigürasyonuna eklenerek yeni konfigürasyon bulunur. Bu işlem iteratif olarak tekrarlanarak, sistem, ya görev denklemini sağlayan bir çözüme, ya da tam çözüm olmadığı durumlarda kalıntı hatasını minimize eden bir çözüme yakınsar.

3.3 Görev Denkleminin Lineerleştirilmesi

Görev fonksiyonuna birinci dereceden lineer yaklaşıklık, o anki q^i konfigürasyonuna göre Taylor serisi açılımı ile aşağıdaki gibi verilir.

$$x\left(q^i + \Delta q\right) = x(q^i) + J(q^i)\Delta q + \ldots \tag{3.3}$$

Burada *J(q)* terimi *x(q)* fonksiyonunun *mxn* boyutundaki jakobian matrisidir. (3.3) eşitliğinden, lineer terimler dışındakileri atarsak, eşitlik,

$$\Delta x \cong J(q^i)\Delta q \tag{3.4}$$

şekline dönüşür. Bu eşitlikteki bilinen $\Delta x = g - x(q^i)$ istenilen görev artışı, bilinmeyen ise Δq eklem koordinatlarındaki artıştır.

İlk olarak, (3.4) eşitliğinden Δq artışı hesaplanır. Bir sonraki konfigürasyon ise aşağıdaki gibi hesaplanabilir.

$$q^{i+1} = q^i + \Delta q$$

Görev fonksiyonunun lineer olmaması nedeniyle, (3.4) eşitliğindeki yaklaşım sadece küçük görev artışları için geçerlidir. Bu işlem için Δq alındığında oluşan hata, Δx istenilen artış ile $x(q^{i+1}) - x(q^i)$ şeklinde ifade edilen gerçek artış ile arasındaki farktır. Eğer bu hata çok büyükse, gereken pek çok iterasyon süresince uç noktasının izlediği yol düz değil kararsız olabilir. Yeterli küçüklükte görev artışı seçmek için, Δx sabit bir eşiğin altına sıkıştırılır. *"Doğru araştırması"* bahsedilen yöntemden daha karmaşık bir yöntem olarak olarak bilinir [77]. Bu yöntemde, Δq yön vektörü hatayı olabildiği kadar minimize etmeyi sağlayacak şekilde seçilen bir çarpan ile ölçeklendirilir. Bu parametrenin seçimi, genellikle yön vektörü boyunca iteratif bir araştırma gerektirir. Sims [86] ve Watt&Watt tarafından tanımlanan bir diğer yönteme göre ise, izleme hatasının belirlenen eşiğin altında kalacak en büyük artışını bulmak için, pek çok denemeye dayanan adaptif artışlar kullanılmıştır. Adaptif artış kullanımı, iterasyon sayısını ve buna bağlı olarak matris tersi alma işleminin zorluğunu azaltır. Fakat diğer taraftan, hatanın değeri zor değerlendirilebilir. Bu sebepten dolayı, bu çalışmadaki sistemlerimizde sabit uzunluklu artışlar kullanılmıştır**.**

Bu aşamada, jakobian tersi alma işlemi için 2. bölümde anlatılan tekil noktaların varlığı bir sorun yaratabilir. Bu sorundan kurtulmak için, kullanışlı bir araç olan Tekil Değer Ayrıştırması (SVD)'nı inceleyelim.

3.4 Tekil Değer Ayrıştırması (SVD)

Artık robot kolları analizinde özel bir ilgi alanı olan tekil değer ayrıştırması (SVD) güçlü bir lineer cebir aracıdır. [77,8,33,44,54]. Çünkü SVD, bir matrisin temel alt

uzayları için açıkça ortonormal bir baz oluşturur. SVD' nin bu kullanışlılığı, bir matrisin kötü konumlanmış olduğu noktaları bulma kabiliyetinde yatar [77,33].

3.4.1 Temel Uzaylar

Herhangi bir *mxn* boyutlu *J* matrisi için [33] tarafından aşağıdaki şekilde tanımlanan $\Re(J)$ değer uzayı, *N(J)* boş uzay olmak üzere, iki lineer alt uzay olduğunu düşünebiliriz.

$$\Re(J) := \{v \in \Re^m \,|\, \exists \omega \in \Re^n, J\omega = v\}$$

$$N(J) := \{v \in \Re^n \,|\, Jv = 0\}$$

Bu kavramlar dikdörtgen veya tekil matrisler için önemlidir. Değer uzayı, *J*' nin uygulanmasıyla ulaşılabilen alt uzaydır ve boyutu *J*' nin rankı olarak adlandırılır. Boş uzay ise boş vektörü gösteren alt uzaydır ve boyutu *J*' nin boşluğu olarak adlandırılır. Bu alt uzayların ortogonal elemanları sırasıyla $\Re(J)^{\perp}$ ve $N(J)^{\perp}$ dir bu elemanlar için aşağıdaki temel ilişkiden söz edebiliriz [33].

$$\Re(J)^{\perp} = N(J^T)$$

$$N(J)^{\perp} = \Re(J^T) \tag{3.5}$$

3.4.2 SVD' nin Tanımlanması

Rankı *r* olan *mxn* boyutlu bir *J* matrisinin SVD'si:

$$J = U\Sigma V^T \tag{3.6}$$

şeklindedir. Bu eşitlikteki $U=\begin{bmatrix} u_1 & \dots & u_m \end{bmatrix}$ terimi *mxm* boyutlu, $V=\begin{bmatrix} v_1 & \dots & v_n \end{bmatrix}$ terimi *nxn* boyutlu birer ortogonal matristir, $\sum$ terimi ise aşağıdaki şekle sahip *mxn* boyutlu bir matristir.

$D = \begin{bmatrix} \sigma_1 & 0 \\ \dots & \dots \\ 0 & \sigma_r \end{bmatrix}$ *rxr* boyutlu köşegenel bir matris olmak üzere $\sum = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$

Bu ifadedeki σ_i terimi, her zaman pozitif olan *i*. nci tekil değeri göstermektedir. *U* matrisinin ve *V* matrisinin sütunları, *J* ile ilişkili dört temel alt uzayı doldurur. Bunlar:

$$B_{R(J)} = \begin{bmatrix} u_1 & \dots & u_r \end{bmatrix}$$

$$B_{R(J)^{\perp}} = \begin{bmatrix} u_{r+1} & \dots & u_m \end{bmatrix}$$

$$B_{N(J)^{\perp}} = \begin{bmatrix} v_1 & \dots & v_r \end{bmatrix}$$

$$B_{N(J)} = \begin{bmatrix} v_{r+1} & \dots & v_n \end{bmatrix} \tag{3.7}$$

olarak ifade edilir. Herhangi *S* alt uzayı üzerine ortogonal izdüşüm matrisi $P_s = B_s B_s^T$ şeklinde basitçe hesaplanabilir. Ayrıca, *J* matrisi

$$J = B_{R(J)} D B_{N(J)^{\perp}}^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T \tag{3.8}$$

olarak yazılabilir.

3.5 Lineer Sistemin Çözümü

Öncelikle *J* jakobian matrisi ve Δx görev artışı hesaplanır. Daha sonra (3.4) eşitliği Δq artışını elde etme yoluyla çözülür. Bu işlem jakobian karesel, tekil değil ise basittir. Nadiren tek çözüm $\Delta q = J^{-1}\Delta x$ şeklinde basitleştirilebilir. *m* sınırlama sayısı genellikle *n* serbestlik derecesinden küçüktür. $m<n$ olan bu eksik sınırlandırılmış durumda, klasik ters alma işlemi uygulanamaz ve çözüm tek kalamaz. Tersi durumda, yani $m>n$ iken, sistem aşırı sınırlandırılmış olabilir. Bu durumda, tam çözüm yoktur ve $J\Delta q - \Delta x$ kalıntı hatası sıfır olamaz. Her iki durumda da, optimal bir çözüm seçilerek ek kriterler tanımlanmalıdır. Eksik sınırlandırılmış durumda, geçerli olan çözümler arasından en iyisi seçilmelidir. Aşırı sınırlandırılmış durumda ise, en iyi yaklaşık çözüm seçilmelidir.

3.5.1 Çözüm Uzayının Ayrıştırılması

$\Re^n$ çözüm uzayı iki ortogonal alt uzaya bölünebilir. Bunlardan *N(J),* Δx'e ulaşma probleminin sağlanmasına katkıda bulunmayan çözüm kümelerinin verir. $N(J)^{\perp}$ ortogonal tümleyeni ise belirtilen katkıyı sağlar. Bu nedenle genel çözüm iki terimden oluşur. Bunlardan ilki, (3.4) eşitliğini mümkün olduğu kadar sağlayan özel çözüm terimi, diğeri ise kullanılabilecek kriterleri sağlayan homojen çözüm terimidir.

3.5.2 En Küçük Kareler Yöntemi

Özel çözüm genellikle *J*'nin $J^{\dagger}$ en küçük kareler tersine dayanır.

$$\Delta q_{en-küçük-kareler} = \mathrm{J}^{\dagger}\Delta x \tag{3.9}$$

Bazen Moore-Penrose tersi veya sözde ters olarak adlandırılan en küçük kareler tersi, hem eksik sınırlandırılmış hem de aşırı sınırlandırılmış durumda en küçük kareler kriterini sağlayan genelleştirilmiş terstir [7,68]. Daha açarsak, $J^{\dagger}\Delta x$ en küçük kareler çözümü bunların arasında $\left\|J^{\dagger}\Delta x\right\|$ şeklinde ifade edilen kalıntı hatasını minimize eden minimum normlu vektördür. Başka bir deyişle, en küçük kareler çözümü eğer mümkünse eklem artışlarını küçük tutarak istenilen Δx artışlarını karşılamaya çalışır. Burada, $\left\|J^{\dagger}\Delta x\right\|$ minimizasyonunun Δq minimizasyona göre, önceliğe sahip olduğu dikkate alınmalıdır.

mxn boyutlu *J* matrisinin $J^{\dagger}$ en küçük kareler tersi aşağıdaki dört özelliği sağlayan tek matris olarak da tanımlanabilir.

$$JJ^{\dagger}J = J$$

$$J^{\dagger}JJ^{\dagger} = J^{\dagger}$$

$$(JJ^{\dagger})^{T} = JJ^{\dagger}$$

$$(J^{\dagger}J)^{T} = J^{\dagger}J \tag{3.10}$$

Bu tersi hesaplamak için bazı kaynaklarda başka özellikler ve yöntemler de bulunabilir [7,49]. Pek çok uygulama alanı olduğu için en küçük kareler problemi ile ilgili önemli sayıda yöntem geliştirilmiştir [8,68,46].

3.5.3 Homojen Çözüm

(3.9) eşitliğinin özel çözümü $N(J)^{\perp}$ alt uzayına sıkıştırılmıştır. Çünkü görev artışını gerçekleştirmede kullanışlı elemanlar bu alt uzaydadır. Artık problem durumunda, $N(J)^{\perp}$ 'nin ortogonal tümleyeni N(J), görevin sağlanmasında etkisi olmayan ek elemanlar içerir. Bu nedenle, boş uzay bazen artık uzayı olarak da adlandırılabilir [35]. Bu sebeple, *N(J)* 'ye ait bir elemanın çözüme eklenmesi, görevi gerçekleştirebilecek pek çok çözümden birinin seçilmesine izin verir. Bu eklenen eleman, homojen çözüm olarak adlandırılır ve çeşitli amaçlar için kullanılabilir.

$$\Delta q_{\text{homojen}} = P_{N(J)} z$$

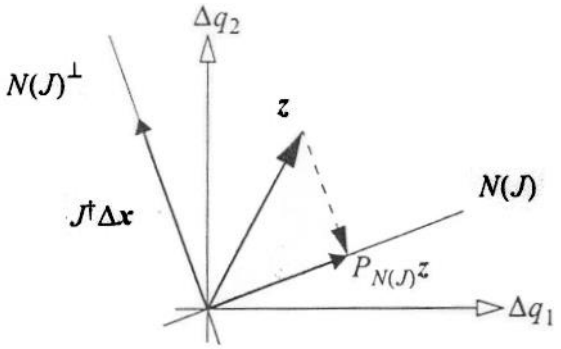
Buradaki $P_{N(J)} = I_n - J^{\dagger} J$ terimi *nxn* boyutlu *N(J)* üzerine ortogonal bir izdüşüm operatörüdür. $z \in \Re^n$ ise keyfi bir vektördür.

3.5.4 Genel Çözüm

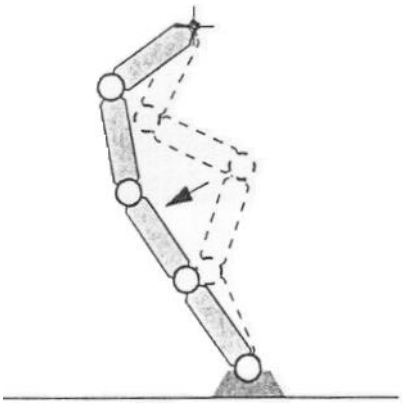
Özet olarak, (3.4) eşitliğinin genel çözümü Şekil 3.2'de görüldüğü gibi özel ve homojen çözümün birleşmesinden oluşur.

$$\Delta q = J^{\dagger} \Delta x + P_{N(J)} z \tag{3.11}$$

z vektörü, kazanç çarpanı ε<0 olmak üzere $z = \varepsilon \nabla h(q)$ şeklinde yerine yazılmasıyla, *h(q)* skaler kriterini lokal olarak minimize etmek için kullanılabilir. Bu robotik alanında iyi bilinen bir sonuçtur [48]. Örnek bir uygulaması, Şekil 3.3'de görüldüğü gibi eklem açılarını mümkün olabildiği kadar istenilen değerlere yakın tutabilmesidir.



Şekil 3.2 İki Boyutlu Eklem Değişim Uzayında (*n=2*) Gösterilen Bir Boyutlu Görev (*m=1*) için (3.11) eşitliğinin Genel Çözümünün İki Elemanı



Şekil 3.3 Tepe Konumu Sınırlamasına Aykırı Hareket Etmeden *h(q)* Skaler Kriterinin Minimizasyonu

$h(q) = \|q - q_{istenilen}\|^2$ şeklindeki skaler kriteri minimize ederek açılar, istenilen $q_{istenilen}$ duruşuna olabildiği kadar yakın tutulabilir.

3.5.5 En Küçük Kareler Çözümünün Sorunu: Düzenlenme Gereksinimi

Görülen çekiciliğine rağmen en küçük kareler çözümünün büyük bir sorunu vardır. Tekil nokta yakınlarında, problem kötü konumlanmış hale gelir. Bu bölgelerde, en küçük kareler çözümü kalıntı hatasını minimize etmeye çalışırken, sonsuza doğru yönelebilir [54]. Bu sorun, pratikte (3.4) eşitliğindeki küçük artışlar hipotezine uymadığı için kabul edilemez. Bu problemin çözümü ile ilgili aşağıdaki düzenleme teknikleri geliştirilmiştir;

- Kısaltılmış singüler değer ayrıştırması (SVD)
- Sönümlendirilmiş en küçük kareler (Tikhonov düzenlemesi [TIK 63])

- Sayısal filtreleme [52] (sönümlendirilmiş en küçük kareler tekniğinin genişletilmiş hali)

3.5.6 Sönümlendirilmiş En Küçük Kareler

Bu yöntemde, sadece izleme hatasını minimize etmek yerine, aşağıdaki ağırlıklandırılmış izleme hatası ve çözüm normu kombinasyonu minimize edilir.

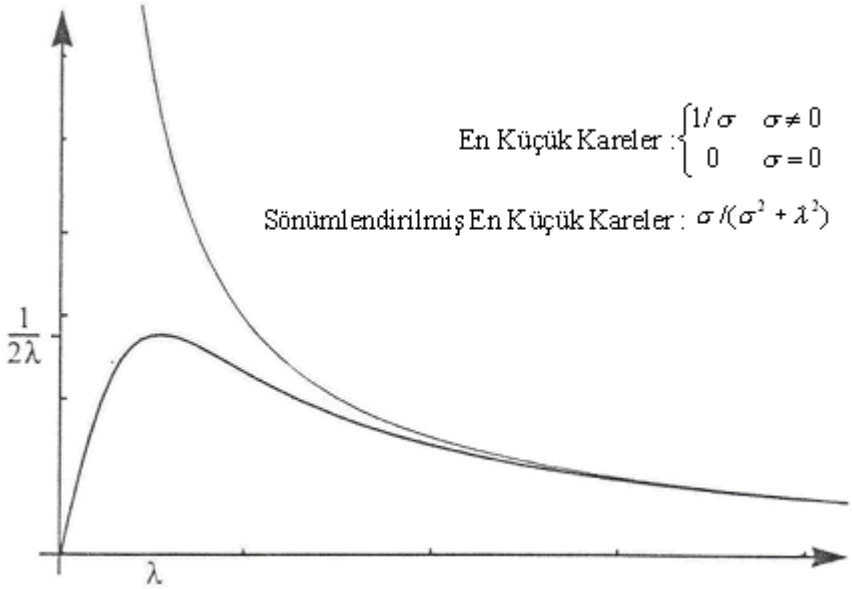
$$\left\| J\Delta q - \Delta x \right\|^2 + \lambda^2 \left\| \Delta q \right\|^2 \tag{3.12}$$

Bu eşitlikteki λ terimi sönümlendirme çarpanı olarak bilinir ve izleme hatasının çözüm normuna göre önemini ağırlıklandırır. $\lambda=0$ olduğunda, sadece izleme hatası minimize edilir ve bu çözüm en küçük kareler çözümüne denktir. λ artarsa, izleme hatasının doğruluğunun zorlaşmasıyla çözüm normu azalmaya zorlanır. Bundan dolayı, ödünleşim (trade-off) bulunmalıdır.

(3.12) eşitliğini minimize edecek çözümün hesaplanmasında kullanılan yöntem en küçük kareler tersinin genel hali olan sönümlendirilmiş en küçük kareler yöntemidir. Robotik alanında kullanılan bu teknik, Wampler [90] ve Nakamura [64] tarafından önerilmiştir. $\lambda>0$ durumunda J matrisinin sönümlendirilmiş en küçük kareler tersi:

$$J^{\dagger\lambda} = J^T (JJ^T + \lambda^2 I_m)^{-1} \tag{3.13}$$

şeklinde ifade edilir.



Şekil 3.4 En Küçük Kareler Çözümü İle Sönümlendirilmiş En Küçük Kareler Çözümünün σ Skaler Değerinin Fonsiyonu Olarak Karşılaştırılması

Yukarıdaki Şekil 3.4'de iki çözüm yöntemi tek boyut boyunca karşılaştırılmıştır. Açık olarak görülebileceği gibi, en küçük kareler çözümü sınırlı olmamasına rağmen sönümlendirilmiş çözüm sınırlıdır. σ=0 tekil noktasında en küçük kareler fonksiyonu süreksiz iken, sönümlendirilmiş en küçük kareler fonksiyonu süreklidir ve λ>0 olmak üzere 1/(2λ) ile sınırlıdır.

Sönümlendirilmiş en küçük kareler tersi kullanıldığında, en küçük kareler tersinin bazı özelliklerinin geçerliliğini koruyamaması dikkate alınmalıdır. Örneğin, λ>0 olduğunda, aşağıda da görülebileceği gibi (3.10) eşitliklerinin ilk ikisi geçerliliğini yitirir.

$$JJ^{\dagger\lambda}J \neq J$$

$$J^{\dagger\lambda}JJ^{\dagger\lambda} \neq J^{\dagger\lambda}$$

Sönümlendirilmiş ters alma işlemi (3.11) eşitliğinde görülen izdüşüm matrisinde çekici olmasına rağmen, hatalı olabilir. Çünkü λ>0 olduğunda, görevin yerine getirilmesini etkileyerek istenilmeyen $J(I_n - J^{\dagger\lambda}J)z \neq 0$ görev hareketine neden olabilir.

3.5.7 SVD ile Sönümlendirilmiş En Küçük Kareler Tersinin Hesaplanması

r ranklı *J* matrisinin SVD' sinden pek çok önemlik nicelik, özellikle de sönümlendirilmiş en küçük kareler tersi hesaplanabilir.

$$\left(J = \sum_{i=1}^{r} \sigma_i u_i v_i^T \right) \rightarrow \begin{cases} J^{\dagger} = \sum_{i=1}^{1} \frac{1}{\sigma_i} v_i u_i^T \\ J^{\dagger\lambda} = \sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i u_i^T \\ J^{\dagger} J = P_{N(J)^{\perp}} = \sum_{i=1}^{r} v_i v_i^T \end{cases} \tag{3.14}$$

SVD' nin hesaplanışının zorluğu herkes tarafından bilinir. Bu yüzden, robotik alanında gerçek zamandaki hesaplamalardan daha çok problem analizinde tercih edilir. Maciejewski, bir iterasyondan diğerine artan değerlendirmeli SVD kullanıldığında bu hesaplama zorluğunun basitleşebildiğini göstermiştir [53]. Buna rağmen, bu yöntem yerine getirilmek için oldukça karmaşıktır. SVD kullanımı, şu an kullanılan bilgisayarlarda gerçek zaman etkileşimini önemli biçimde etkilemez, fakat bir algoritmanın verimliliğinin geliştirmesinde her zaman tercih edilir. SVD' nin tamamı yerine azaltılmış (thin) SVD kullanımı basit bir hızlanma sağlar.

Azaltılmış SVD, SVD' nin budanmış hali olarak çok kullanılır [33] (s. 72). Tek farkı, $R(J)^{\perp}$ için bir ortogonal baz sağlayamamasıdır. Yine de, SVD' nin tamamının uygulanmasından daha hızlı olma avantajıyla, (3.14) eşitliğinin hesabı için yeterlidir. SVD' yi gerçekleştirmek için gereken kaynak kodu "Numerical Recipes" adlı kitapta bulunabilir [77].

Hesaplama açısından bakıldığında, $m<n$ olmak üzere mxn boyutlu bir J matrisinin azaltılmış SVD' si J^T devriğine göre daha yavaş hesaplanabilmesi ilginç bir nottur. Azaltılmış SVD kodunda [77] gerçekleşen hız ölçümü $(n/m)^{3/4}$ hızlandırma çarpanını gösterir. Bu durum, $J^{\dagger\lambda}$ ve $J^{\dagger}J$ hesaplamalarının zorluğunun azaltılmasında kullanılabilir. Çünkü J^T' nin SVD' si (3.5) eşitliği sayesinde bu hesaplamalar için önemli bir temel sağlar.

3.5.8 Uygun Sönümleme Çarpanının Hesabı

En küçük kareler yöntemindeki problem, bütün durumlar için optimum λ sönümlendirme çarpanının bulunmasıdır: λ çarpanı mümkün olan en iyi takibi sağlamak için tekil noktalardan uzakta sıfır, tekil noktaların yakınında ise istenmeyen osilasyonları azaltmak için yeterince büyük olmalıdır. Buna rağmen, çok yüksek λ değerleri takip doğruluğunun zayıflamasına yol açar.

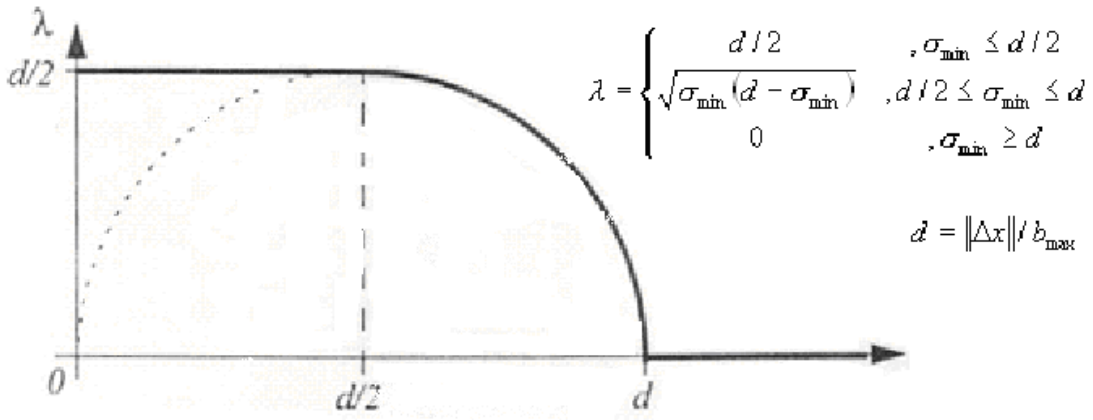
Nakamura [64], Wampler [90] ve Maciejewksi [52] sönümlenme çarpanının hesaplanmasında değişik yollar üzerinde tartışmışlardır. Hepsinin ortak yöntemi çözümün normuna bir b_{max} sınırı koymaktır:

$$\left\| J^{\dagger\lambda} \Delta x \right\| \le b_{\max} \tag{3.15}$$

bu eşitsizlik (3.14) yardımıyla aşağıdaki şekilde yazılabilir:

$$\left\| J^{\dagger\lambda} \Delta x \right\| = \left\| \left(\sum_{i=1}^{r} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i u_i^T \right) \Delta x \right\| = \sqrt{\sum_{i=1}^{r} \left(\frac{\sigma_i}{\sigma_i^2 + \lambda^2} \left(u_i^T \Delta x \right) \right)^2} \le b_{\max} \tag{3.16}$$

Çözüm normu, beklenildiği gibi λ arttıkça monoton olarak azalmaktadır. Bu yüzden, jakobianın SVD' si elde edilebildiğinde, verilen bir b_{max} sınırı için en uygun λ çarpanı Newton' un yöntemiyle iteratif olarak bulunabilir. Jakobianın SVD' si elde edilemiyorsa (hesaplanmasının çok zor olduğu düşünüldüğünde), σ_{min} minimum tekil değeri (ya da tahmini) en uygun olmasa da λ sönümleme çarpanının hesabı için yeterlidir. Bu çalışmada, Maciejewski [52] tarafında önerilmiş ve aşağıdaki şekilde özetlenen, sönümleme çarpanı kullanılmıştır.



Şekil 3.5 λ Sönümleme Çarpanının Minimum Tekil Değerin Fonksiyonu Olarak Seçimi

3.6 İteratif İşlemin Çözüme Yakınsaması

İteratif yöntemle birlikte ortaya çıkan önemli soru, çözümün istenilen çözüme yakınsayıp yakınsamayacağıdır.

Newton-Raphson yöntemi, uygun koşullar altında, çözüme yeterince yakın verilen bir başlangıç noktasıyla, non-lineer eşitlikler kümesine yakınsar. [73] Çözüme yakınsama için teorik koşullar oldukça sınırlayıcıdır, fakat pratikte yöntem daha iyi çalışır. Orjinal Newton yöntemindeki sınırlama çözümü olmayan sistemlerin üzerinden gelememesidir. Bu sınırlamanın üstesinden gelmek için Ben Israel, (yukarıdaki iteratif yöntemde tanımlanan) olası tekil jakobian matrislerinin tersini almak için en küçük kareler tersi ile genişletilmiş bir yöntem önermiş ve yakınsaması üzerinde çalışmıştır. [5,6] İteratif prosesin, $e(q)$ (3.2) eşitliğindeki kalıntı hatasını göstermek üzere, $e(q)^2$' nin durağan bir q^* noktasına yakınsayacağı kanıtlanmıştır.

Bu sebeple q^*,(3.1) eşitliğine ters kinematik problemin çözümü olmadığı durumlarda, $e(q)$'yu minimize eden bir çözüm bulan, bir en küçük kareler çözümüdür. Bu durumda ise diferansiyel seviyede en küçük kareler kriteri kullanılması son çözüm tercihine yansır.

3.7 Sonlandırma

Sonlandırma koşulu, iteratif işlemin ne zaman durdurulabileceğini tanımlamalı ve o anki konfigürasyonun çözüm olarak alınmasını sağlamalıdır. Bu noktada, istenilen doğruluğa ulaşılıp ulaşılmadığını ($e(q) < \varepsilon$) denetlemek için, bir ε yakınsama toleransı tanımlanmalıdır. Bu koşul, çözüm olduğunda ve çözüme yakınsandığında, işlemi durdurur. Fakat çözüm olmadığında hedefe uzaklık çok küçük olmadığından sonlandırma algoritmasını farkedemez. Algoritmayı sonlandırmak için diğer olasılık, $e(q)$ kalıntı hatasının azalmasının durmasıdır. Bu olasılık durumu, sadece alınan adım büyüklüğü için *"doğru araştırması algoritması"* kullanıldığında güvenilirdir. [77] Çünkü bu algoritma, başarılı iki iterasyon arasında hatanın düşmesini garanti eder.

İteratif işlem boyunca herhangi bir anda görevin değişebildiği bir robot kolu sistemi içerisinde, sonlandırma koşuluna olan ihtiyaç daha azdır. Çünkü algoritma sürekli olarak çalışmayı sürdürecektir.

3.8 Sonuç

Bu bölümde, ters kinematik probleminin nümerik çözünürlüğü için klasik yöntemler ana hatları ile incelenmiştir.

4 TERS KİNEMATİK ÇÖZÜMDEKİ SINIRLAMALAR

4.1 Artık Robot Kolları

İnsan kolunun yedi serbestlik derecesine sahip olduğu düşünülebilir. Bir kol, altı serbestlik derecesine (üçü uzayda bir nesneyi belirli bir konumda tutmak, diğer üçü de bu nesnenin yönelimini belirlemek, olmak üzere) ihtiyaç duyar. Bu yüzden, insan kolu tutulan nesnenin konumlandırılması ve yönlendirilmesi temel görevi için bir artık serbestlik derecesine sahiptir. Bu durum, bir nesneyi sabit bir konum ve yönelimde tutarken dirseğimizi nasıl hareket ettirebildiğimizi açıklar. Bahsedilen fazla serbestlik derecesi artıklıktır ve bu insan koluna büyük bir çok yönlülük ve geniş bir uygulanabilirlik kazandırır.

Aynı iddia, robot kollarına uygulanır. Altı serbestlik derecesi uç noktasının tüm pozisyon ve yönelimi için yeterli olduğu için yedi serbestlik dereceli bir robot koluna artık denilebilir. Hatta altı veya daha az serbestlik dereceli robot kolları da, yerine getirdikleri özel görev robot kolunun sahip olduğundan daha az serbestlik derecesi gerektirdiğinde, sıklıkla artıklığa sahip sayılırlar. Bu durum, artıklık konusu üzerinde düşünme nedenlerimizden bir tanesidir.

Artık robot kollarının avantajları nelerdir? Genel söylemde, insan kolundaki gibi, çok yönlülük ve uygulanabilirlikte üstünlük kazandırır. Daha açık olarak, artık robot kolları tekil noktalardan, engellerden, yapısal limitlerden (örneğin, dönel eklemlerin açı limitleri) sakınma, uygun hareketleri yerine getirme (örneğin, düşük enerji tüketen hareket, ayrı eklem hızları arasında dengeli hareket, eklemleri sürme kuvvetleri arasında dengeli konfigürasyon), nesnelerin arkasına ulaşma, içbükeylerin arasında sürünme gibi hareketleri yerine getirmede avantaj sağlar. Artıklık, bir robot kolunu bazı eklemlerin arızalanması durumunda bile belirli görevleri yerine getirmede, daha güvenilir yapar. Bu özellik, uzay ve derin denizler gibi bazı uygulama alanlarında çok önemlidir.

Tabii ki, artık robot kollarının dezavantajları da vardır. Daha fazla eklem ve eyleyiciye sahiptirler. Yapıları daha karmaşık, daha hantal ve daha ağırdır. Daha

karmaşık kontrol algoritmaları gerektirler, bu yüzden gerekli hesaplama miktarı artar. Artık robot kolları, bu dezavantajları avantajları tarafından dengelenmedikçe tamamen yararlı olmayacaktır.

4.2 Görev Ayrıştırması Yaklaşımı

4.2.1 Görevi Öncelik Sırası ile Alt Görevlere Ayrıştırma

Bir robot koluna verilen en karmaşık görevler, görevin öncelik sırası ile birkaç alt göreve bölünmesi yoluyla kesin ve açık bir şekilde belirtilebilir. Bütün daha yüksek önceliğe sahip alt görevler yerine getirildikten sonra her alt görev, kalan serbestlik derecesi kullanılarak yerine getirilir. Örneğin kaynak yapma görevi, ilk öncelik el konumlamasında olmak üzere el konum kontrolü ve el yönelim kontrolü olarak alt görevlere bölünebilir. Çalışma uzayındaki engelli bir yörünge takibi görevi, ilk öncelik elin konum ve yönelim kontrolünde olmak üzere, elin konum ve yönelim kontrolü ve engellerden sakınma olarak ayrıştırılabilir. Artık robot kollarının kontrol problemlerine, alt görev yöntemi üzerinden yaklaşılabileceği açıktır. Alt görev yönteminde, bir artık robot kolu tarafından gerçekleştirilen görev ilk önceliğe sahip alt görev, artıklık kullanım (tekil noktalardan sakınma, engellerden sakınma gibi) nedeni ise ikinci önceliğe sahip alt görev sayılmaktadır [35,65].

Burada her alt görevin, ya alt görevi tanımlamak için uygun değişkenlerin istenilen yörüngesi, ya da belirli bir ölçüt, biçiminde verileceği kabul edilecektir. İlk biçim, verilen alt görevi başaran bir yörünge verilmesi durumunda kullanılır. İkinci biçim ise, istenilen gerçek yörüngenin bilinmediği fakat yörüngenin nasıl değerlendireceğinin bilindiği durumda kullanılır. Örneğin alt görev, robot kolu konfigürasyonunu tekil noktalardan uzak tutma ya da eklem açılarını sınırları içinde tutma ise, bir ölçüt kullanılarak konfigürasyon değerlendirilir.

Açıklamayı kolaylaştırmak için tartışma, görevin iki alt göreve bölündüğü ve ilk alt görevin istenilen yörünge biçiminde verilmesi basit durumuyla sınırlandırılacaktır.

4.2.2 Temel Denklemler

n serbestlik dereceli bir robot kolu düşünelim. i. nci eklemin eklem değişkeni q_i ($i=1,2,...,n$) dir. Robot kolu konfigürasyonu, $q=[q_1,q_2,...,q_n]^T$ ile gösterilir.

İlk alt görevin m_1 boyutlu q'nın fonksiyonu olan aşağıdaki y_1 vektörü ile tanımlanabileceği varsayalım.

$$y_1 = f(q) \tag{4.1}$$

Ayrıca, t_f bitiş zamanını göstermek üzere $0 \le t \le t_f$ aralığında y_1 için istenilen yörüngenin, y_{1d} olduğunu kabul edelim. Bu durumda, manipülasyon görevini tanımlamak için uygun y_1 tipindeki vektörler, manipülasyon vektörü olarak adlandırılırlar.

İkinci alt görev için aşağıdaki iki durumu düşünelim:

Durum 1 m_2 boyutlu y_2 manipülasyon vektörü,

$$y_2 = f_2(q), \tag{4.2}$$

şeklinde verilir ve y_2 için ikinci alt görev $0 \le t \le t_f$ aralığında istenilen $y_{2d}(t)$ yörüngesi ile belirtilir.

Durum 2 Aşağıdaki eşitlik biçiminde bir ölçüt fonksiyonu verilir

$$p = V(q) \tag{4.3}$$

ve ikinci alt görev bu ölçütü mümkün olduğu kadar büyük tutmaktır.

Üzerinde düşünülmesi gereken ilk problem, $y_1=y_{1d}$ eşitliğini sağlayan ilk alt görevi başarıyla yerine getiren eklem hızlarını elde etmektir. Sonraki problem ise, eğer kalan (kullanılmayan) yetenek varsa, ikinci alt görevi her an mümkün olduğunca fazla yerine getirmektir. Bu düşünceler, robot kolu dinamiği düşünülmeden probleme tamamen kinematik açıdan bakılarak ifade edilmiştir. Her iki durum için de problemi, bölüm 4.2.5'de yapılacağı gibi daha ayrıntılı bir matematik kullanılarak anlık optimal bir kontrol problemi olarak ifade etmek mümkündür. Yine de, problemi ilk olarak oldukça buluşsal bir yolla çözelim.

Bu yolla, ilk alt görevi yerine getiren eklem hızlarının genel çözümü verilecektir. (4.1) eşitliğinin zamana göre diferansiyelini alırsak,

$$\dot{y}_1 = J_1 \dot{q} \tag{4.4}$$

eşitliğini elde ederiz. Bu eşitlikteki $J_j = df_i(q)/dq^T$ (j = *1,2*) terimi y_i'nin q'ya göre jakobian matrisidir. y_{1d} istenilen yörüngesi verildiğinde $\dot{q}$ için (4.4) eşitliğinin genel çözümü,

$$\dot{q} = J_1^{\dagger}\dot{y}_{1d} + (I - J_1^{\dagger}J_1)k_1 \tag{4.5}$$

olarak elde edilir. Burada, k_1 terimi n boyutlu keyfi bir sabit vektördür. Bu eşitliğin sağ tarafındaki ilk terimi, $y_{1d}(t)$ istenilen yörüngeyi gerçekleyen eklem hızıdır.

(4.4)eşitliğini $\dot{q}$ için gerçekleyen birden fazla çözüm olması durumunda, aynı terim $\dot{q}$'nin Öklid normu olan $\|\dot{q}\|$'yı minimize eden çözümü verir. Yine aynı terim, (4.4) eşitliğini gerçekleyen çözüm olmaması durumunda $\|\dot{y}_{1d} - J_1\dot{q}\|$ normunu minimize eden yaklaşık sonucu verir. Bu, yakınsama hatası için bir çeşit büyüklüktür. (4.5) eşitliğinin sağ tarafındaki ikinci terim ise, ilk alt görev gerçekleştirildikten sonra kalan artıklığı gösterir. (4.5) eşitliği artıklık kullanımı için basit bir denklemdir.

4.2.3 İstenilen Yörünge ile Verilen İkinci Alt Görev

Bu alt bölümde, ikinci alt görevin y_2 manüpülasyon değişkeninin istenilen $y_2d(t)$ yörüngesi ile tanımlandığı, durum 1 göz önüne alınacaktır. $y_2d(t)$ yi mümkün olana en yakın şekilde gerçekleştirmek için yapmamız gereken, (4.5) eşitliğindeki k_1 terimini seçmektir. (4.2) eşitliğinin diferansiyelini alırsak,

$$\dot{y}_2 = J_2\dot{q} \tag{4.6}$$

bu eşitlikte $\dot{y}_2$ yerine $\dot{y}_{2d}$ yazıp, (4.5) eşitliğini (4.6) eşitliğinde yerine koyarsak,

$$\dot{y}_{2d} - J_2J_1^{\dagger}\dot{y}_{1d} = J_2(I - J_1^{\dagger}J_1)k_1 \tag{4.7}$$

eşitliğini elde ederiz. $J_2 = J_2(I - J_1^{\dagger}J_1)$ yazıp lineer cebir özelliklerini kullanırsak, (4.7) eşitliğimiz,

$$k_1 = J_2^{\dagger}(\dot{y}_{2d} - J_2J_1^{\dagger}\dot{y}_{1d}) + (I - J_2^{\dagger}J_2)k_2 \tag{4.8}$$

şekline dönüşür. Bu eşitlikteki k_2 terimi n boyutlu keyfi bir sabit vektördür. Dikkat edilecek bir diğer nokta,

$$(I - J_1^{\dagger} J_1) J_2^{\dagger} = J_2^{\dagger} \tag{4.9}$$

ilişkisinin geçerli olduğudur**.** Böylece, (4.5), (4.8) ve (4.9) eşitliklerinden,

$$\dot{q}_d = J_1^{\dagger} \dot{y}_{1d} - J_2^{\dagger} (\dot{y}_{2d} - J_2 J_1^{\dagger} \dot{y}_{1d}) + (I - J_1^{\dagger} J_1 - J_2^{\dagger} J_2) k_2 \tag{4.10a}$$

eşitliğini elde ederiz. Burada, $\dot{q}_d$ terimi ilk olarak y_{1d} yörüngesini daha sonra kalan artıklığı kullanarak mümkün olduğunca yakın bir şekilde y_{2d} yörüngesini gerçekleştiren, istenilen eklem hızlarını gösterir.

Eğer $(I - J_1^{\dagger} J_1 - J_2^{\dagger} J_2)$ terimi sıfır değilse, bunlardan başka bir miktar daha artıklık üçüncü bir alt görevi gerçekleştirmek için kalmıştır. $y_2 = q$ olduğunda (ikinci alt görev, tüm robot kolu konfigürasyonunun istenilen yörüngesini sağladığında), $J_2 = I$ olur. Böylelikle,

$$J_2^{\dagger} = (I - J_1^{\dagger} J_1)^{\dagger} = (I - J_1^{\dagger} J_1) \tag{4.10b}$$

eşitliklerini elde ederiz. Buradaki eşitlikler yardımıyla, (4.10a) eşitliği aşağıdaki k_2 teriminin yok olduğu basit duruma dönüşür.

$$\dot{q}_d = J_1^{\dagger} \dot{y}_{1d} + (I - J_1^{\dagger} J_1) \dot{y}_{2d} \tag{4.10c}$$

Her eklem eksen fonksiyonlarının servo döngüleri istenilen eklem hızlarını kusursuz bir şekilde ürettiğinde, $y_{1d}(t)$ yörüngesi hatasız bir şekilde izlenir. Kalan serbestlik derecesinin yetersiz oluşu nedeniyle, $y_{2d}(t)$ istenilen alt görev yörüngesi ile $y_2(t)$ gerçekte izlenen yörünge arasında sıklıkla sapmalar oluşur. Böyle bir durumla başa çıkmanın bir yolu $\dot{y}_{2d}$ istenilen hızını sapmaları hesaba katarak değiştirmektir. Örneğin, $\dot{y}_{2d}^{*}$ değiştirilmiş hızını H_2 terimi uygun köşegensel kazanç matrisini göstermek üzere,

$$\dot{y}_{2d}^{*} = \dot{y}_{2d} + H_2 (\dot{y}_{2d} - y_2) \tag{4.11}$$

şeklinde tanımlayabiliriz. Sonra da (4.10a) veya (4.10c) eşitliklerindeki $\dot{y}_{2d}$ terimi yerine $\dot{y}_{2d}^{*}$ yazabiliriz.

4.2.4 Ölçüt Fonksiyonu ile Verilen İkinci Alt Görev

Bu alt bölümde, ikinci alt görevin bir ölçüt fonksiyonu ile tanımlandığı, durum 2 göz önüne alınacaktır. Durum 1 de olduğu gibi, (4.5) eşitliğindeki k_1 terimini, (4.3) eşitliğinde verilen p ölçütünü mümkün olan en büyük yapacak şekilde seçeriz. k_1 'i belirlemek için aşağıdaki eşitlikler doğal bir yaklaşımdır: [48]

$$k_1 = \xi k_p \,, \tag{4.12}$$

$$\xi = [\xi_1, \xi_2, ... \xi_n]^T, \tag{4.13}$$

$$\xi_1 = \partial V(q) / \partial q_1 \,, \tag{4.14}$$

Bu eşitliklerdeki k_p terimi uygun bir pozitif sabittir. $\dot{q}_d$ istenilen hızları ise,

$$\dot{q}_d = J_1^{\dagger} \dot{y}_{1d} + (I - J_1^{\dagger} J_1) \xi k_p \tag{4.15}$$

ile verilir. Buradan,

$$\dot{p} = \xi^T J_1^{\dagger} \dot{y}_{1d} + \xi^T (I - J_1^{\dagger} J_1) \xi k_p \tag{4.16}$$

elde edilir. $(I - J_1^{\dagger} J_1)$ teriminin tanımı gereği negatif olmamasından dolayı, (4.16) eşitliğinin sağ tarafı da negatif değildir. Bu durum, p ölçüt değerinin artmasına neden olur.

Matematiksel olarak, (4.12) eşitliğindeki k_1 vektörü $V(q)$ fonksiyonunun en yüksek eğim vektörüdür. $(I - J_1^{\dagger} J_1) \xi k_p$ vektörü, k_1 'in J_1 boş uzayı üzerine dik izdüşümüne karşılık gelir. k_p sabiti ise, $\dot{q}_d$ 'yi aşırı büyük hale getirmemek koşulu altında p 'yi mümkün olan en çabuk şekilde arttırabilmek için seçilir.

4.2.5 Anlık Optimizasyon Problemi Olarak Formülize Etme

4.2.2 alt bölümündeki problem formülasyonu ve 4.2.3 ve 4.2.4 alt bölümlerindeki çözümler oldukça sezgiseldir. Bu alt bölümde, bu sonuçların ikinci alt görev için bazı anlık optimizasyon problemlerine uygun ölçüt fonksiyonunun uyarlanması ile daha özenli bir şekilde elde edilebileceği gösterilecektir.

Durum 1 de, ikinci alt görev için ölçüt fonksiyonu,

$$p_1 = \left\| \dot{y}_{2d} - \dot{y}_2 + H_2(y_{2d} - y_2) \right\| \tag{4.17}$$

olsun. **(**4.17) eşitliğini her zaman aralığında (4.5) eşitliğindeki sınırlama altında minimize eden optimum çözüm**,** **(**4.10a) ve (4.11) eşitlikleri ile verilir.

Durum 2 için, (4.3) eşitliğindeki *V(q)* orijinal ölçütü kullanılarak, ikinci alt görev için ölçüt fonksiyonu,

$$p_2 = k_p \frac{dV(q)}{dt} - \frac{1}{2} \left\| \dot{q} - J_1^{\dagger} \dot{y}_{1d} \right\|^2 \tag{4.18}$$

olsun. (4.18) eşitliğini her zaman aralığında (4.5) eşitliğindeki sınırlama altında maksimize eden optimum çözüm, (4.15) eşitliği ile verilir. (4.18) eşitliğindeki ilk terim, orijinal ölçüt fonksiyonunu arttırmaya çalışır; ikinci terim ise $\dot{q}$ ile ilk alt görevin $J_1^{\dagger}\dot{y}_{1d}$ minimal norm çözümü arasındaki farkı minimize etmeye çalışır. k_p sabiti, ilk gayretin (effort) ikinci gayrete göre öneminin ağırlığını gösterir.

4.3 Tekil Noktalardan ve Mekanik Eklem Limitlerinden Sakınma Uygulamaları

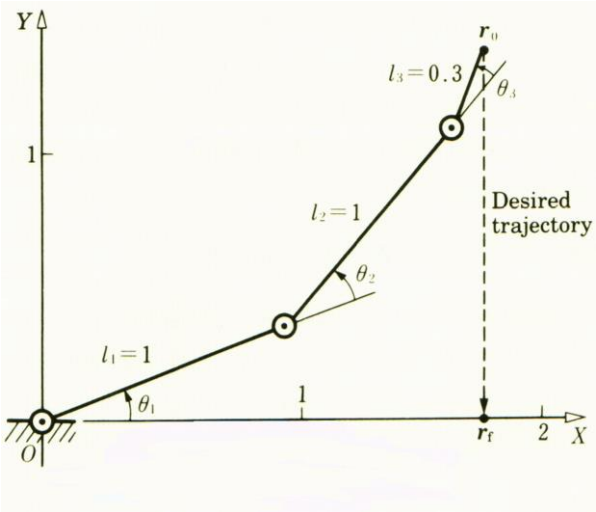
4.3.1 Tekil Noktalardan Sakınma

İkinci alt görevin ölçüt fonksiyonu ile belirlendiği Durum 2'ye örnek olarak, tekil konfigürasyonlardan kaçınmanın mümkün olduğu kadar çok olduğu, uç noktasının istenilen yörüngeyi izlemesi problemini düşünelim [97].

Bir robot kolu genellikle, uç noktasının konum veya yönelimini belirli yönlerde değiştirmek için hareketini daha fazla sürdüremediği konfigürasyonlara sahiptir. Bu konfigürasyonlar, tekil konfigürasyonlar olarak adlandırılırlar. Bir robot kolu için sadece tekil konfigürasyonlara gelmesi değil aynı zamanda yaklaşması da istenilmeyen bir durumdur. İstenilen yörüngeyi korumaya çalışırken bazı eklem hızları, aşırı büyük hale gelebilir. Manipüle edilebilirlik ölçüsü, $V(q)$ manipüle edilebilirliği göstermek üzere aşağıdaki formülasyonla hesaplanabilen, bir robot kolunun uç noktasını herhangi bir yönde serbestçe hareket ettirebilme yeteneğinin nicel bir ölçüsüdür.

$$V(q) = \sqrt{\det(J(q)J^T(q))}$$

Bu ölçü, tekil konfigürasyonlardan olan uzaklığın bir göstergesi sayılabilir. Bu sebeple, manipüle edilebilirlik ölçüsü ikinci alt görev tekil noktalardan sakınma olduğunda ölçüt olarak kullanılabilir.



Şekil 4.1 Üç Linkli Robot Kolu

Şekil 4.1' de görülen üç linkli robot kolunu düşünelim. *r* noktası konumu, y_1 ilk manipülasyon değişkeni olarak alınır. *r* için hesaplanan manipüle edilebilirlik ölçüsü, ikinci alt görev için ölçüt olarak alınır. Manipüle edilebilirlik ölçüsü ve üç linkli robot kolunu için elde edilen J_1 jakobian matrisi kullanıldığında,

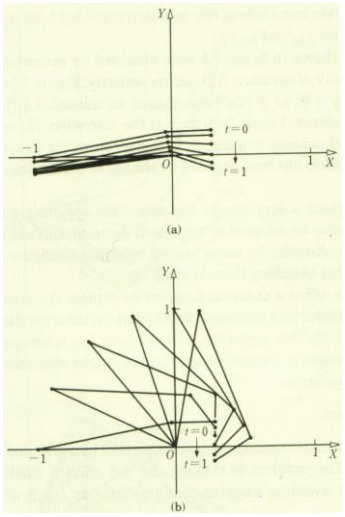
$$V(q) = \sqrt{\det J_1 J_1^{\mathrm{T}}} \tag{4.19}$$

elde edilir. Bu yüzden, (4.14) eşitliğindeki ξ_1 terimi aşağıdaki eşitlik ile verilir;

$$\xi_1 = \frac{1}{2}\sqrt{\det J_1 J_1^{\mathrm{T}}} \sum_{i,j=1}^{2} q_{ij}\,(J_{1il} J_{1j}^T + J_{1jl} J_{1i}^T) \qquad ,l = 1,2,3 \tag{4.20}$$

Buradaki q_{ij} terimi ($J_1 J_1^{\mathrm{T}}$)' in ters matrisinin (*i*,*j*) nci elemanını, J_{1i} terimi J_1' in *i*.nci satır vektörünü ve J_{1il} terimi J_{1i}' in θ_1' e göre kısmi türev katsayısını göstermektedir. (4.15) eşitliğindeki ξ vektörü (4.20) eşitliğinden elde edilir.

Üç linkli robot kolunun modellenmesiyle oluşan sonuçlar aşağıda Şekil 4.2'de gösterilmiştir.



Şekil 4.2 Tekil Noktalardan Kaçınma Benzetimi

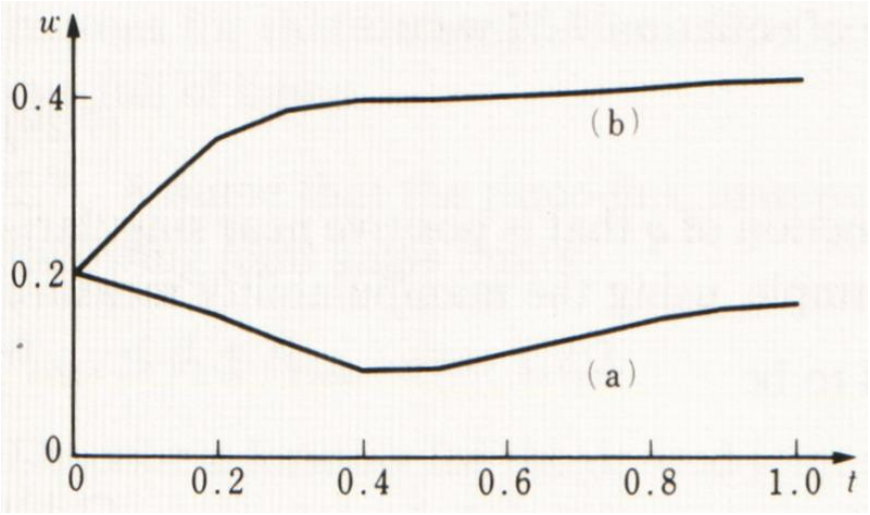
(a) Artıklık Kullanımı Olmadan (k_p=0)

(b) Artıklık Kullanarak (k_p=20)

$r_0 = [x_0,y_0]^T = [0.28,0.17]^T$ uç noktasına karşılık gelen başlangıç konfigürasyonu $q_0 = [180^o, 170^o, -10^o]^T$ dır. İstenilen yörünge, r_0' dan $r_r = [x_0, -0.1]^T$ son konumuna giderken y eksenine paralel bir doğru olan,

$$y_{1d}(t) = \begin{bmatrix} x_0 \\ y_0 - (3-2t)t^2(y_0 + 0.1) \end{bmatrix}, \qquad 0 \le t \le 1 \qquad (4.21)$$

şeklinde verilsin. Şekil 4.2a (4.15(eşitliğindeki k_p' nin 0 olduğu, basit sözde tersi kontrol kanununun uygulandığı ve herhangi bir engelden sakınma çabası olmadığı durumdaki, kol konfigürasyonundaki değişimi göstermektedir. Şekil 4.2b de ise k_p=20 için, manipüle edilebilirlik ölçüsü büyük tutulmaya çalışılarak tekil noktalardan sakınılmıştır. Şekil 4.3, robot kolunun Şekil 4.2a ve Şekil 4.2b durumlarındaki manipüle edilebilirlik ölçüsünün değişimini göstermektedir. Şekilden de görülebildiği gibi artıklık kullanımı tekil noktalardan kaçınmada oldukça etkilidir.



Şekil 4.3 Manipüle Edilebilirlik Ölçüsünün Yörüngesi

(a) Artıklık Kullanımı Olmadan (k_p=0)

(b) Artıklık Kullanarak (k_p=20)

4.3.2 Mekanik Eklem Sınırlarından Sakınma

Bu tip sakınma için maksimize edeceğimiz ölçüt, mekanik eklem sınırlarına olan uzaklıktır. q_{iM} maksimum eklem sınırını, q_{im} minimum eklem sınırını, $\overline{q}_i$ ise eklem aralığının orta noktasını göstermek üzere, mekanik eklem sınırlarına olan uzaklık;

$$V(q) = -\frac{1}{2n}\sum_{i=1}^{n}\left(\frac{q_i - \overline{q}_i}{q_{iM} - q_{im}}\right)^2 \tag{4.22}$$

şeklinde tanımlanabilir. Bu ölçüt (4.14) eşitliğinde yerine yazılarak ξ_1, ξ_1 de (4.12) eşitliğinde yerine yazılırsa k_1 bulunabilir. Bulanan k_1, (4.5) denkleminde yerine konulursa mekanik eklem sınırlarından sakınmayı sağlayan bir çözüm bulunmuş olur.

4.4 Engellerden Sakınma Uygulamaları

Bu uygulamadaki amaç, engelli bir ortamda robotun hareketine izin verecek kontrol algoritmaları sağlamaktır. Engellerden kaçınmak için doğal yöntem, robot kolunun hiçbir engelle temas halinde olmadığı bir konfigürasyonda, robot kolunu engellerden uzakta hareket ettirmektir. Uç noktasının hareketini değiştirmeden, robot kolunu yeniden konfigüre ederek, çarpışmadan bağımsız (collision-free) konfigürasyona getirme işi sadece robot kolunun artık serbestlik derecesine (DOF) sahip olması

durumunda yapılabilir. Robot kolunun esnekliği artık serbestlik derecesi sayısı gibi artıklık derecesine bağlıdır. Robot kolu engellerle pek çok potansiyel çarpışma (olasılığının) bulunduğu bir ortamda çalışırken, yüksek artıklık derecesi daha özel bir öneme sahip olur. Engellerden sakınma için bazı lokal yöntemler geliştirilmiştir [51,40,16,22,12,34,31,95]. Bu lokal yöntemler, engellerden sakınmaya bir kontrol problemi olarak bakar. Amaçları, daha yüksek seviyeli (global) çarpışmadan bağımsız yörünge planlaması değil, düşük seviyeli kontrol yeteneklerinin kullanımını gerçekleştirmektir. Örnek olarak bu yöntemler, engel hareket ederse veya pozisyonu tam olarak bilinmiyorsa yörüngeyi değiştirmek için sensör bilgisini kullanabilirler.

Engellerden sakınma problemini kinematik seviyede çözen pek çok lokal yöntemdeki [51,22,41,31,34,84] gibi, bu çalışmada da, robot kolunun gövdesindeki engele yakın tüm noktalar, engelden ters yöndeki hareket elemanı olarak atanırlar. Bu çalışmadaki vurgu, hareketten kaçınma tanımı olarak verilmiştir. Kaçınma hareketi genellikle kartezyen uzayında tanımlanır. Engellerden sakınma tipik olarak tek boyutlu problem olarak alındığında, her kritik nokta için tek boyutlu işlemsel uzay kullanılır. Bu yüzden, genel olarak yeterli artıklık olmasa bile bazı tekil nokta problemlerinden kaçınılabilir. Ayrıca, bu çalışmada tam çözümden daha hızlı olan yaklaşık hesaplama hareketi kullanılmıştır [103]. Bu uygulamadaki bir diğer önemli konu ise, robot kolunun komşuluğunda anlık pek çok engel varken engellerden sakınmanın nasıl gerçekleştirileceğidir. Bu sorunu çözmek için robotun komşuluğundaki tüm engelleri hesaba katan bir algoritma kullanılmıştır [103]. Kullanılan kontrol algoritmalarının verimliliği, mekanik sınırlar içinde kalmak koşuluyla Mitsubishi PA 10 robot kolu üzerinde denenmiştir.

Bu çalışmada üzerinde çalışılan robotik sistemleri, seri robot kollarıdır. n eklem uzayı m görev uzayından büyük olan artık sistemler dikkate alınmıştır. n ile m arsındaki fark r ile gösterilir ve artıklık olarak adlandırılır, $r = n-m$. Bu tanımlamadaki artıklık sadece robot kolunun kendisinin özelliği değil aynı zamanda görevin de özelliğidir. Bu durum, artık olmayan bir robot kolunun özel bir görev için artık hale gelebileceği anlamına gelir.

i) Kinematik

Robot kolu konfigürasyonu *n* boyutlu *q* eklem pozisyonları vektörü ve uç noktası pozisyon (ve yönelimi) *m* boyutlu *x* pozisyonları (ve yönelimleri) vektörü ile gösterilsin. Böylelikle, kinematik aşağıdaki eşitliklerle tanımlanabilir;

$$x = f(q) \tag{4.23}$$

$$\dot{q} = J^{\dagger}\dot{x} + N\dot{\phi} \tag{4.24}$$

$$\ddot{q} = J^{\dagger}(\ddot{x} - \dot{J}\dot{q}) + N\ddot{\phi} \tag{4.25}$$

Bu eşitliklerdeki, *f* robot kolu ileri yöndeki kinematiğini gösteren *m* boyutlu vektör fonksiyonu, *J mxn* boyutlu robot kolu jakobian matrisi, $J^{\dagger}$ *J* jakobian matrisinin genelleştirilmiş tersi, *N* ($N = I - J^{\dagger}J$) ise *J'* nin boş uzayındaki izdüşüm vektörüdür.

ii) Kinematik Kontrol

Artık robot kolları tarafından gerçekleştirilen pek çok görev farklı önceliklerle bazı alt görevlere ayrılır. Bu çalışmada, ana görev kadar bahsedilen en yüksek öncelikli görev uç noktasının görev uzayındaki konumlanması diğer alt görevler ise (artıklık derecesi yeteri kadar fazla ise) engellerden sakınma ve diğer ek görevler olarak düşünülmüştür.

Hız kontrolü için aşağıdaki kinematik kontrolcü kullanılabilir;

$$\dot{q} = J^{\dagger}\dot{x}_c + N\dot{\phi} \tag{4.26}$$

Bu denklemdeki, $\dot{x}_c$ ve $\dot{\phi}$ sırasıyla görev uzayı kontrol kuralını ve keyfi eklem hızlarını göstermektedir. $\dot{x}_c$ görev uzayı kontrolü, e $e = x_d - x$ izleme hatasını, $\dot{x}_c$ istenilen görev uzayı hızını, K_p sabit kazanç matrisini göstermek üzere, aşağıdaki şekilde seçilebilir.

$$\dot{x}_c = \dot{x}_c + K_p e \tag{4.27}$$

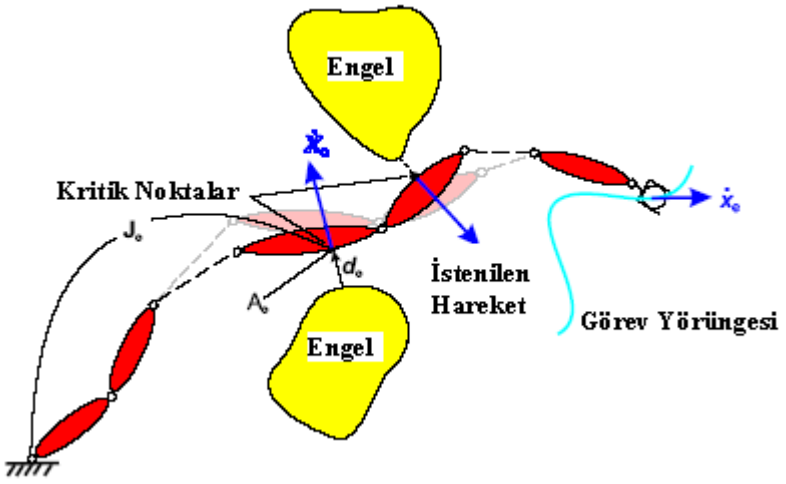
Ek alt görevi yerine getirmek için $\dot{\phi}$ hızı kullanılır. *p* istenilen performans ölçütünü gösteren fonksiyon olsun. Böylelikle *p'* yi optimize etmek için $\dot{\phi}$ yi,

$$\dot{\varphi} = k_p \nabla p \tag{4.28}$$

olarak seçebiliriz. Buradaki ∇p , p' nin gradyanı k_p ise kazançtır.

4.4.1 Engellerden Sakınma Stratejisi

Engellerden sakınma stratejisi robot kolu üzerindeki engellere yakın noktaları saptamak ve daha sonra bu noktaları engellerden uzağa hareket eden hareket elemanı olarak belirlemektir. (Şekil 4.4) Robotun hareketi sadece, robotun en azından bir parçası bir engelin kritik komşulunda bulunması durumunda (uzaklığın minimum uzaklıktan küçük olması), değiştirilir. Bu çalışmada, robot kolu gövdesindeki bu engellere en yakın noktalar kritik noktalar olarak gösterilecektir. Ayrıca, uç noktasının hareketinin bir engel tarafından bozulmadığı varsayılacaktır. Aksi takdirde, görevin uygulanması yarıda kesilmeli ve daha yüksek dereceli yörünge planlamasıyla uç noktasının istenilen hareketini tekrar hesaplanmalıdır. Engellerden sakınmanın çevrimiçi (on-line) yapıldığı durumda, engellerin yerini ustalıkla tam olarak bilmek zorunlu değildir. Robot kolunun yapısal olmayan (unstructured) ve/veya dinamik bir ortamda çalışmasını sağlamak için engellerin konumlarını belirlemek veya engellerle robot gövdesi arasındaki uzaklıkları ölçmek için bazı sensörler kullanılmalıdır. Bu çalışmada, engellerden sakınma çevrimdışı (off-line) olarak çözülmüş daha sonra sonuçlar Mitsubishi PA 10 robot koluna uygulanmıştır.



Şekil 4.4 Bazı Engellerin Varlığında Robot Kolu Hareketi

Önerilen hız stratejisi engellerden sakınma probleminin kinematik düzeyde olduğunu varsaymaktadır. $\dot{x}_\xi$ uç noktasının istenilen hızı ve A_0 bir engelin komşuluğundaki

kritik nokta olsun. Olası bir çarpışmadan kaçınmanın bir olasılığı A_0' a [51]'de önerildiği gibi engelden uzağa hareket ettirecek bir hız atamaktır. Bu nedenle, uç noktasının hareketi ve kritik nokta aşağıdaki eşitliklerle tanımlanabilir;

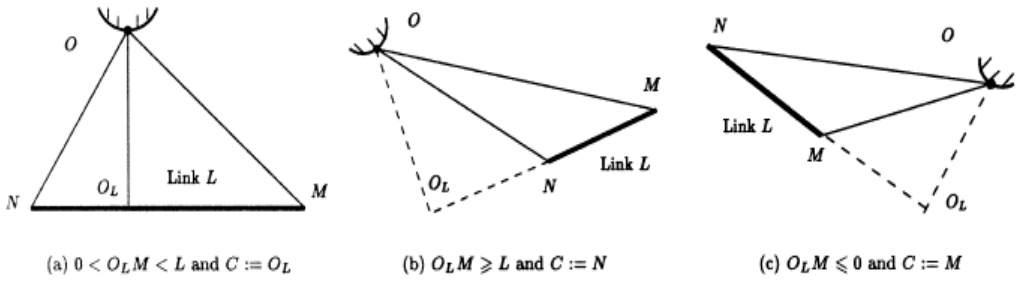
$$J(q)\dot{q} = \dot{x}_c \tag{4.29}$$

$$J_0(q)\dot{q} = \dot{x}_0 \tag{4.30}$$

Buradaki J_0, A_0 noktasındaki jakobian matrisidir. Bu iki eşitliğin ortak çözümünü bulmak için bazı yollar vardır.

4.4.1.1 Kritik Noktaların Yerini Saptama

Engellerden kaçınma için başlangıç adımı genellikle, herhangi bir $t \in [t_0, t_f]$ anında robot kolunun L linki ile O engel noktası arasındaki $OC(t, \theta)$ uzaklığını hesaplayarak, C kritik noktasının yerini saptamaktır [106].



Şekil 4.5 L Robot Kolu Linkine Göre O Engel Noktasının Üç Olası Konumuna Karşılık Gelen C Kritik Noktasının Yeri

Şekil 4.5' de gösterildiği gibi, L linki üzerindeki C kritik noktası için O engel noktasının değişik konumlarına karşılık gelen üç durum söz konusudur. Geometrik özellikten dolayı,

$$OO_L^2 = OM^2 - O_L M^2 = ON^2 - O_L N^2$$

eşitliği vardır. Burada, $NO_L = NM - O_L M$ dir ve link uzunluğu $NM = L > 0$ dir. İlk iki eşitliği birleştirirsek, $OM^2 - O_L M^2 = ON^2 - (NM - O_L M)^2$ ve

$$O_L M = (OM^2 + NM^2 - ON^2)/(2NM) \tag{4.31}$$

eşitliklerini elde ederiz.

(4.31) nolu eşitlikteki $O_L M$ ' nin işaret ve genliğine göre, Şekil 4.5'de de gösterildiği gibi üç durum söz konusudur:

Durum a) $0 < O_L M < L$: C kritik noktası O_L' dir ve OC en kısa uzaklığı OO_L' dir.

Durum b) $O_L M \geq L$: C kritik noktası N' dir ve OC en kısa uzaklığı ON' dir.

Durum c) $O_L M \leq 0$: C kritik noktası M' dir ve OC en kısa uzaklığı OM' dir.

Yukarıda anlatılan, C kritik noktasının yerini O engel noktası üzerinden dinamik olarak saptama yöntemi, model tabanlı ve sensör tabanlı robot kolu kontrolü için basit ortak gereksinimdir. Model tabanlı kontrolde, çevreye ilişkin kartezyen dünya haritası bilgisi kullanılan bir önceliktir. Bu yüzden O engel noktası, robot kolu linki **(**sıfır kalınlığında çubuk ile gösterilen) ile engel nesnesi (genellikle, dışbükey çokyüzlüler, küreler veya elipsoidlerle gösterilen) arasındaki çevrimiçi veya çevrimdışı uzaklığın minimizasyonu yoluyla belirlenir. [51,14,31,41,50]. Sensör tabanlı kontrolde, O engel noktası sensör bileşim teknolojilerinin sentetik bilgisi ile belirlenir (Örneğin, kamera, ultrasonik sensör) [51,31,62,9].

4.4.1.2 Tam Çözüm

$\dot{x}_c$ yerine $\dot{x}_e$ yazıp, (4.26) ve (4.30) eşitliklerini birleştirirsek [51,65],

$$\dot{\varphi} = (J_0 N)^{\dagger} (\dot{x}_0 - J_0 J^{\dagger} \dot{x}_e) \tag{4.32}$$

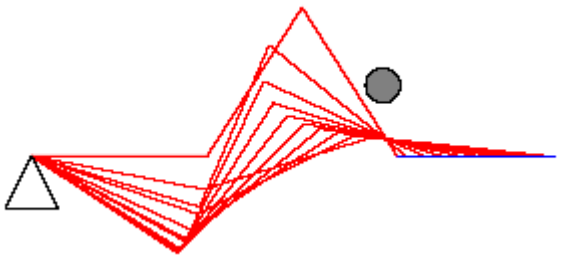
eşitliğini elde ederiz. Bu sonucu (4.26) eşitliğinde kullanırsak N hem hermityan hem de idempotent olduğu için $\dot{q}$ için son çözüm formu,

$$\dot{q} = J^{\dagger} \dot{x}_c + (J_0 N)^{\dagger} (\dot{x}_0 - J_0 J^{\dagger} \dot{x}_e) \tag{4.33}$$

şeklindedir. Bu eşitlikteki terimlerin anlamı kolayca açıklanabilir. İlk $J^{\dagger} \dot{x}_c$ terimi istenilen uç noktası hızı için gerekli eklem hareketini garanti eder (burada herhangi görev uzayı izleme hatasını karşılamak için $\dot{x}_e$ yerine $\dot{x}_c$ kullanılmıştır). $J^{\dagger} \dot{x}_c$'dan sonraki terimlerin tamamı A_0 noktasının hareketini gösteren $\dot{q}_n$ homojen çözümüdür. $J_0 J^{\dagger} \dot{x}_e$ terimi, uç noktası hareketi nedeniyle A_0 noktasındaki hız, $J_0 N$ matrisi ise

istenilen kritik nokta hızını, kritik noktanın işlemsel uzayından eklem uzayına dönüştürmek için kullanılan terimdir. Bu çözüm, istenilen $\dot{x}_0$'ya tam olarak ulaşmayı sadece robot kolunun artıklık derecesinin yeterli olması durumunda garanti eder.

J_0N matrisi, A_0 kritik noktasının kinematiği ile tüm robot kolunun boş uzay matrisini birleştirir ve bu nedenle J_0N matrisinin özellikleri sistemin engellerden kaçınmak için esnekliğini tanımlar. Burada, J_0N matrisinin özelliklerinin sadece A_0 noktasının konumuna değil aynı zamanda kritik nokta ile birleştirilen işlemsel uzayın tanımına da bağlı olduğu gösterilmek istenmektedir. Genellikle bütün kritik noktaların kazrtezyen uzayına ait olduğu varsayılır. Bu nedenle, $\dot{x}_0$ hızı 3 boyutlu bir vektördür ve J_0N matrisi $3xn$ boyutundadır. Bu durum aynı zamanda, bir noktanın engelden uzağa hareketi için 3 artıklık derecesine gereksinimi gerektirir. Sonuç olarak, iki artıklık dereceli bir robot kolunun engellerden sakınmak yeteneği olmadığı gözükmektedir. Tabii ki, bu doğru değildir. Örneğin, Şekil 4.6'da gösterilen bir doğru üzerinde hareketi sağlanan 3 serbestlik dereceli bir düzlemsel robot kolunu düşünelim. Bu düzlemsel durumda, görev uzayı (x ve y olarak) 2 boyutlu ve robot kolu bir artıklık derecesine sahiptir. $\dot{x}_0$ hızını aynı uzayda 2 boyutlu vektör biçiminde uç noktasının hızı olarak tanımlamak, J_0N matrisinin boyutunun $2x3$ olmasını gerektirir. Ayrıca, bir artıklık derecesi nedeniyle $\dot{x}_0$ hız vektörünün elemanları bağımsız değildir. Bu yüzden, J_0N matrisinin rankı 1 dir ve $(J_0N)^{\dagger}$ sözde tersi mevcut değildir.



ŞEKİL 4.6 Bir Doğruyu İzleyen ve J_{d0} Jakobianını Kullanarak Engelden Sakınan 3 Serbestlik Dereceli Düzlemsel Robot Kolu

Engellerden sakınma stratejisi, sadece kritik nokta ile engel üzerindeki en yakın noktayı birleştiren doğru yönündeki hareket gerektirir. Bu tek boyutlu kısıtlamadır ve genellikle engelden sakınmak için sadece bir artıklık derecesi gerekir. Böylelikle, J_0 jakobianını aşağıdaki şekilde tanımlamayı önerebiliriz.

d_0 Şekil 4.4'de görüldüğü gibi engel ile robot kolu üzerindeki en yakın noktaları birleştiren vektör olsun ve A_0'daki işlemsel uzay d_0 yönündeki tek boyutlu uzay olarak tanımlansın. Bu durumda $\dot{\phi}$ eklem uzayı hızları ile d_0 yönündeki hızı ilişkilendiren jakobian,

$$J_{d0} = n_0{}^T J_0 \tag{4.34}$$

şeklinde hesaplanabilir. Bu eşitlikteki J_0 kartezyen uzayında tanımlanan jakobiandır ve n_0 ($n_0 = \frac{d_0}{\|d_0\|}$) d_0 yönündeki birim matristir. Bu durumda, J_{d0} matrisinin boyutu *1xn* dir ve $\dot{x}_0$ ve $J_{d0}J^{\dagger}\dot{x}_e$ birer skalere dönüşür. Sonuç olarak, $(J_{d0}N)^{\dagger}$ hesaplaması daha hızlıdır. Çünkü, aşağıdaki şekilde tanımlanan ($J_{d0}N$)' ın *"Moore-Penrose"* sözde tersinde,

$$\left(\mathrm{J}_{\mathrm{d0}}N\right)^{\dagger} = \left(\mathrm{J}_{\mathrm{d0}}N\right)^{\mathrm{T}}\left(\mathrm{J}_{\mathrm{d0}}N\left(\mathrm{J}_{\mathrm{d0}}N\right)^{\mathrm{T}}\right)^{-1} = N\mathrm{J}_{\mathrm{d0}}{}^{T}\left(\mathrm{J}_{\mathrm{d0}}N\mathrm{J}_{\mathrm{d0}}{}^{T}\right)^{-1} \tag{4.35}$$

($\mathrm{J}_{\mathrm{d0}}NJ_{d0}{}^{T}$) terimi skaler olduğu için herhangi bir matris tersi alma zorunluluğumuz yoktur. Tekrar Şekil 4.6' daki örneğe dönecek olursak $\left(\mathrm{J}_{\mathrm{d0}}N\right)^{\dagger}$ sözde tersi mevcuttur ve robot kolu birincil görevi uygulayabilir ve aynı anda şekilde gösterilen engelden sakınabilir.

Artık robot kollarının kontrolünde önemli bir konu jakobian matrisinin rank kaybettiği tekil konfigürasyonlardır. Genellikle, sadece tüm robot kolunun konfigürasyonu ile ilgilenilir, fakat engellerde sakınma konusunda A_0 kritik noktası tarafından tanımlanan iki robot kolu alt gövdesinin tekil noktalarını düşünülmelidir. A_0 tarafından tanımlanan bu yapılar: (a) taban ile A_0 noktası arasındaki robot kolu parçası, (b) A_0 noktası ile uç noktası arasındaki robot kolu parçasıdır. $J^{\dagger}$ uç noktası jakobianının istenilen uç noktası yörüngesi boyunca tekil olmadığı varsayılabilmesine rağmen (aksi taktirde birincil görev yerine getirilemez), bu durum $J_{d0}N$ matrisi için her zaman doğru değildir. Yani, (a) parçası tekil konfigürasyonda iken J_0 tam (full) ranklı değildir ve (b) parçası tekil konfigürasyonda iken J_0 rank yitirmez fakat $J_{d0}N$ tekil hale gelir. Bu nedenle, iki durumdan birinde tekil konfigürasyona ulaşıldığında, $\dot{\phi}_n$ değerleri kabul edilemeyecek kadar büyük hale gelebilir. Robot kolunun yapısal olmayan bir çevrede hareket etmesi varsayıldığında,

pratikte $J_{d0}N$' in ne zaman tekil hale geleceğini bilmek imkânsızdır. Bu nedenle, J_0 yerine önerilen J_{d0} kullanıldığında sistemin önemli ölçüde daha az tekil noktaya sahip olması oldukça önemli bir avantajdır.

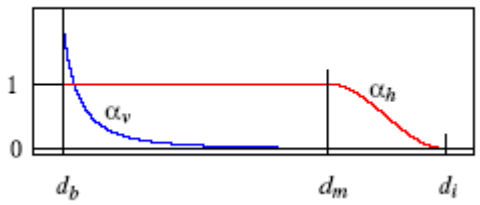
Engelden sakınma algoritmasının verimliliği $\dot{x}_0$ istenilen kritik nokta hızının seçimine de bağlıdır. $\dot{x}_0$'ın aşağıdaki gibi engel uzaklığına göre değiştiği önerilebilir.

$$\dot{x}_0 = \alpha_v v_0 \tag{4.36}$$

Bu eşitlikteki v_0 nominal hızı, α_v aşağıdaki gibi tanımlanan engelden sakınma kazancını,

$$\alpha_v = \begin{cases} \left(\dfrac{d_m}{\|d_0\|}\right)^2 - 1 & , \|d_0\| < d_m \\ 0 & , \|d_0\| \geq d_m \end{cases} \tag{4.37}$$

d_m ise Şekil 4.7'de görülebilen kritik uzaklığı göstermektedir. Eğer engel yakınsa ($\|d_0\| \leq d_b$), ana görev başarılamaz. d_b uzaklığı robot kolunun dinamik özelliklerine bağlıdır ve $\dot{d}_0$ göreli hızının bir fonksiyonu olabilir. Düzgün geçişleri garanti etmek için $\dot{x}_0$ genliğinin d_m de sıfır olması önemlidir.



Şekil 4.7 Engelden Uzaklığa Karşı α_v Engelden Sakınma Kazancı ve α_h Homojen Terim Kazancı

v_0 nominal hızının seçimine özel bir özen gösterilmelidir. v_0'in büyük değerleri gereksiz yüksek hızlara sebep olur ve bu nedenle robot kolu engelden uzağa hareket eder. Bu hareket, robot kolu komşuluğunda fazla engel varsa problemlere neden olabilir. Yani, robot kolu engeller arasında sıçrayabilir. Diğer taraftan, küçük v_0 değerleri için robot kolunun kritik noktaları engelden uzağa hareket edemez.

Düz hareket için, [51]'de önerildiği gibi homojen çözümün miktarını değiştirmek, toplam çözüme dâhil edilmiştir.

$$\dot{q}_{EX} = J^{\dagger}\dot{x}_c + \alpha_h (J_{d0}N)^{\dagger}(\dot{x}_0 - J_{d0}J^{\dagger}\dot{x}_e) \tag{4.38}$$

Bu eşitlikteki α_h aşağıdaki şekilde seçilebilir;

$$\alpha_h = \begin{cases} 1 & , \|d_0\| \le d_m \\ \frac{1}{2}(1-\cos(\pi \frac{\|d_0\| - d_m}{d_i - d_m} & , d_m < \|d_0\| < d_i \\ 0 & , d_i \le d_m \end{cases} \tag{4.39}$$

Buradaki d_i, engelin harekete etkilediği uzaklıktır. Şekil 4.7'den görülebileceği gibi d_b ile d_m arasındaki bölgede, tüm homojen çözüm hareket özelliğine eklenmiştir ve bu bölgede engelden sakınma uzaklıkla ters orantılıdır. d_m ile d_i arasında sakınma hızı sıfırdır ve homojen kısmın sadece bir parçası eklenmiştir. Eğer engel hareket etmiyorsa, homojen çözümün uç noktası hareketi nedeniyle kritik noktadaki hareketi dengelediği gibi engelle kritik nokta arasındaki göreli hız da d_i den d_m e doğru azalır. α_v ve α_h'ın uygun seçilmesiyle düzgün hızlar elde edilebilir.

Eşitlik (4.38)'deki kontrol kanunu tek engel için kullanılabilir. Aynı anda birden fazla engelin aktif olması durumunda, süreksiz hızlara ve bazı durumlarda osilasyonlara neden olan en yakın engel (en kötü durum) seçilmelidir. Yani, aktif engeller aralarında değişirken, özel homojen çözümler homojen değildir ve eklem hızlarında süreksizlikler oluşur. Bu davranışı geliştirmek için tüm aktif engellerin ağırlıklandırılmış homojen çözümünün kullanımı önerilmiştir [103].

$$\dot{q}_{EX} = J^{\dagger}\dot{x}_c + \sum_{i=1}^{no} w_i \alpha_{h,i} \dot{q}_{h,i} \tag{4.40}$$

Bu eşitlikteki, n_0 aktif engel sayısını, w_i $\alpha_{h,i}$ $\dot{q}_{h,i}$ terimleri ise sırasıyla ağırlık çarpanı kazanç ve *i.* aktif engel için homojen çözümü göstermektedir. w_i ağırlık çarpanları aşağıdaki şekilde hesaplanabilir:

$$w_i = \frac{d_i - \|d_{o,i}\|}{\sum_{i=1}^{no}\left(d_i - \|d_{o,i}\|\right)} \tag{4.41}$$

Kritik noktalardaki gerçek hızların istenilen hızlardan farklı olmasına rağmen $\dot{q}_{EX}$ kullanımı sistemin davranışını geliştirir ve herhangi bir nokta bir engele diğerlerinden çok daha yakın olduğunda onun ağırlığı bire yaklaşır ve bu özel noktadaki hız istenilene çok yakın olur.

4.4.1.3 Yaklaşık Çözüm

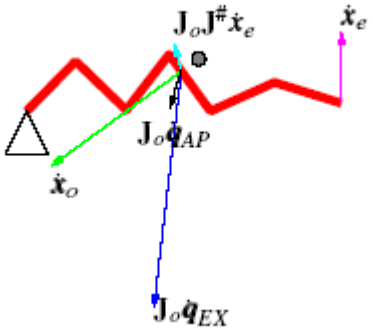
$\dot{q}$ için diğer olası çözüm (4.42) eşitliğindeki ikincil amacı gerçekleyen eklem hızlarını, uç noktasının hareketinin katkısı olmadan, hesaplamaktır.

$$\dot{q} = J_{d0}^{\dagger}\dot{x}_0 \tag{4.42}$$

$\dot{q}$'ı daha sonra (4.26) eşitliğinde yerine koyarsak aşağıdaki eşitliği buluruz.

$$\dot{q}_{AB} = J^{\dagger}\dot{x}_e + NJ_{d0}^{\dagger}\dot{x}_0 \tag{4.43}$$

Bu yaklaşım ($J_{d0}N$) 'ın tekillik probleminden kaçınır [18]. Eşitlik (4.43)'deki formülasyon artıklık derecesi yeterli olsa bile, istenilen $\dot{x}_0$'ı tam olarak elde etmeyi garanti etmez, çünkü genellikle $J_{d0}NJ_{d0}^{\dagger}\dot{x}_0$ terimi $\dot{x}_0$'a eşit olmaz. Şekil 4.8' de değişik stratejiler için hız değerleri gösterilmiştir (Her biri için parametreler eşittir). A_0'daki engelden sakınmak için hedeflenen hız $\dot{x}_0$ vektörü ile gösterilmiştir.



Şekil 4.8 Farklı Yaklaşımlar İçin A_0 Kritik Noktasındaki Kaçınma Hızları Karşılaştırması

Eşitlik 4.33'deki orijinal yöntem kullanıldığında A_0'daki hız tamamen $\dot{x}_0$'dır. $\dot{q}_{EX}$ eklem hızları A_0 noktasındaki $\dot{x}_0$ yönündeki hız elemanını gerektiği gibi temin eder. $\dot{q}_{AP}$ yaklaşık çözümü çoğu durum için $\dot{x}_0$ yönündeki en küçük genlikli hızı verir. Bu

yüzden, robot kolu $\dot{q}_{AP}$ kullanıldığında engele daha yakın hareket eder. Bu çok kritik değildir, çünkü minimal uzaklık aynı zamanda daha yüksek minimal uzaklıklara ulaşmak için arttırılabilen v_0 nominal hızına da bağlıdır. Ayrıca, yaklaşık çözüm pek çok aktif engelin düşünülmesi gerektiğinde, önemli avantajlara sahiptir. Eklem hızları aşağıdaki şekilde hesaplanabilir:

$$\dot{q}_{AP} = J^{\dagger}\dot{x}_c + N\sum_{i=1}^{no} J_{d0,i}^{\dagger}\dot{x}_{0,i} \tag{4.44}$$

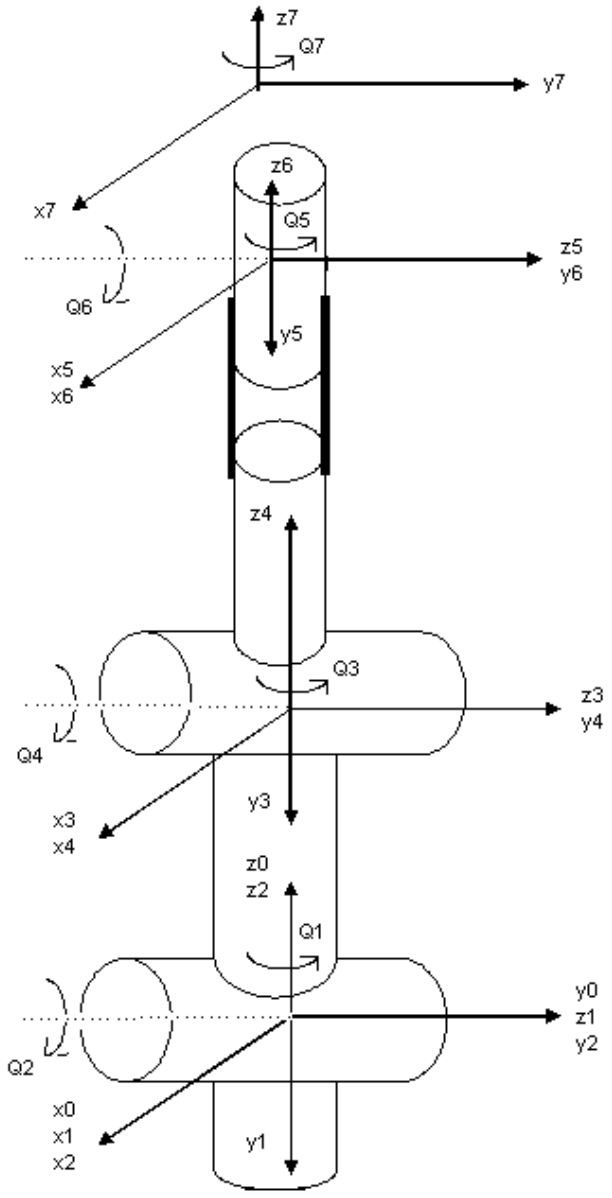
Bu eşitlikteki, n_0 aktif engel sayısını göstermektedir ve N matrisi sadece bir kere hesaplanmaktadır. Fakat $J_{d0,i}^{\dagger}$ sözde tersleri her aktif engel için hesaplanmak zorundadır.

4.4.2 Sonuç

Bu çalışmada anlatılan yaklaşım, hız seviyesindeki artıklık çözünürlüğüne dayanmaktadır. Birincil görev uç noktası yörüngesi tarafından belirlenir ve engelden sakınma için robot kolunun iç hareketi kullanılmıştır. Buradaki amaç, robot kolu üzerindeki engele yakın her noktaya engelden uzaklaşacak yönde bir hız elemanı atamaktır. Engelden sakınmanın, tek boyutlu işlemsel uzayda tanımlanmasının makul olduğu gösterilmiştir. Bu yolla lokal olarak yeterli artıklık olmaması durumunda bazı tekil nokta problemlerinden kaçınılabilir. Ayrıca, J_0 jakobian matrisinin sözde tersinin hesaplanması, matris tersi alma işlemi yerine skaler bölme işlemi içerdiği için daha basittir. Kaçınma hızları için yaklaşık hesaplama kullanımı, hesaplama açısından avantajlıdır ve engelleri anlık olarak değerlendirmek daha kolaydır. Kullanılan algoritmaların verimliliği, bir kinematik artık robot olan Mitsubishi PA 10 benzetimleri ile gösterilmiştir.

5 BENZETİMLER VE DENEY SONUÇLARI

5.1 Mitsubishi PA 10 Robot Kolunun Kinematik Yapısı



Şekil 5.1 Mitsubishi PA 10 Robot Kolunun Eksen Takımlarının Gösterimi

Mitsubishi PA 10 robot kolunun yukarıdaki Şekil 5.1'de görüldüğü gibi eksen takımlarının çıkarılmasıyla robotun aşağıda görülen DH tablosu elde edilmiştir.

Tablo 5.1 Mitsubishi PA 10 Robot Kolunun DH Tablosu

Link	$\mathbf{a_i}$	$\boldsymbol{\alpha_i}$	$\mathbf{d_i}$	$\mathbf{q_i}$
1	0	-90	0	q_1
2	0	90	0	q_2
3	0	-90	d3	q_3
4	0	90	0	q_4
5	0	-90	d5	q_5
6	0	90	0	q_6
7	0	0	d7	q_7

Elde edilen DH değerleri (5.1) eşitliğinde tek tek yerine yazılıp (*i-1*). nci linkden *i.* nci linke olan dönüşüm matrisleri bulunmuştur.

$$T_i^{i-1} = \begin{bmatrix} \cos(q_i) & -\sin(q_i)\cos(\alpha_i) & \sin(q_i)\sin(\alpha_i) & a_i\cos(q_i) \\ \sin(q_i) & \cos(q_i)\cos(\alpha_i) & -\cos(q_i)\sin(\alpha_i) & a_i\sin(q_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

Tabandan *i.* nci linke olan dönüşüm ise (5.2) eşitliğindeki gibi ifade edilmiştir.

$$T_i^0 = T_1^0 T_2^1 T_3^2 ... T_i^{i-1} \tag{5.2}$$

z_{i-1} *i.* nci eklem ekseninin birim vektörünü, p_i *i.*nci eksen takımı orijininin konum vektörünü göstermek üzere, z_i ve p_i'ler T_i^0 dönüşüm matrislerinin sırasıyla üçüncü ve dördüncü sütunlarının ilk üç elamanı olarak elde edilmiştir. Mitsubishi PA 10 robot kolunun tüm eklemleri dönel olduğu için jakobian ifadesi, z_i ve p_i'ler aşağıdaki (5.3) eşitliğinde yerine yazılarak bulunmuştur.

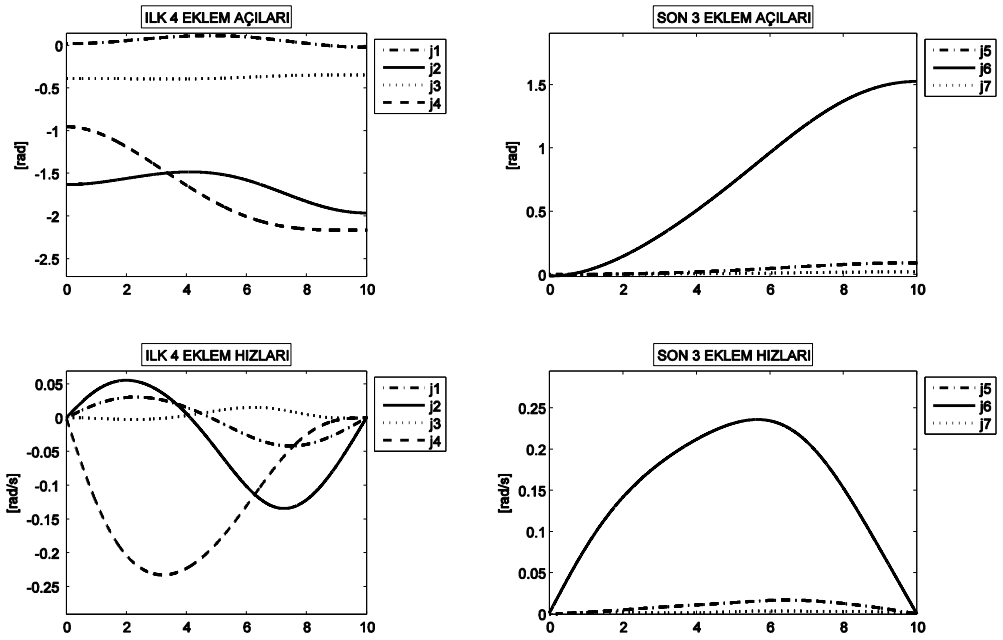
$$J = \begin{bmatrix} z_{i-1} \times (p_n - p_{i-1}) \\ z_{i-1} \end{bmatrix} \tag{5.3}$$

Ayrıca, bu çalışma sırasında yönelim, eksen-açı gösterilimi olarak ele alınmıştır [87].

5.2 Mitsubishi PA 10 Robot Kolu İçin Sınırlamasız Ters Kinematik Çözümler ve Karşılaştırmaları

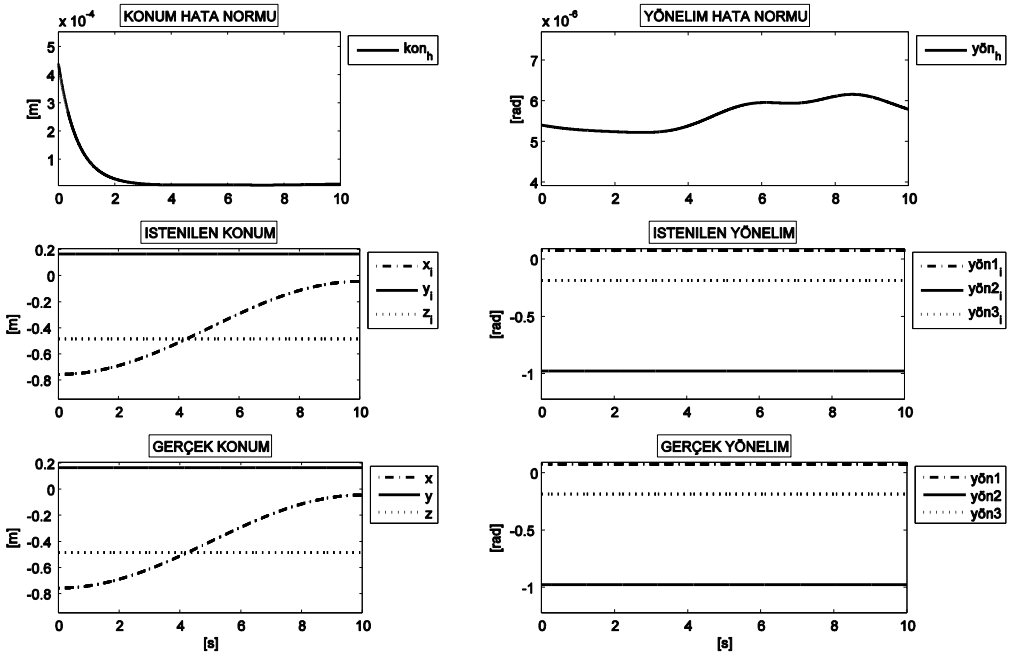
5.2.1 En Küçük Kareler Yöntemi Benzetim Sonuçları

Bu çalışmada benzetim ile çözüm yöntemlerini karşılaştırmak için ilk olarak, robot kolu uç noktasının izlemesi istenilen 5 farklı yörünge belirlenmiştir. Belirlenen ilk yörünge için en küçük kareler yöntemi kullanıldığında, bu yörünge boyunca robotun aldığı eklem açıları ve eklem hızları aşağıdaki Şekil 5.2' de görülebilir.



Şekil 5.2 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Eklem Açı ve Hızları

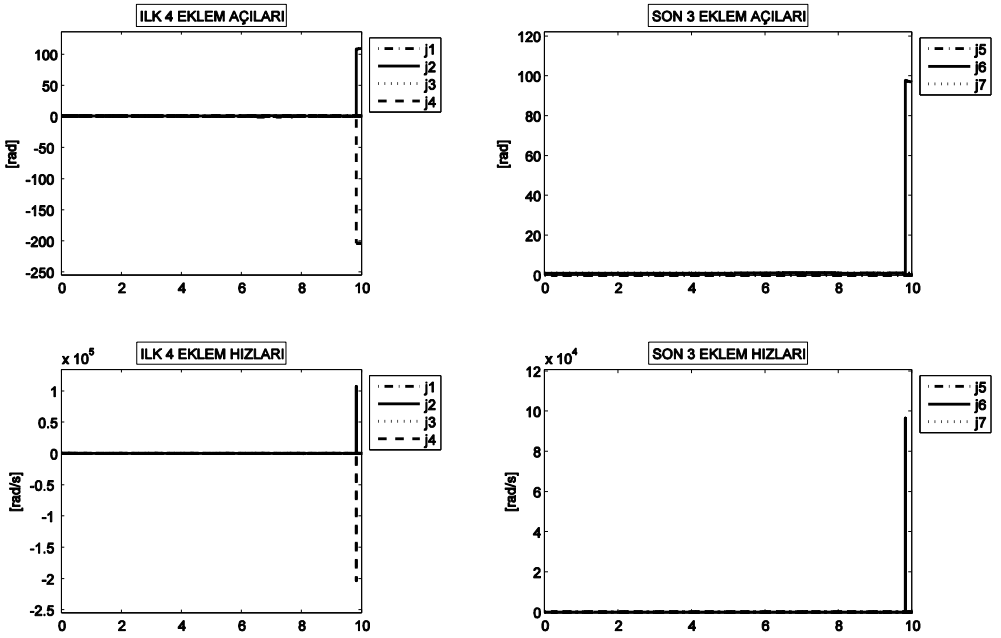
Aşağıdaki Şekil 5.3' de ise aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ve bu ikisi arasındaki fark nedeniyle oluşan konum ve yönelim hatalarının normları görülebilir.



Şekil 5.3 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

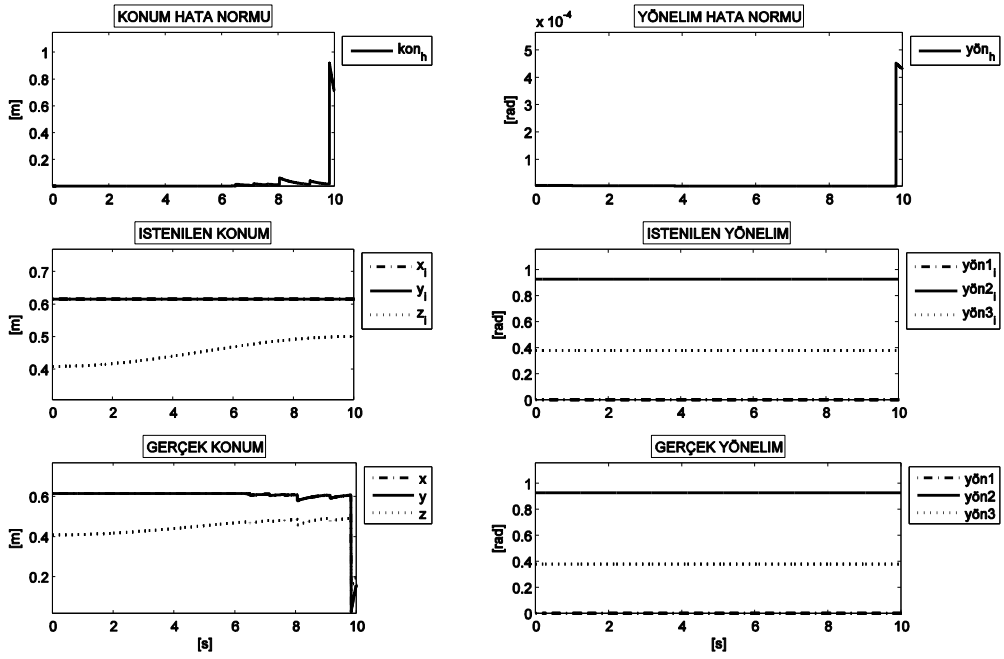
Sonuçları incelersek; konum hatası 10^{-4} mertebesinde, yönelim hatası ise 10^{-6} mertebesinde olmaktadır. Elde edilen sonuçlar oldukça tatminkardır. 3. bölümde de bahsedildiği gibi en küçük kareler yöntemi tekil noktalar ve komşuluğundaki bölgelerde yetersiz kalmaktadır. Elde ettiğimiz sonuçların yeterli olmasından yola çıkarak, yörünge 1 için robotun tekil noktalara girmediğini söyleyebiliriz.

Belirlenen yörünge 2 için, robotun aldığı eklem açıları ve eklem hızları aşağıda Şekil 5.4' de görülmektedir.



Şekil 5.4 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Eklem Açı ve Hızları

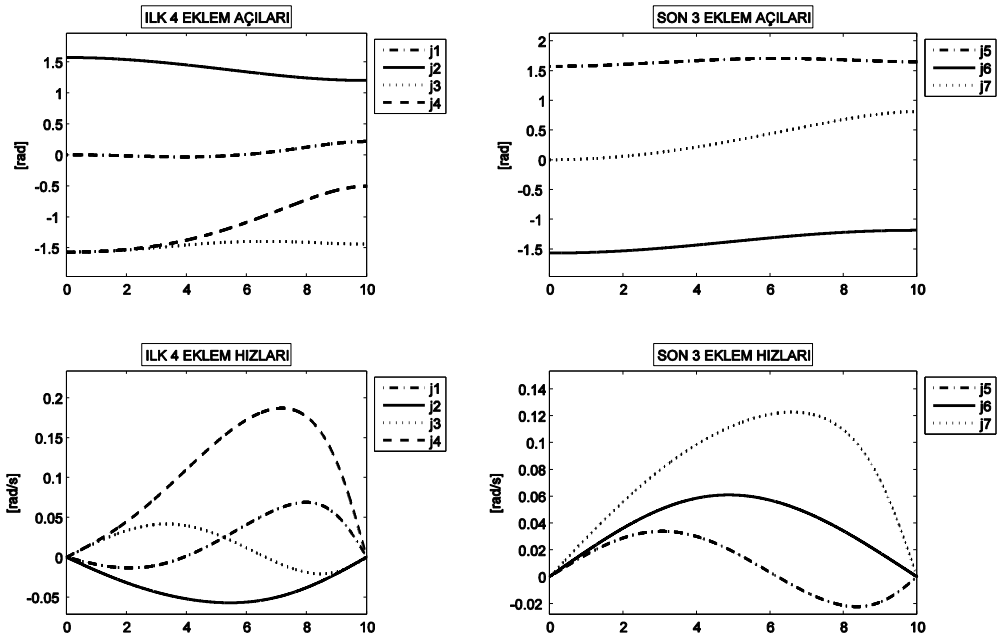
Şekil 5.5' de ise aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları görülmektedir.



Şekil 5.5 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

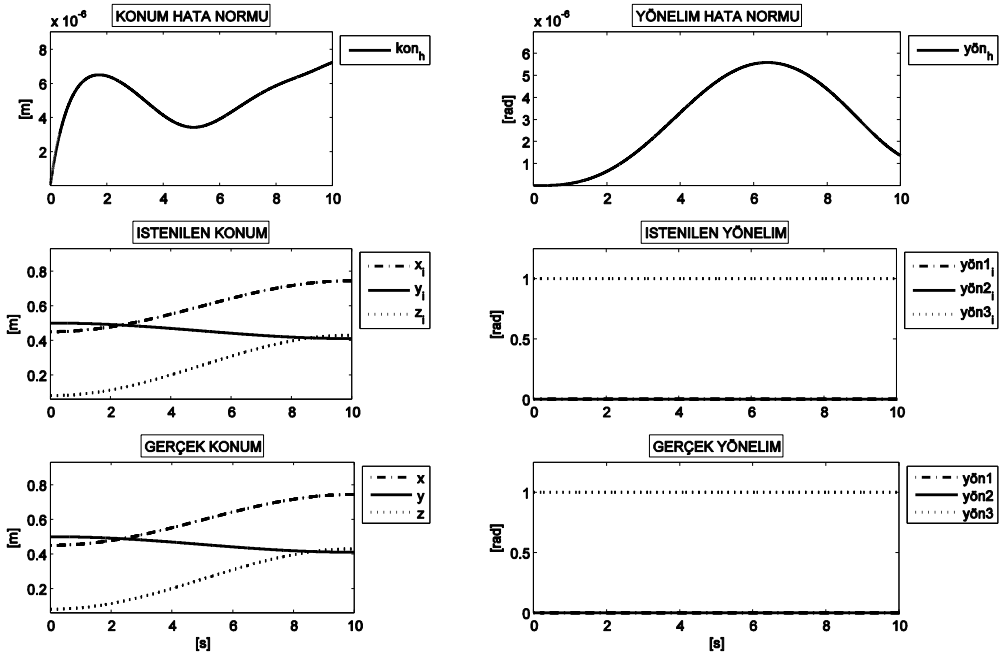
Sonuçları incelersek; benzetimin son anlarında gerek eklem açılarında, gerek eklem hızlarında gerekse hata normlarında bir sapma meydana gelmiştir. Eklem açıları 100 [rad], eklem hızları ise 10^5 [rad/sn] mertebesine çıkmaktadır. Bu durum, yörünge 2 içerisinde tekil noktalarla karşılaşıldığını göstermektedir. Elde edilen sonuçlar en küçük kareler yönteminin tekil noktalar karşısındaki yetersizliğinin bir ispatıdır.

Üçüncü yörünge için, robotun aldığı eklem açıları ve hızları aşağıdaki Şekil 5.6' da görülmektedir.



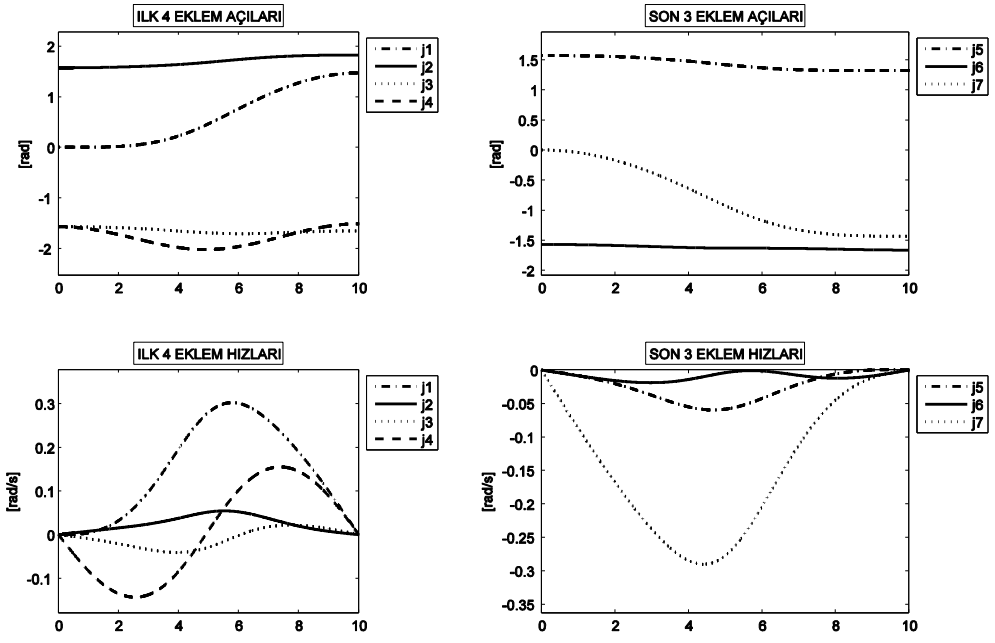
Şekil 5.6 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.7' deki gibidir.



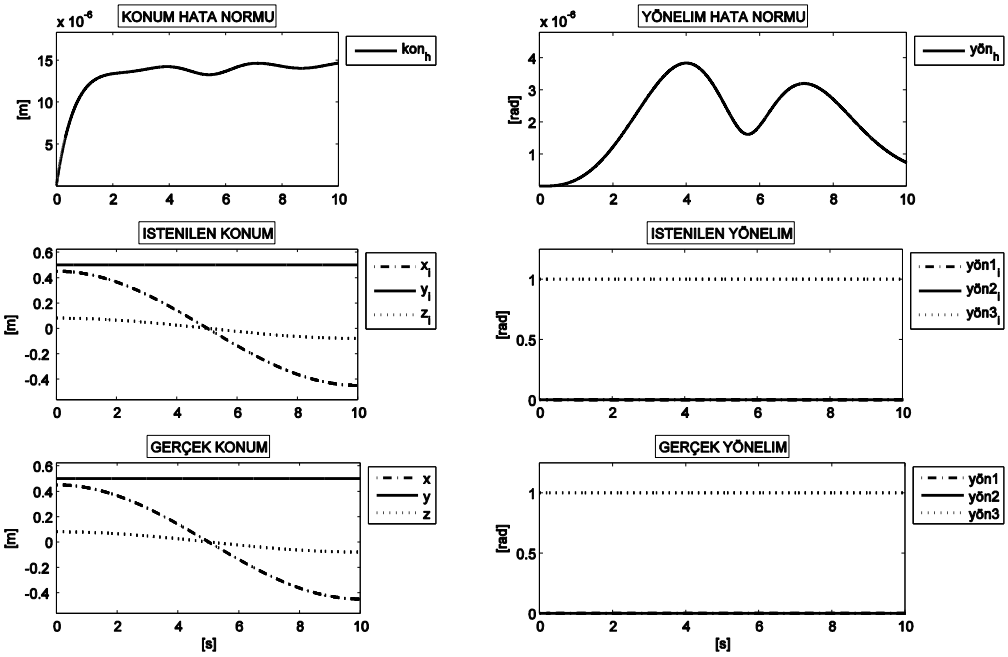
Şekil 5.7 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Dördüncü yörünge için, robotun aldığı eklem açıları ve hızları Şekil 5.8' deki gibidir.



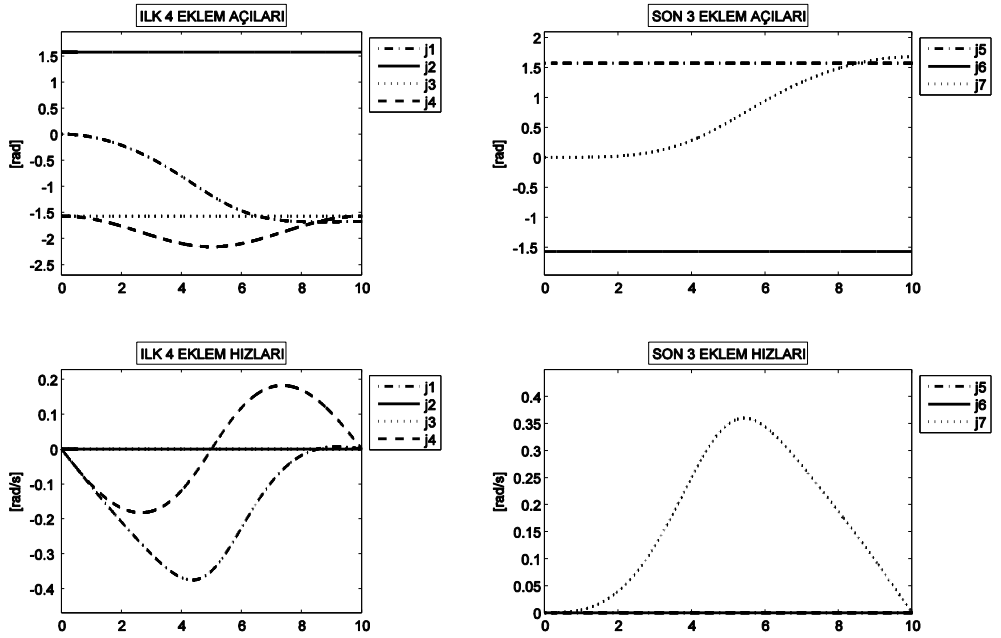
Şekil 5.8 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.9' da görülebilir.



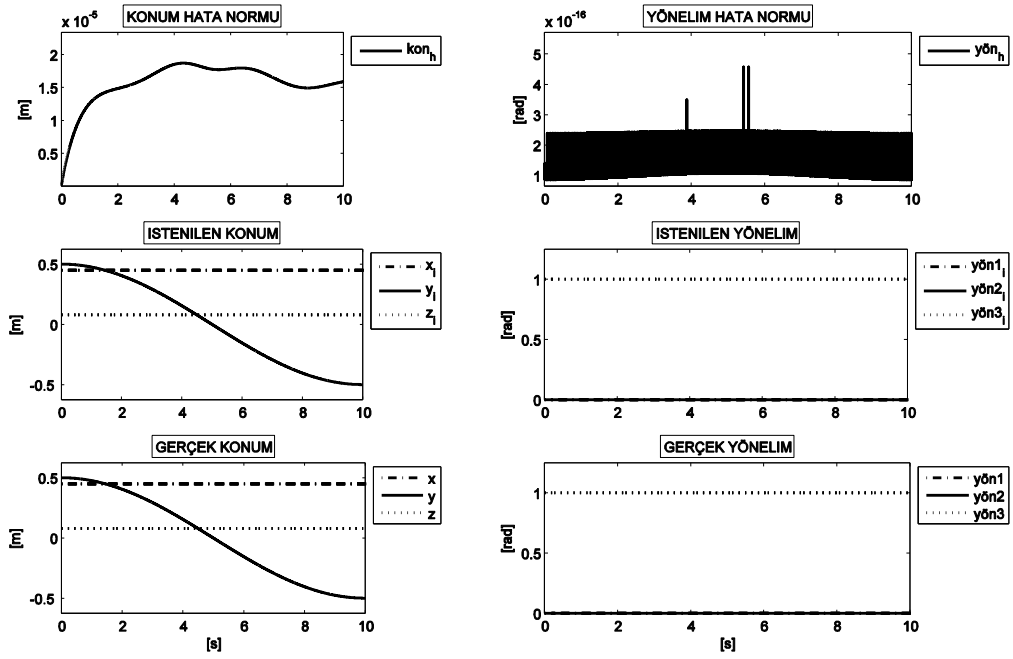
Şekil 5.9 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Belirlenen beşinci ve son yörünge için, robotun aldığı eklem açıları ve hızları aşağıdaki Şekil5.10' da görülmektedir.



Şekil 5.10 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.11' deki gibidir.



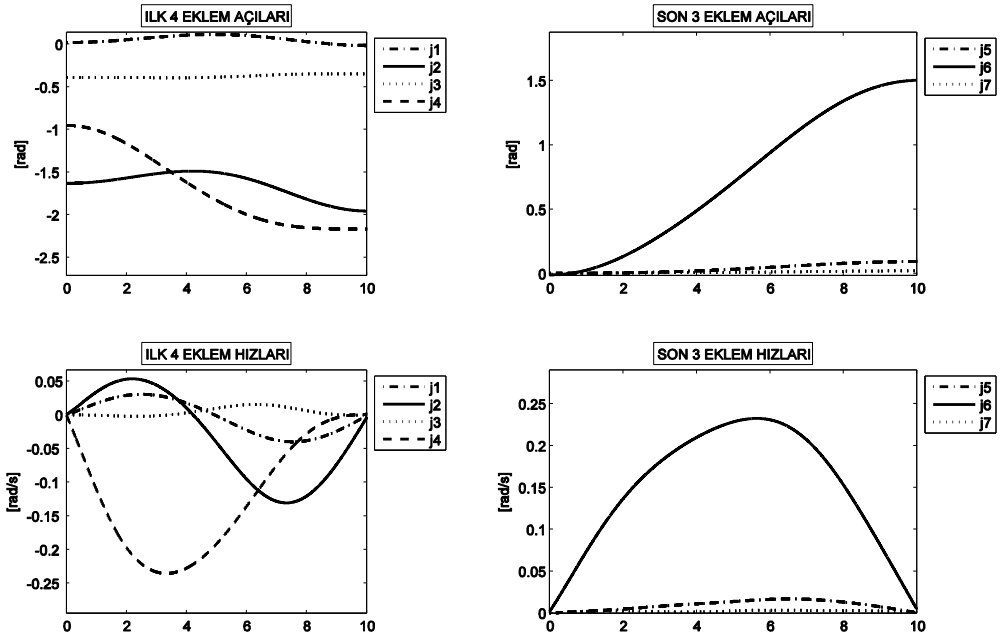
Şekil 5.11 En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Sonuçları yorumlarsak; üçüncü, dördüncü ve beşinci yörüngeler için konum hata normları sırasıyla 10^{-6}, 10^{-6} ve 10^{-5} mertebesinde, yönelim hata normları ise sırasıyla 10^{-6}, 10^{-6} ve 10^{-16} mertebesindedir. Elde edilen eklem hızlarında anormallikler yoktur. Bu yörüngeler için sonuçlar oldukça tatminkardır. Bu durum belirtilen yörüngeler boyunca tekil noktalarla karşılaşılmadığını göstermektedir.

5.2.2 Sönümlendirilmiş En Küçük Kareler Yöntemi Benzetim Sonuçları

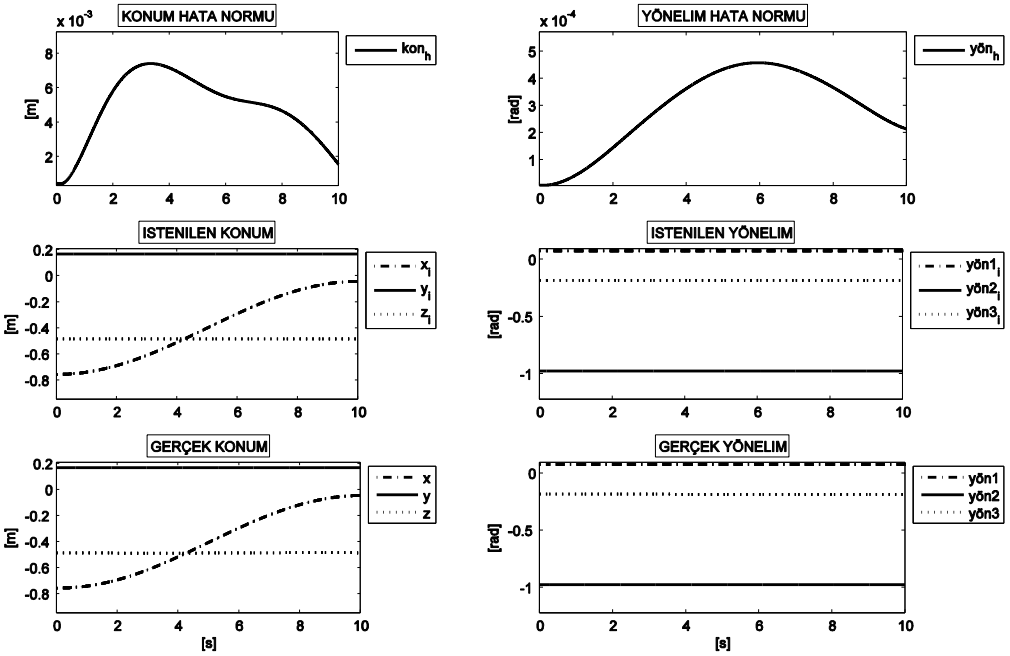
5.2.2.1 Sabit Sönümlendirme Çarpanı Kullanılan Sonuçlar

Önceki bölümdeki yörüngeler için sönümlendirilmiş en küçük kareler yöntemi kullanılarak sonuçları incelenmiştir. İlk aşamada sönümlendirme çarpanı sabit alınmıştır. Sönümlendirme çarpanı 0.1 seçildiğinde birinci yörünge boyunca robotun aldığı eklem açıları ve eklem hızları aşağıdaki Şekil 5.12' de görülebilir.



Şekil 5.12 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Eklem Açı ve Hızları

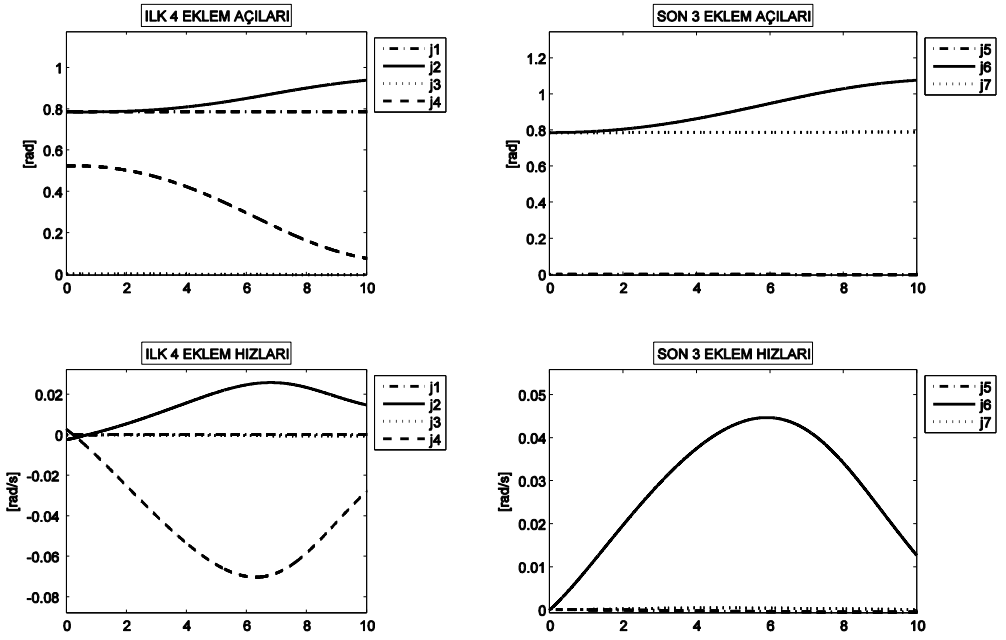
Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ve bu ikisi arasındaki fark nedeniyle oluşan konum ve yönelim hatalarının normları aşağıdaki Şekil 5.13' de görülebilir.



Şekil 5.13 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

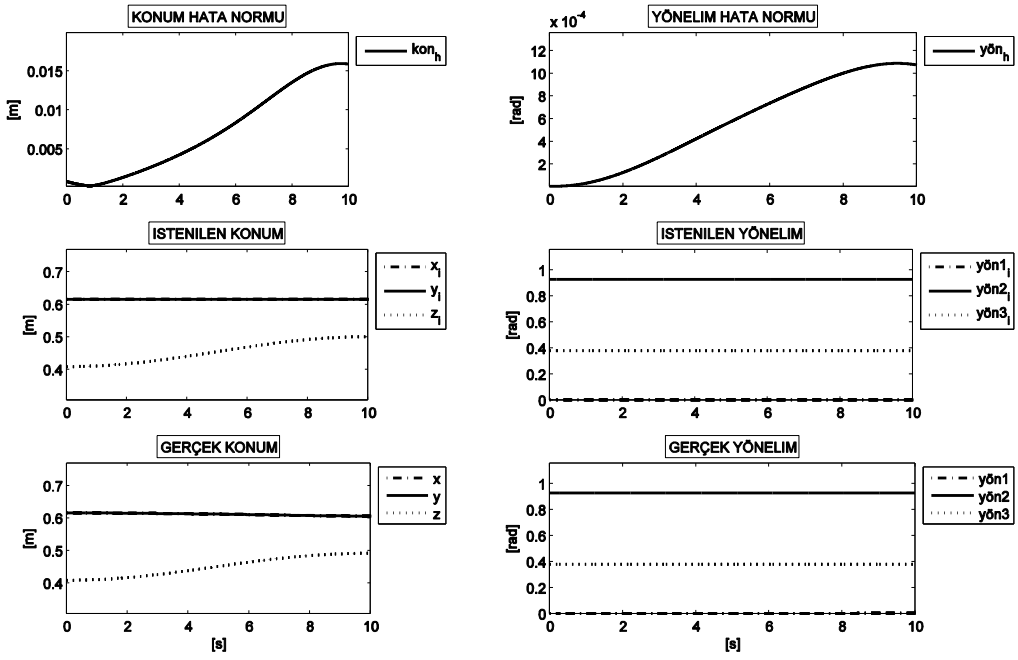
Sonuçları incelersek; konum hata normu 10^{-3} mertebesinde, yönelim hata normu ise 10^{-4} mertebesinde olmaktadır. Elde edilen sonuçlar oldukça tatminkardır. Ancak dikkati çeken nokta, önceki bölümde tekil noktaların bulunmadığı saptanan yörünge 1 için sabit çarpanlı sönümlendirilmiş en küçük kareler yönteminin, hata normlarındaki artış nedeniyle en küçük kareler yönteminden daha iyi sonuçlar vermediğidir.

Yörünge 2 için, robotun aldığı eklem açıları ve eklem hızları ise Şekil 5.14' deki gibidir.



Şekil 5.14 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Eklem Açı ve Hızları

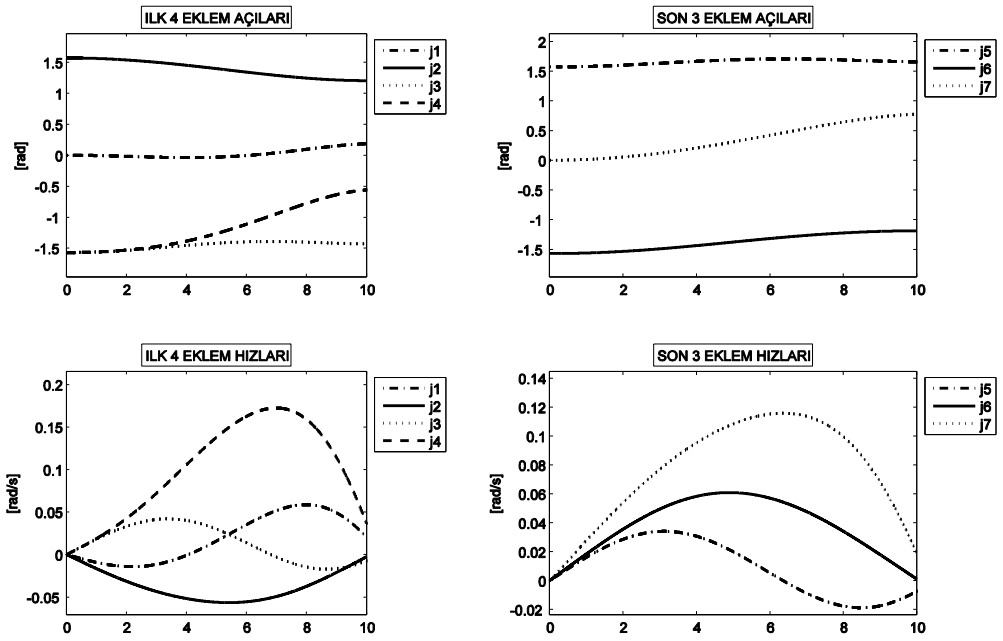
Aşağıdaki Şekil 5.15' de ise aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları görülmektedir.



Şekil 5.15 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi İle Yörünge 2 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

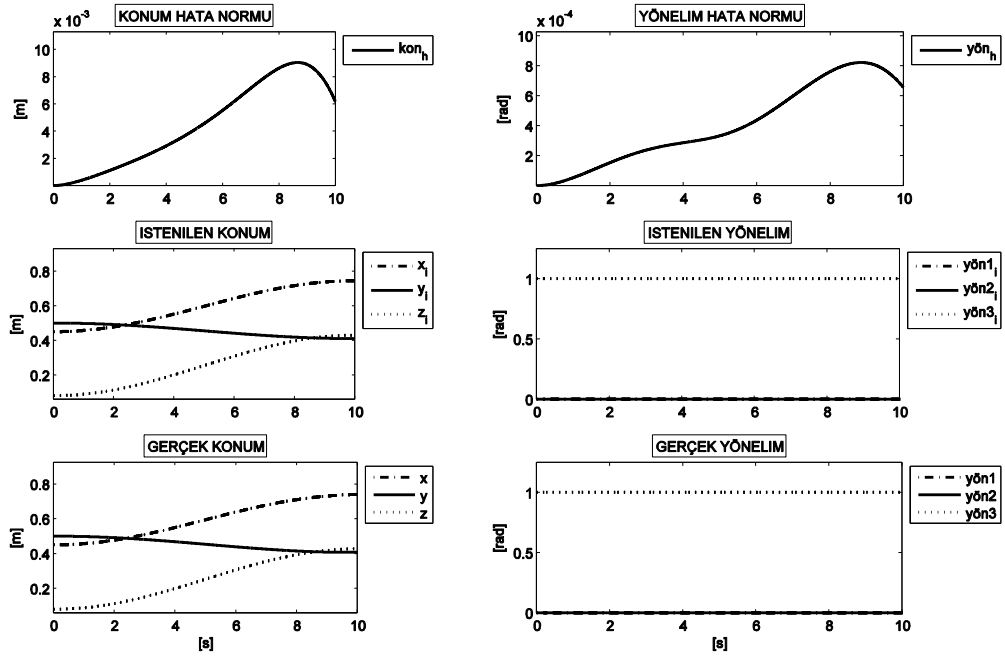
Sonuçları incelendiğinde, en küçük kareler yöntemi kullanıldığında elde edilen anormal eklem açı ve hızları ortadan kaybolmuştur. Bu durumda robot 10^{-2} hata normu mertebesiyle istenilen konumu izleyebilmektedir. Elde edilen sonuç sönümlendirilmiş en küçük kareler yönteminin tekil noktalarda daha doğru sonuçlar verdiğini göstermektedir. Ancak dikkati çeken nokta, yörünge boyunca istenen uç noktası hızının önce artıp sonra azalarak sıfır olması planlanırken, son durumda hızın sıfıra gitmediğidir.

Üçüncü yörünge için, robotun aldığı eklem açıları ve hızları Şekil 5.16' da görülebilir.



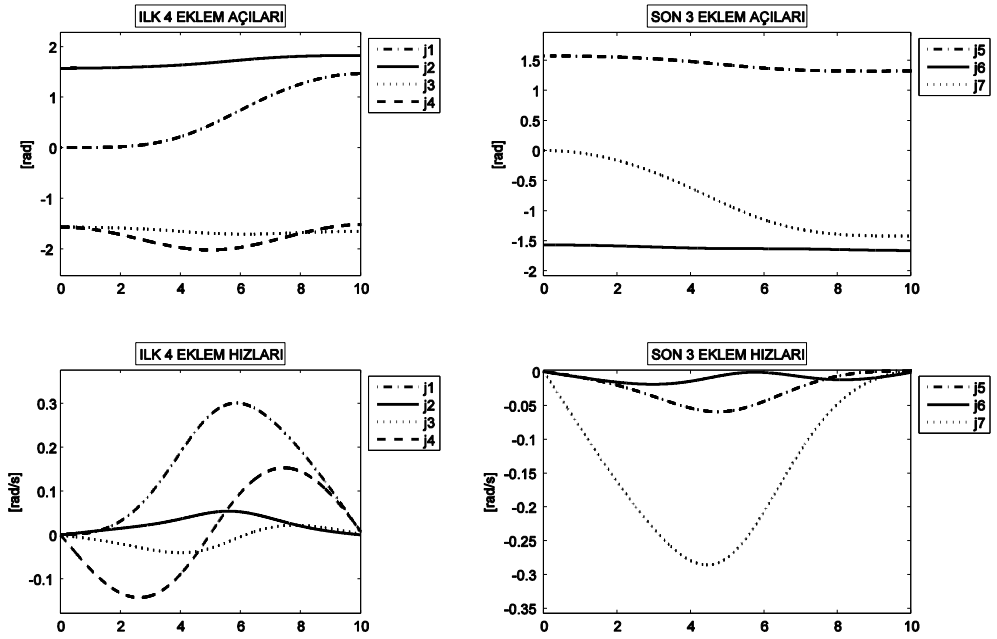
Şekil 5.16 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.17' de görülebilir.



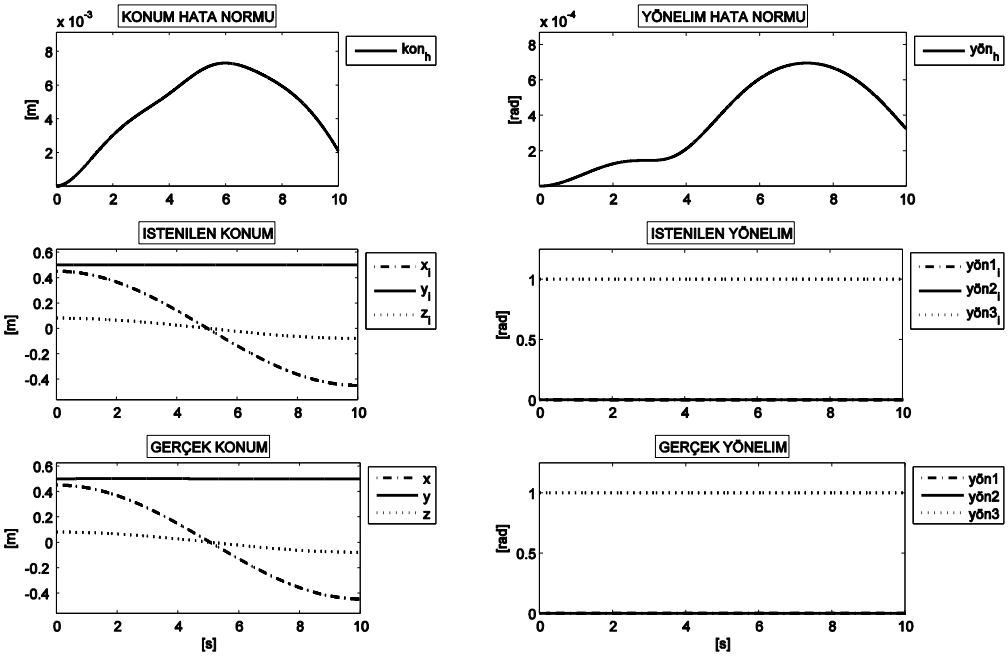
Şekil 5.17 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Dördüncü yörünge için, robotun aldığı eklem açıları ve hızları Şekil 5.18'deki gibidir.



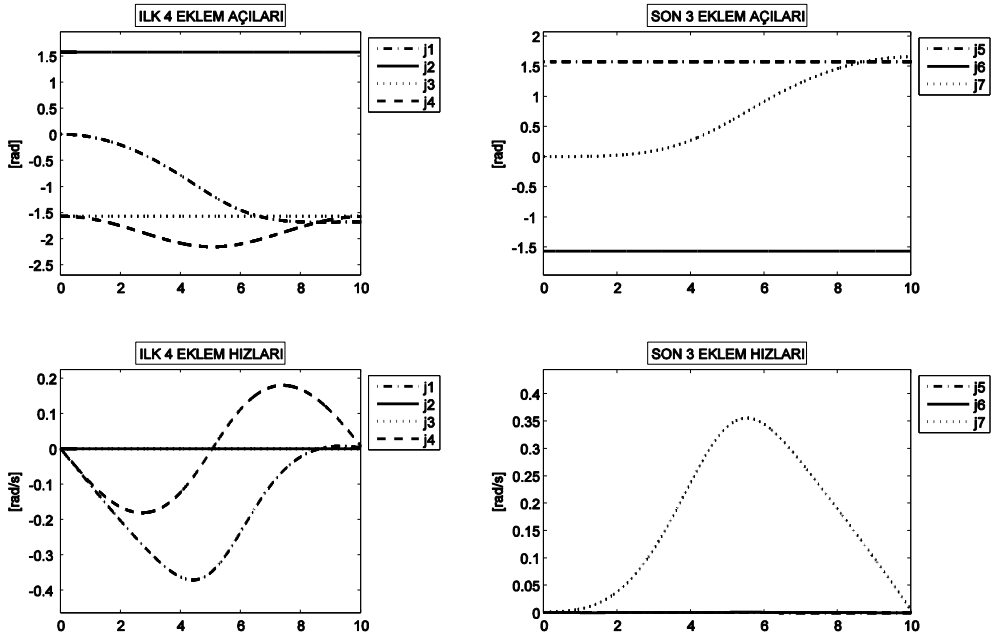
Şekil 5.18 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.19'daki gibidir.



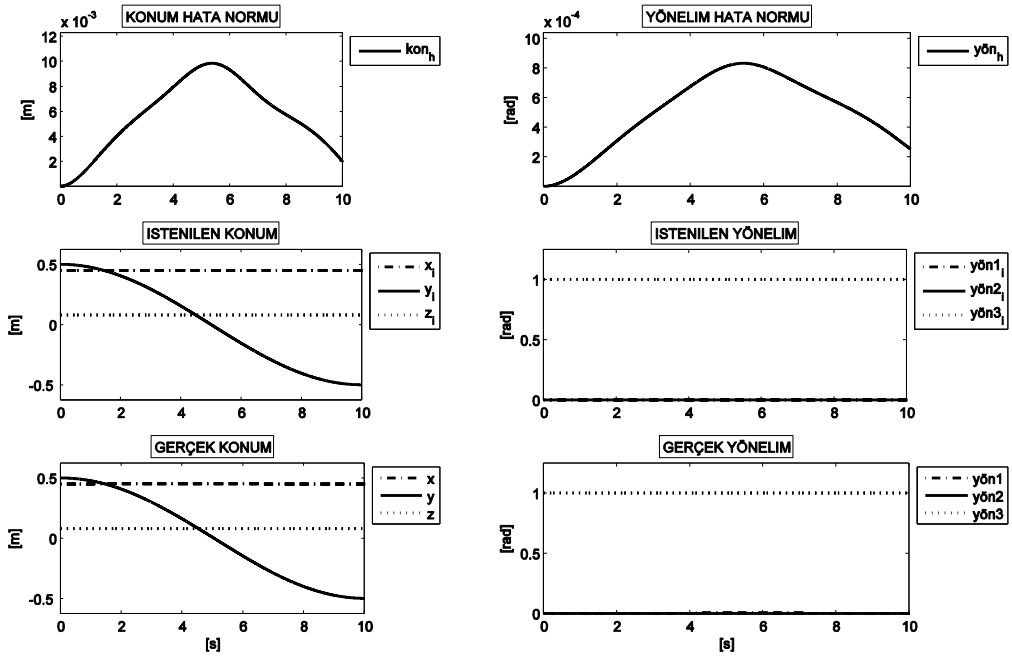
Şekil 5.19 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Beşinci ve son yörünge için, robotun aldığı eklem açıları ve hızları aşağıdaki Şekil 5.20'de görülmektedir.



Şekil 5.20 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 21' de görülebilir.

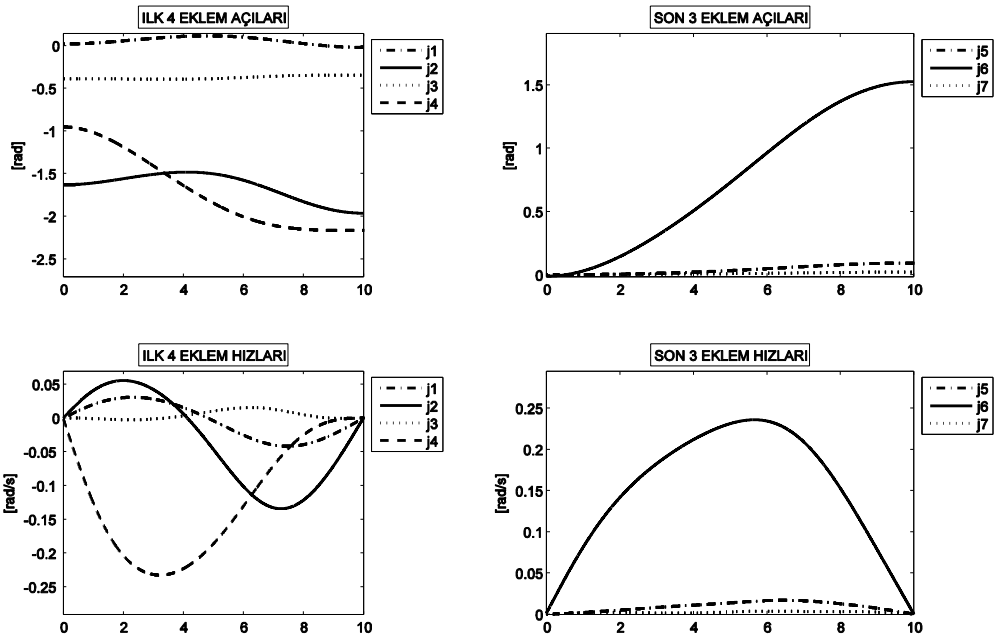


Şekil 5.21 Sabit Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Sonuçları yorumlarsak; üçüncü, dördüncü ve beşinci yörüngeler için konum hata normları sırasıyla 10^{-3}, 10^{-3} ve 10^{-3} mertebesinde, yönelim hata normları ise sırasıyla 10^{-4}, 10^{-4} ve 10^{-4} mertebesindedir. Elde edilen eklem hızlarında anormallikler yoktur. Ayrıca ikinci yörüngedeki eklem hızlarının sıfıra gitmemesi durumuyla karşılaşılmamıştır. Bu yörüngeler için sonuçlar, en küçük kareler yöntemindeki kadar olmasa da tatminkardır. Bu durum belirtilen yörüngeler boyunca tekil noktalarla karşılaşılmadığını göstermektedir.

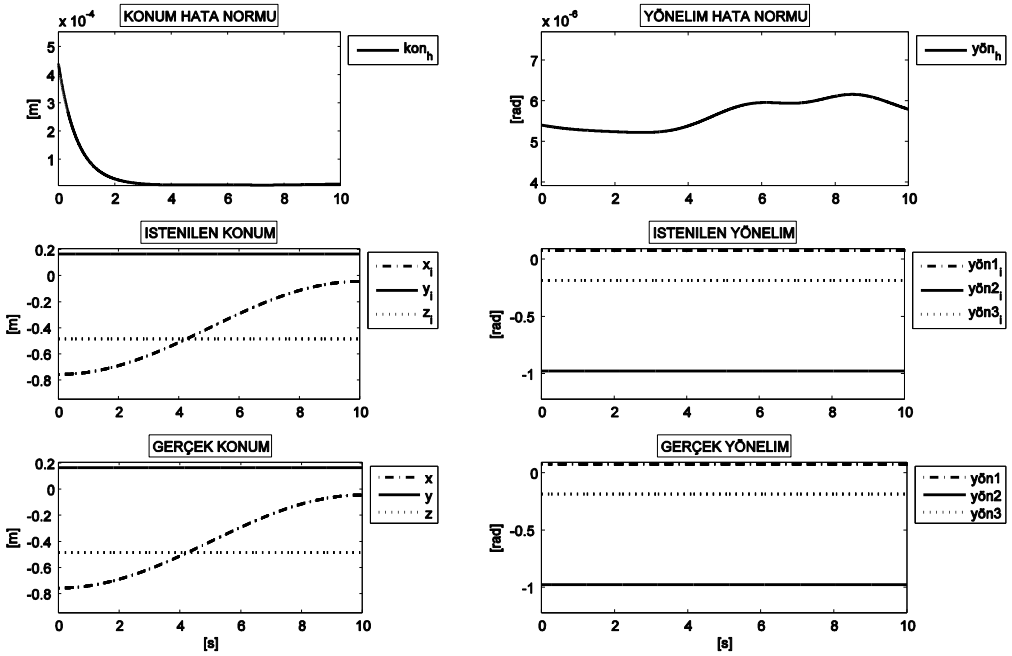
5.2.2.2 Adaptif Sönümlendirme Çarpanı Kullanılan Sonuçlar

Önceki bölümdeki yörüngeler için ikinci aşamada, sönümlendirilmiş en küçük kareler yöntemindeki sönümlendirme çarpanı adaptif olarak seçilerek, sonuçlar elde edilmiştir. Belirtilen yöntemde, birinci yörünge boyunca robotun aldığı eklem açıları ve eklem hızları aşağıdaki Şekil 5.22' de görülebilir.



Şekil 5.22 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Eklem Açı ve Hızları

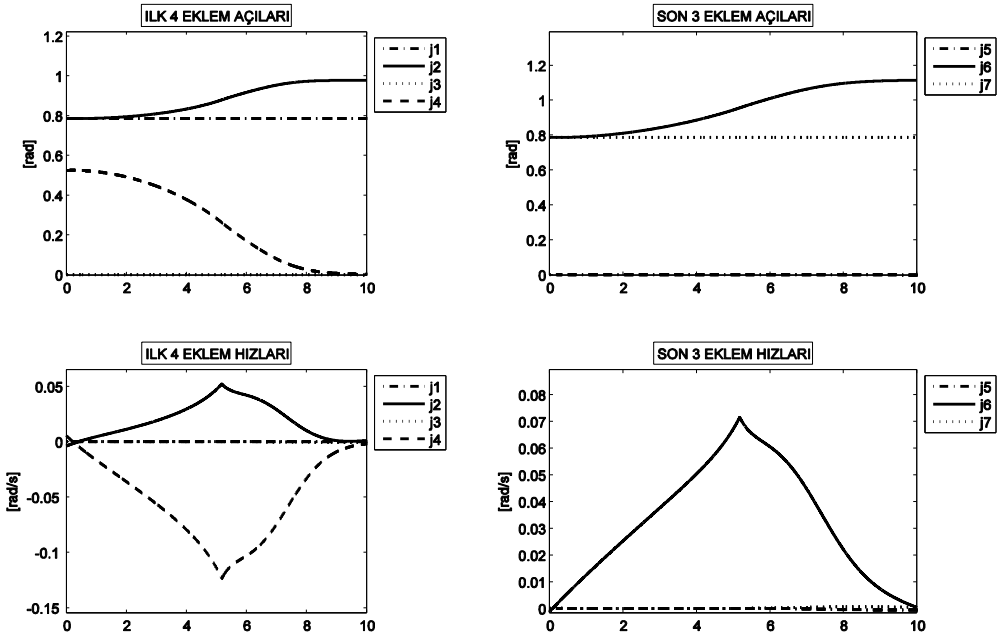
Şekil 5.23'de ise aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ve bu ikisi arasındaki fark nedeniyle oluşan konum ve yönelim hatalarının normları görülebilir.



Şekil 5.23 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

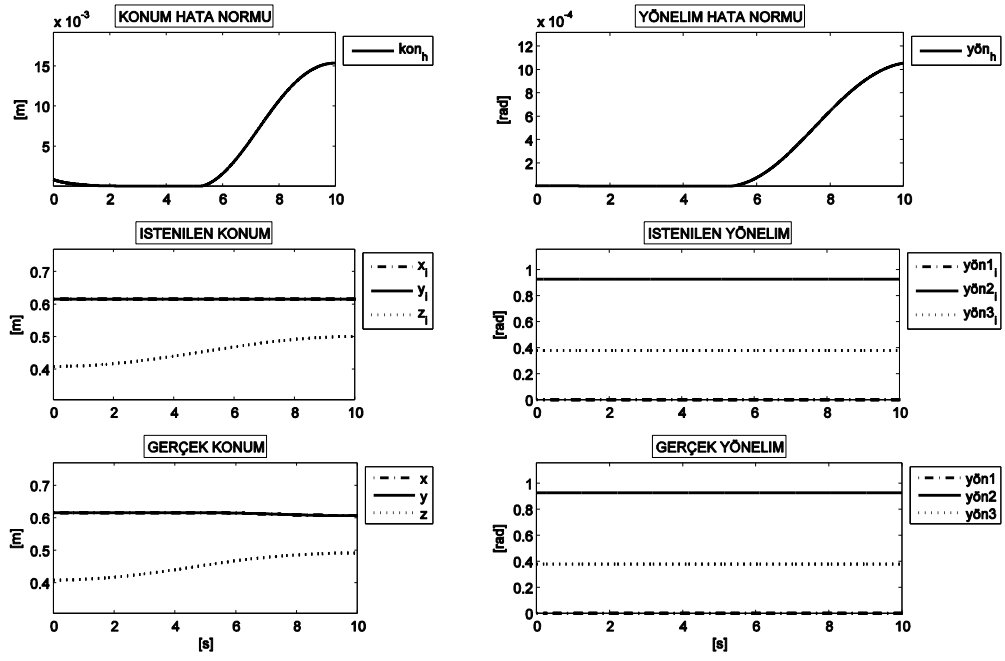
Sonuçları incelersek; konum hatası 10^{-4} mertebesinde, yönelim hatası ise 10^{-6} mertebesinde olmaktadır. Elde edilen sonuçlar oldukça tatminkardır. Sabit katsayılı sönümlendirilmiş en küçük kareler yöntemindeki hata normlarında meydana gelen artış, adaptif katsayı kullanıldığında ortadan kalkmaktadır.

İkinci yörünge için, robotun aldığı eklem açıları ve eklem hızları ise Şekil 5.24'de görülmektedir.



Şekil 5.24 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Eklem Açı ve Hızları

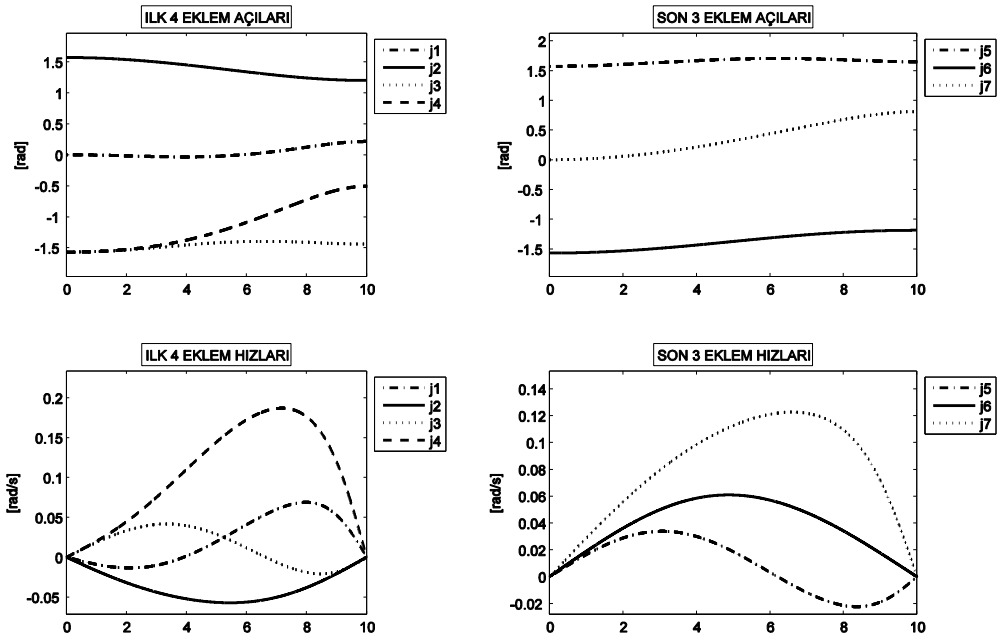
Şekil 5.25'de ise aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri, konum ve yönelim hatalarının normları görülmektedir.



Şekil 5.25 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

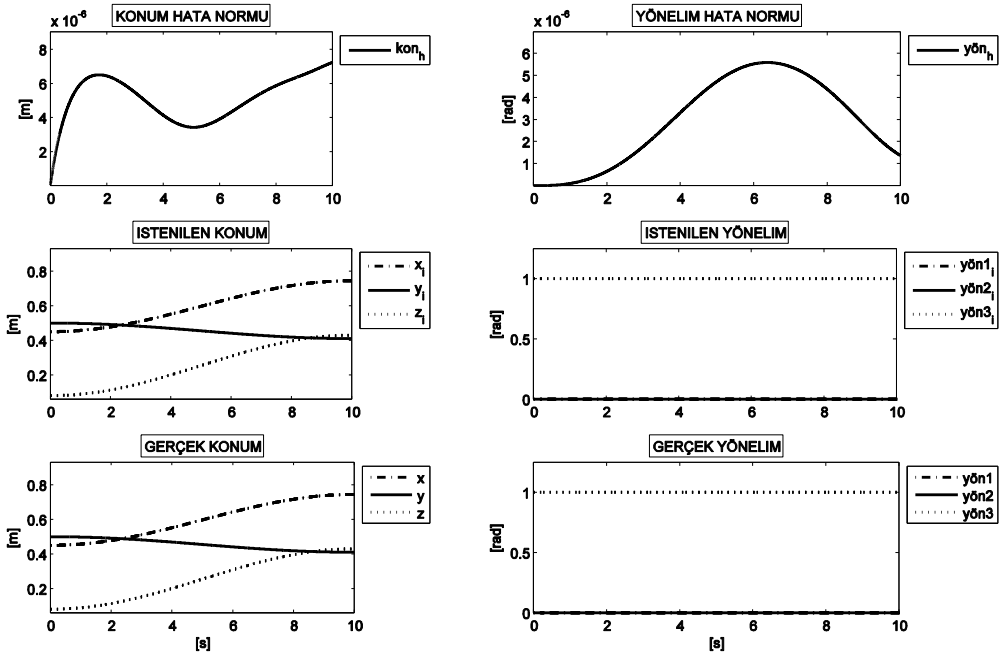
Sonuçları incelediğimizde, en küçük kareler yönteminde meydana gelen anormal hızlar ve sabit katsayılı sönümlendirilmiş en küçük kareler yöntemindeki eklem hızlarının sıfıra gitmemesi olumsuzluğunun ortadan kalktığını görebiliriz. Bu durum, daha önceki sonuçlardan tekil noktalarla karşılaşıldığı anlaşılan yörünge 2 için en iyi sonucu adaptif katsayılı sönümlendirilmiş en küçük kareler yönteminin verdiğini göstermektedir.

Üçüncü yörünge için, robotun aldığı eklem açıları ve hızları Şekil 5.26'daki gibidir.



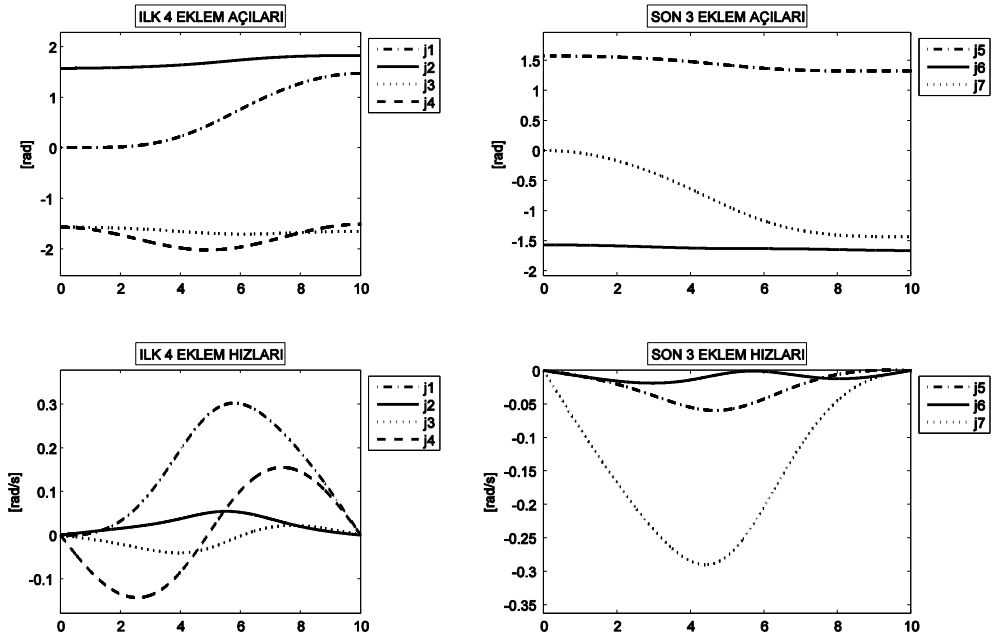
Şekil 5.26 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.27'de görülebilir.



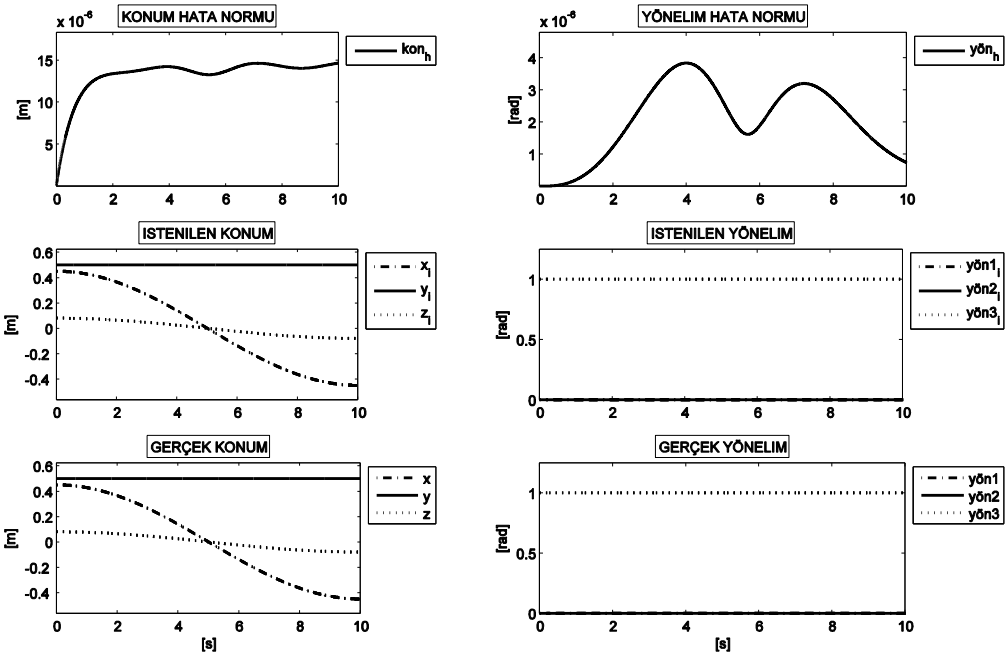
Şekil 5.27 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Dördüncü yörünge için, robotun aldığı eklem açıları ve hızları Şekil 5.28'de görülebilir.



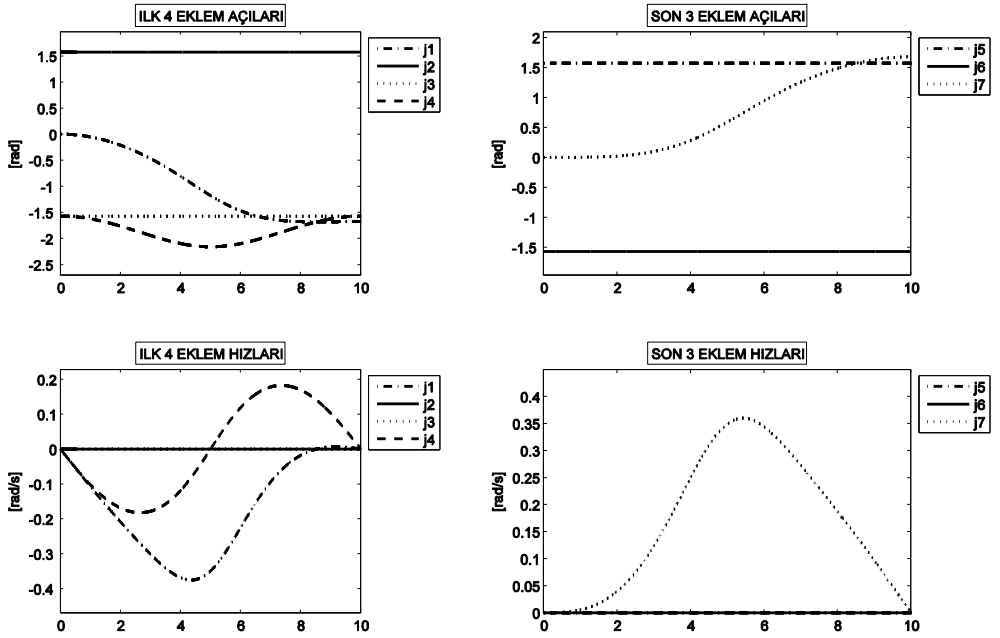
Şekil 5.28 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları ise Şekil 5.29'daki gibidir.



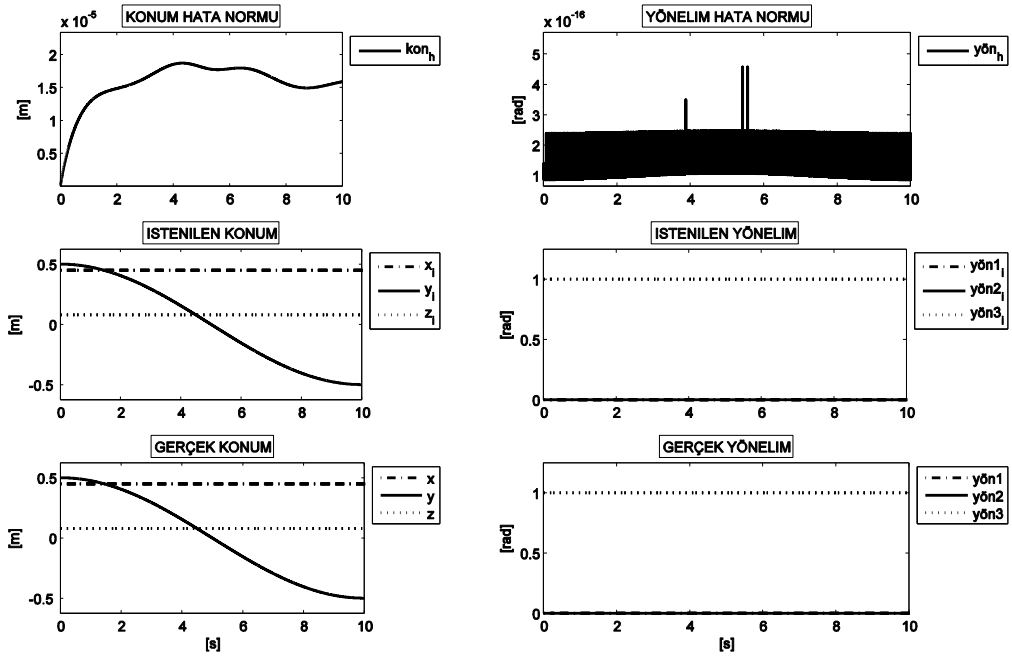
Şekil 5.29 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Beşinci ve son yörünge için, robotun aldığı eklem açıları ve hızları ise Şekil 5.30'da görülmektedir.



Şekil 5.30 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.31'de görülebilir.

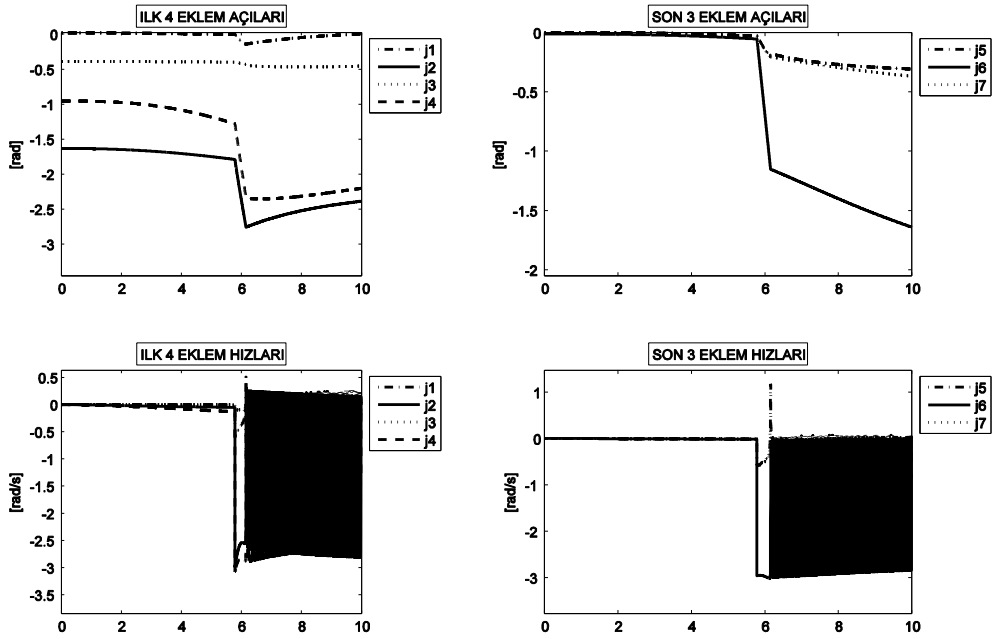


Şekil 5.31 Adaptif Çarpanlı Sönümlendirilmiş En Küçük Kareler Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Sonuçları yorumlarsak; üçüncü, dördüncü ve beşinci yörüngeler için konum hata normları sırasıyla 10^{-6}, 10^{-6} ve 10^{-5} mertebesinde, yönelim hata normları ise sırasıyla 10^{-6}, 10^{-6} ve 10^{-16} mertebesindedir. Elde edilen eklem hızlarında anormallikler yoktur. Bu yörüngeler için sonuçlar oldukça tatminkardır. Ayrıca, sabit katsayılı en küçük kareler yönteminde hata normlarında meydana gelen artış ortadan kalkmıştır. Bu durum, daha önce tekil noktalarla karşılaşılmadığı tespit edilen bu yörüngeler için adaptif çarpanlı sönümlendirilmiş en küçük kareler yönteminin, en küçük kareler yöntemini ile aynı doğrulukta sonuçlar verdiğini göstermektedir.

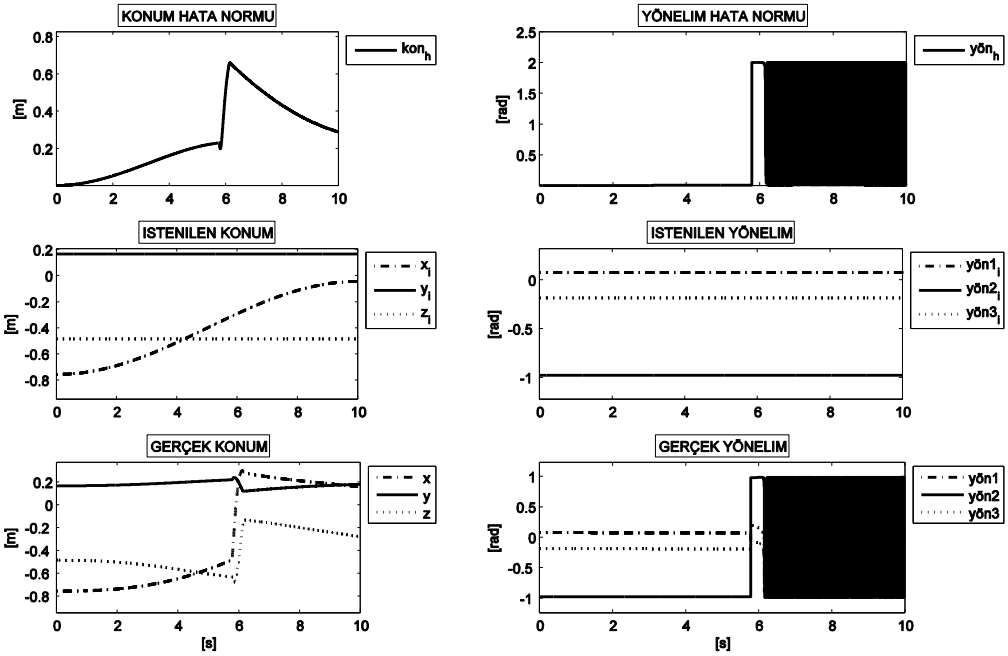
5.2.3 Jakobian Devriği Yöntemi Benzetim Sonuçları

Önceki bölümlerde kullanılan yörüngeler için bu aşamada, bölüm 3.2.2'de hesaplama kolaylılığının yanı sıra yakınsama özelliğinin zayıflığından bahsedilen jakobian devriği yöntemi kullanılarak sonuçlar elde edilmiştir. İlk yörünge için, robotun aldığı eklem açıları ve hızları aşağıdaki Şekil 5.32'de görülebilir.



Şekil 5.32 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Eklem Açı ve Hızları

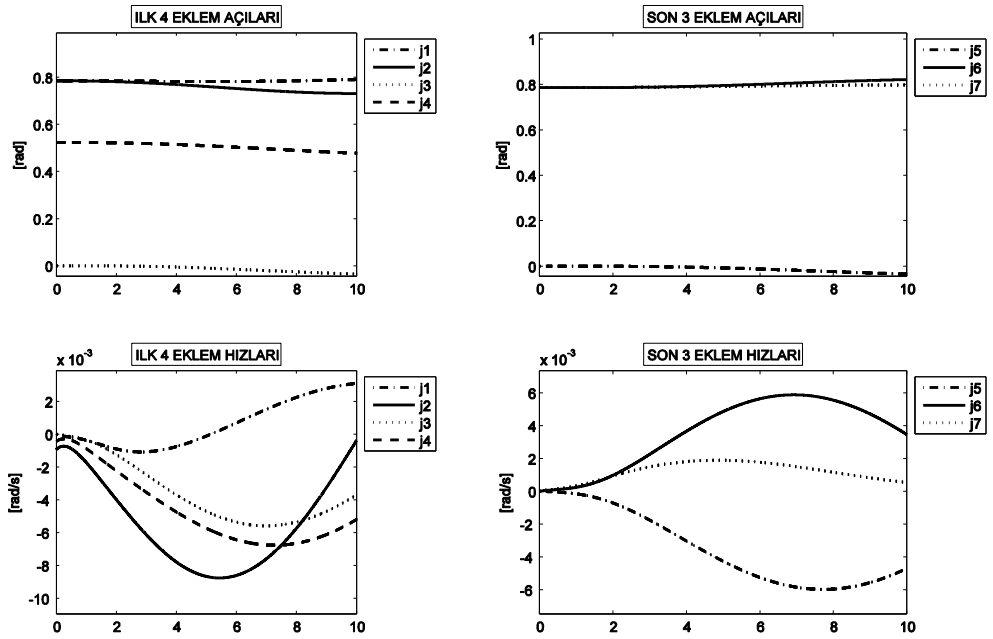
Aşağıdaki Şekil 5.33'de ise aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ve bu ikisi arasındaki fark nedeniyle oluşan konum ve yönelim hatalarının normları görülebilir.



Şekil 5.33 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 1 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

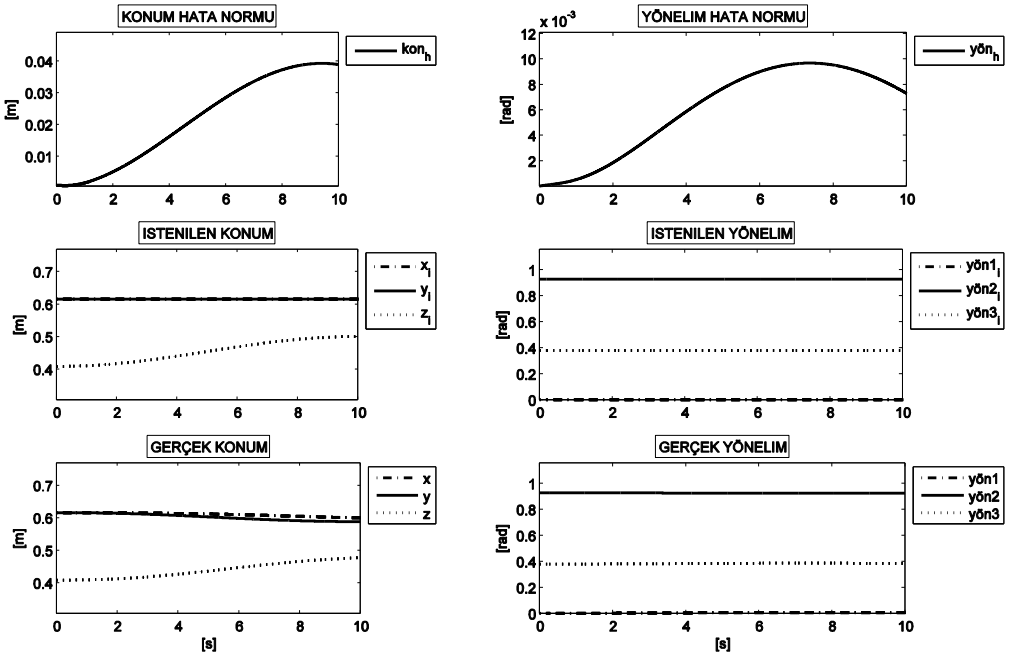
Sonuçları incelersek; konum ve yönelim hataları istenilmeyen derecede yüksek olmaktadır. Elde edilen sonuçlar baştan da beklenildiği gibi tatminkar değildir.

İkinci yörünge için, robotun aldığı eklem açıları ve hızları aşağıdaki Şekil 5.34'de görülmektedir.



Şekil 5.34 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Eklem Açı ve Hızları

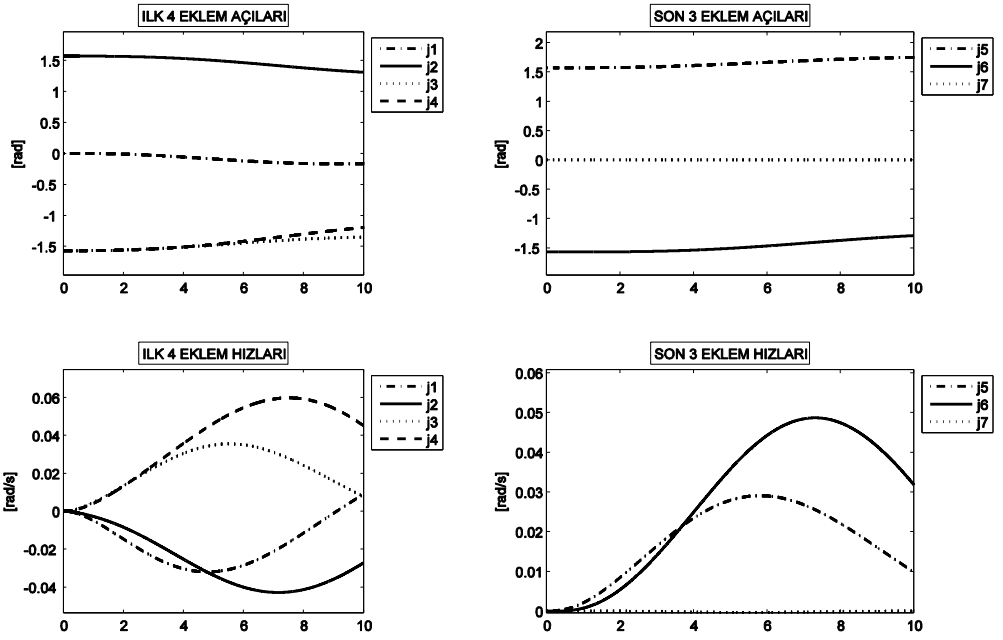
Aşağıdaki Şekil 5.35'de ise aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları görülmektedir.



Şekil 5.35 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 2 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

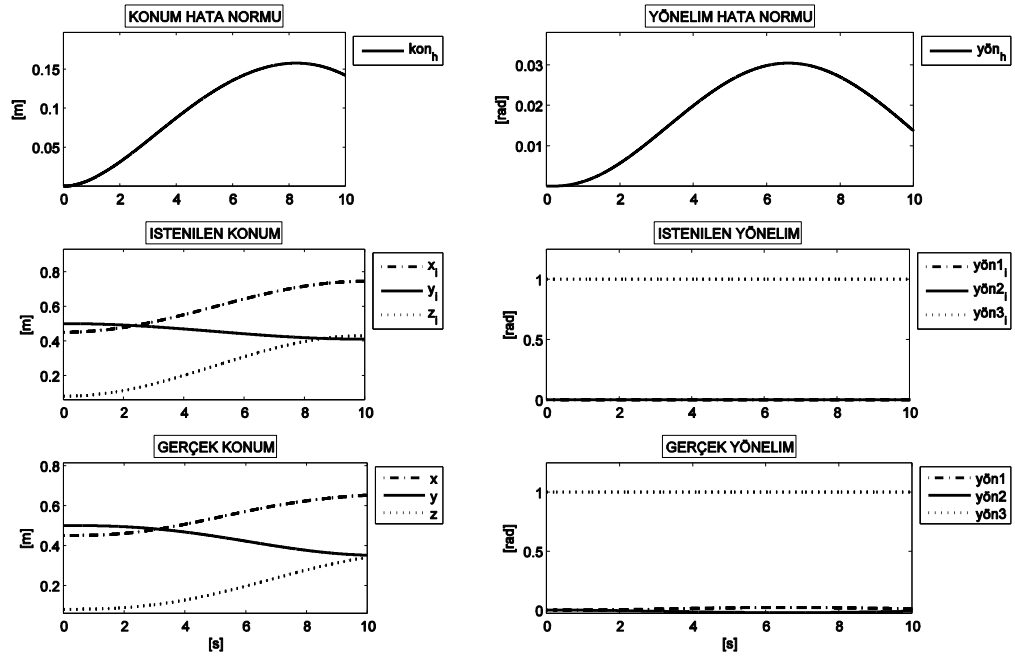
Sonuçları incelediğimizde, istediğimiz yörünge tam olarak takip edilemese de en küçük kareler yönteminde meydana gelen anormal hızlar görülmemektedir. Jakobian devriği alma işleminde tekil noktaların sorun oluşturmadığı düşünüldüğünde, daha önceki sonuçlardan tekil noktalarla karşılaşıldığı anlaşılan ikinci yörünge için kullanılan jakobian devriği yönteminin en küçük kareler yönteminden daha iyi çalışması doğal karşılanmalıdır.

Üçüncü yörünge için, robotun aldığı eklem açıları ve hızları Şekil 5.36'da görülebilir.



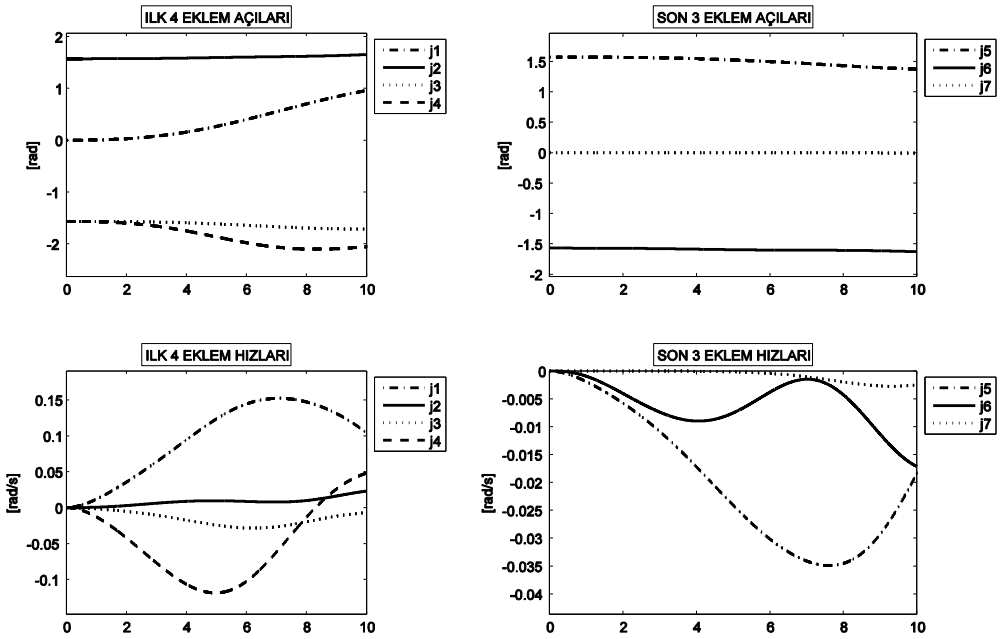
Şekil 5.36 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları ise Şekil 5.37' deki gibidir



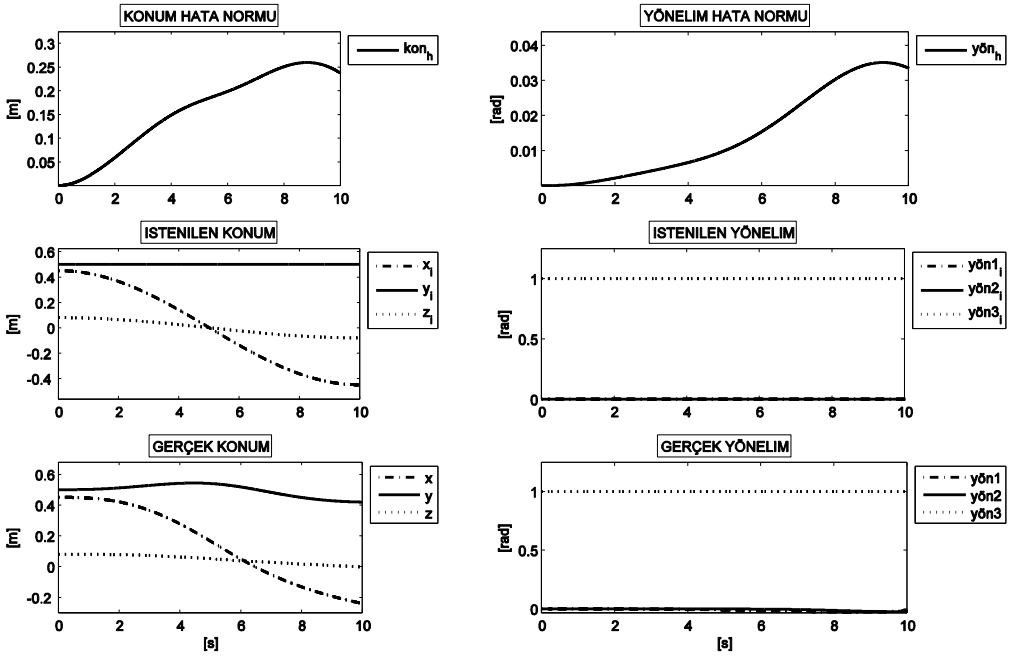
Şekil 5.37 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 3 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Dördüncü yörünge için, robotun aldığı eklem açıları ve hızları Şekil 5.38'de görülebilir.



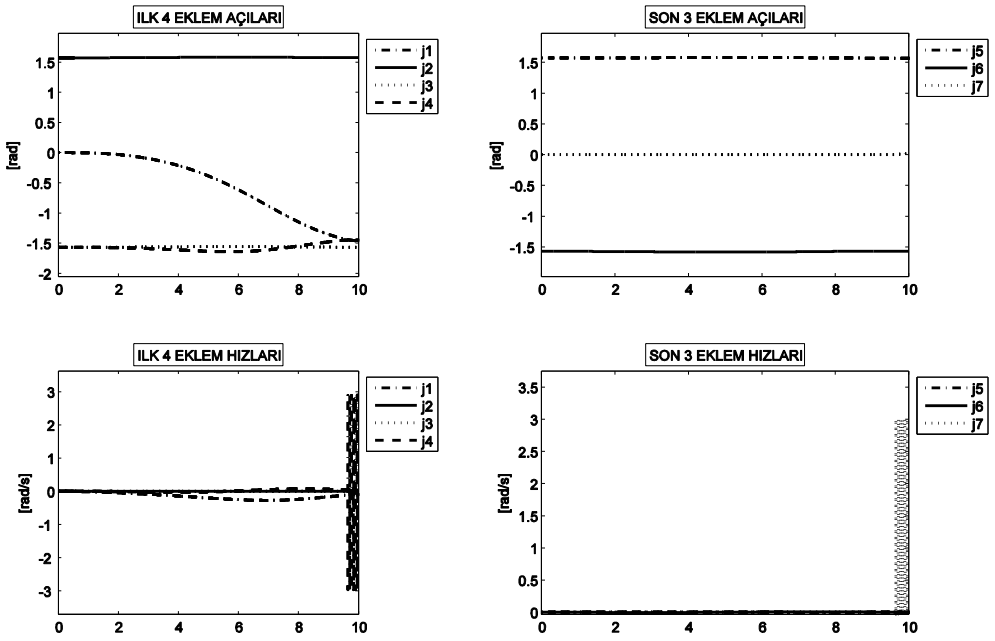
Şekil 5.38 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.39'de görülebilir.



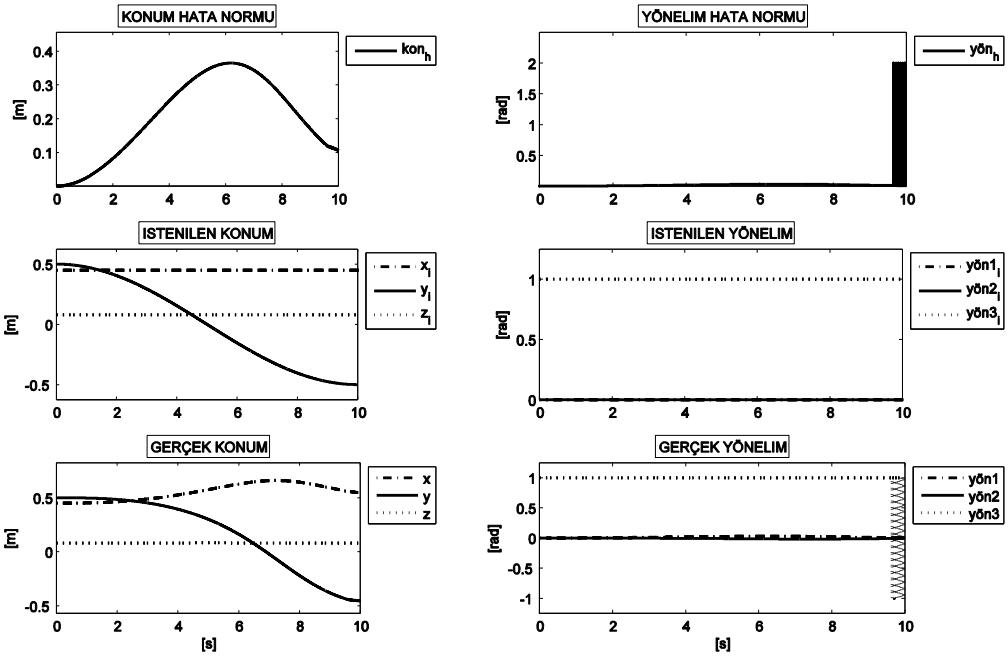
Şekil 5.39 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 4 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Beşinci ve son yörünge için, robotun aldığı eklem açıları ve hızları ise Şekil 5.40'da görülmektedir.



Şekil 5.40 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.41'deki gibidir.



Şekil 5.41 Jakobian Devriği Yöntemi Kullanıldığında Yörünge 5 İçin Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Sonuçları incelersek; robotun uç noktası üçüncü, dördüncü ve beşinci yörüngeleri tam olarak takip edememektedir. Konum hata normları 10^{-1} mertebesinde, yönelim hata normları ise sırasıyla 10^{-2}, 10^{-2} ve 10^{1} mertebesindedir. Hesaplama kolaylığı dikkate alındığında özellikle üçüncü ve dördüncü yörüngeler için, hassas sonuçların gerekmediği durumlarda bu çözüm yöntemi dikkate alınabilir. Ancak beşinci yörünge için eklem hızlarında artışlar söz konusu olduğundan bu yöntem dikkate alınmamalıdır.

5.2.4 Sınırlamasız Çözüm Karşılaştırmaları

Elde ettiğimiz sonuçları karşılaştıracak olursak; en küçük kareler yönteminde yörünge boyunca, tekil noktalarla karşılaşılmadığı ya da tekil noktaların komşuluğuna girilmediği sürece robot istenilen yörüngeyi, istenilen doğrulukta takip etmektedir. Ancak tekil noktalarla karşılaşıldığında, σ_i sıfıra giderken kullanılan tekil değer ayrıştırmasındaki $1/\sigma_i$ terimi sonsuza yaklaşmaktadır. Bu durum robot eklem hızlarının çok fazla olmasına sebep olmaktadır. Bu nedenle, en küçük kareler

yöntemi tekil noktalarla karşılaşılmayan yörüngelerde sorun çıkarmamasına rağmen, tekil noktalarla karşılaşılan yörüngelerde yüksek eklem hızlarına neden olmaktadır. Bu çalışma sırasında yapılan benzetim sonuçlarına göre de yörünge 2 için yüksek eklem hızları meydana gelmekte, diğer yörüngeler için herhangi bir olumsuzlukla karşılaşılmamaktadır. Bu durum, yapılan benzetimin teoriyle uyuştuğunu gösterir.

Yüksek eklem hızlarını engellemek için sabit çarpanlı sönümlendirilmiş en küçük kareler yöntemi kullanıldığında, ikinci yörüngedeki yüksek eklem hızları azalmasına rağmen başka bir takım olumsuzluklar ortaya çıkmaktadır. Tekil noktalarla karşılaşılmayan yörüngelerde, oluşan konum ve yönelim hatalarının normları en küçük kareler yöntemine göre artış göstermektedir. Ayrıca, istenilen hareket tamamlandığında eklem hızları sıfıra gitmemektedir. Bunun nedeni, tekil noktalardan uzaktayken bile jakobian matrisinin tersi alınırken kullanılırken $\frac{\sigma_i}{\sigma_i^2 + \lambda^2}$ terimindeki, λ teriminin çözüme etki etmesidir. λ terimi, σ_i sıfıra giderken eklem hızlarının sonsuza gitmemesi için gerekli bir terimken, σ_i sıfırdan yeterince büyük olduğunda, sonucu değiştirdiği için gereksiz bir terim haline gelmektedir.

Oluşan bu olumsuzlukları aşmak için adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi kullanılmıştır. Kullanılan bu yöntemle, tekil noktalarla karşılaşılan yörünge 2 için sabit çarpanlı sönümlendirilmiş en küçük kareler yönteminden daha düşük hata normları ile sonuç elde edilebilirken, hareket sonunda eklem hızlarının sıfıra gitmesi sağlanabilmektedir. Ayrıca bu yöntemle, tekil noktalarla karşılaşılmayan diğer yörüngelerde de sabit çarpanlı sönümlendirilmiş en küçük kareler yönteminin neden olduğu hata normu artışı ortadan kaldırılmakta, en küçük kareler yöntemi ile aynı hata başarımlarına ulaşılarak, eklem hızlarının sıfıra gitmesi sağlanabilmektedir. Bunun nedeni, jakobian matrisinin tersi alınırken kullanılırken $\frac{\sigma_i}{\sigma_i^2 + \lambda^2}$ terimindeki, λ teriminin σ_i sıfıra yaklaştığında sonuca etki ederken σ_i'nin sıfırdan uzak olduğu durumlarda sıfır olup sonuca etki etmemesidir.

Son olarak benzetimi yapılan jakobian devriği yönteminin, işlem kolaylığı sağlamasına rağmen istenilen yörünge takibini yapamadığı ya da kısmen yapabildiği ve hata normlarının teoriden de beklendiği üzere yüksek olduğu gözlenmiştir. Dikkati çeken nokta, matris tersi alma işleminde sorun oluşturan tekil değerlerin,

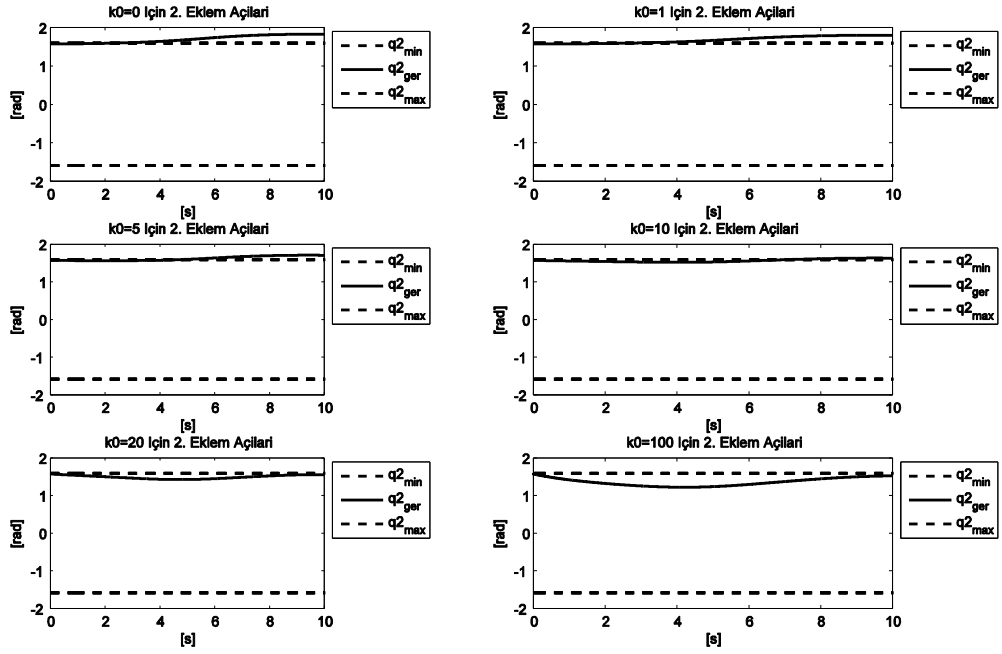
matris devriği alma işleminde sorun teşkil etmemesi nedeniyle tekil noktalarla karşılaşılan yörünge 2 için en küçük kareler yönteminden daha iyi sonuç vermesidir.

Yapılan benzetimlerle, adaptif çarpanlı sönümlendirilmiş en küçük kareler yönteminin en iyi sonuçları veren yöntem olduğu görülmüştür. Tekil noktalarla karşılaşılıp karşılaşılmadığına, istenilen hareketin hassasiyetine veya çözüm kolaylığına göre diğer yöntemlerin de kullanılabileceği gözlenmiştir.

5.3 Mitsubishi PA 10 Robot Kolu İçin Sınırlamalı Ters Kinematik Çözümler ve Karşılaştırmaları

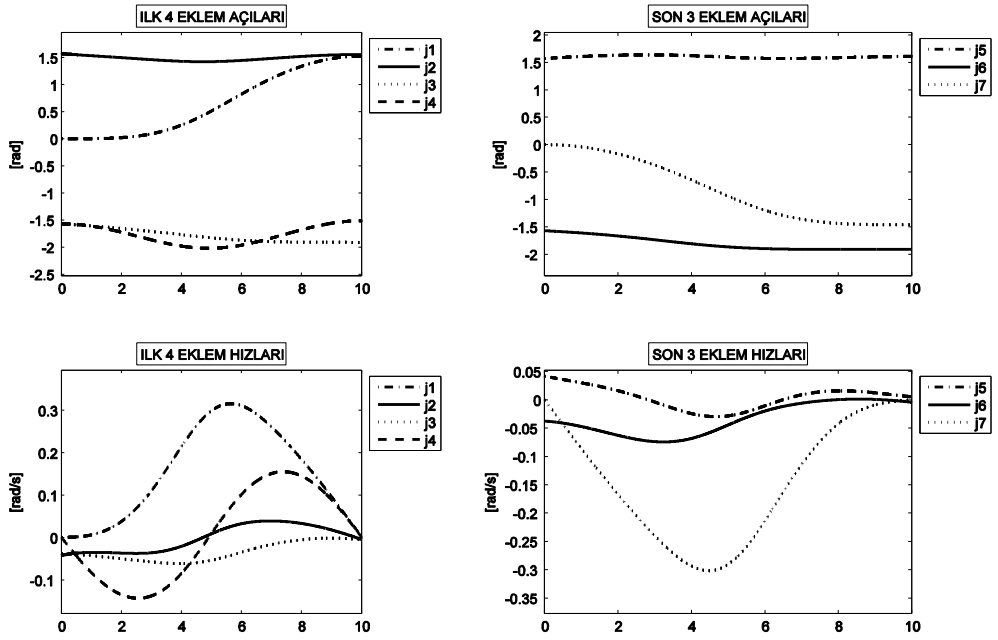
5.3.1 Mekanik Eklem Sınırlarından Sakınma Uygulanan Benzetim Sonuçları

Eklemlerin mekanik sınırlarından sakınmak için önceki bölümde en iyi sonuç veren yöntem olduğu anlatılan adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi kullanılmıştır. Sakınma ölçütü ise (4.22) eşitliğindeki gibi alınmıştır. Yörünge 4 için herhangi bir sakınma yokken, ikinci eklem açısı maksimum mekanik sınırının dışına çıkmaktadır. Diğer tüm eklemler ise mekanik sınırları içinde kalmaktadır. İkinci eklemi mekanik sınırları içerisinde çalıştırmak için sakınma ölçütü bazı katsayılarla ağırlıklandırılmıştır. k_0 ağırlık katsayısını göstermek üzere aşağıdaki Şekil 5.42'de k_0' ın değişik değerleri için ikinci eklemin aldığı açılar görülebilir.



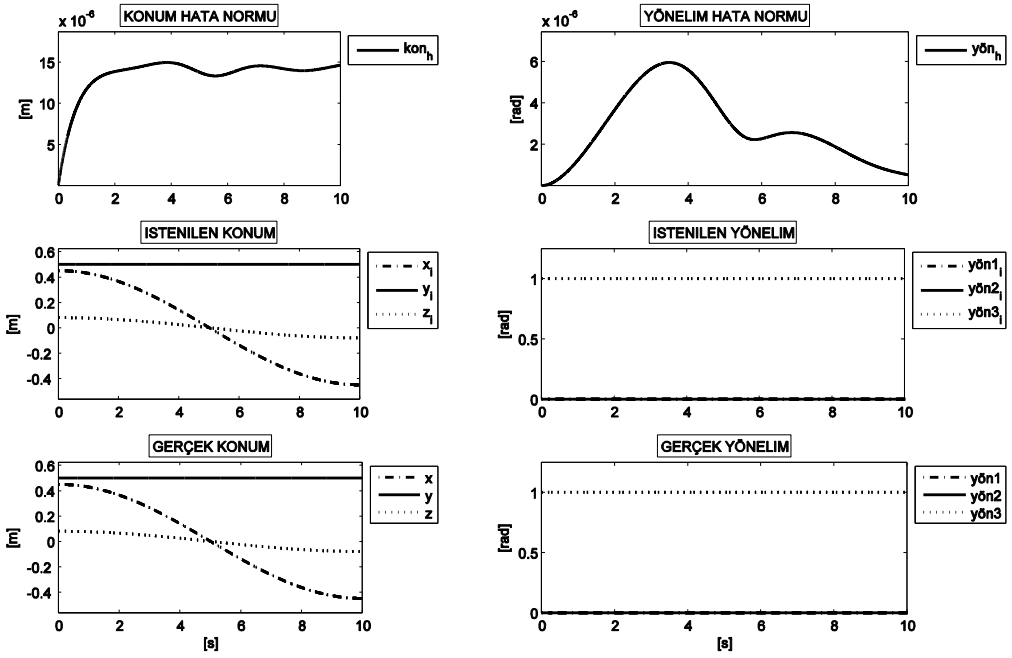
Şekil 5.42 Farklı Ağırlıklandırma Katsayıları İçin Yörünge 4' deki İkinci Eklem Açılarının Zamanla Değişimi

Şekil 5.42'de görüldüğü gibi k_0 ağırlık katsayısının sıfır olduğu (yani herhangi bir sakınma uygulanmadığı durumda) istenilen yörünge takip edilirken ikinci eklem mekanik sınırları dışına çıkmaktadır. Bu sonuçları gerçek zamanda robota uygulamak robotun zarar görmesine yol açar. İkinci eklemi mekanik sınırlarına çekmek için k_0 ağırlık katsayısı ilk aşamada bir yapılmış, ikinci eklemin maksimum değeri azalmasına rağmen sınırları içine çekilemediği görülmüştür. k_0 katsayısı giderek arttırılmış ve değeri 20 olduğunda ikinci eklem mekanik sınırları içine çekilmiştir. k_0 katsayısının daha da arttırılması durumunda ikinci eklemin maksimum değerinin azalmaya devam ettiği yine Şekil 5.42'de görülmektedir. Aşağıdaki Şekil 5.43'de ikinci eklemin mekanik sınırları içine çekildiği ilk ağırlık katsayısı olan 20 değeri alınarak bu durumuna ilişkin yörünge boyunca eklemlerin aldığı açı ve hızlar görülmektedir.



Şekil 5.43 Yörünge 4 İçin Mekanik Eklem Sınırlarından Sakınıldığında Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları ise Şekil 5.44'de görülmektedir.

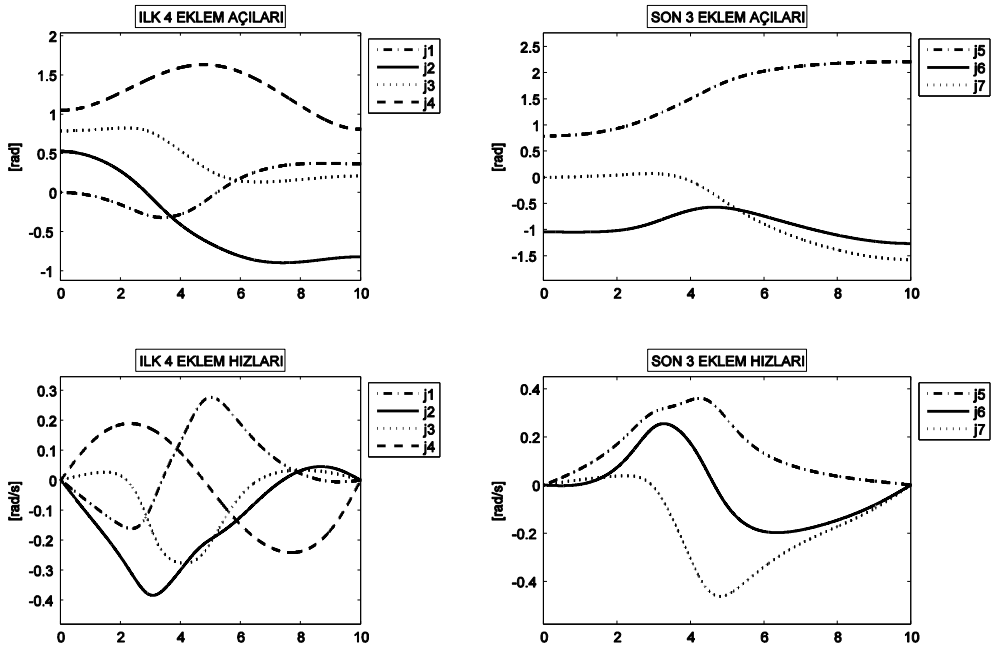


Şekil 5.44 Yörünge 4 İçin Mekanik Eklem Sınırlarından Sakınıldığında Robotun Uç Noktasının İstenilen ve Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Elde edilen bu sonuçları, sakınma uygulanmayan haldeki Şekil 5.28 ve Şekil 5.29'daki sonuçlarla karşılaştıracak olursak, robot kolunun eklem konfigürasyonunda küçük değişikler yaparak istenilen yörüngeyi aynı doğrulukla takip edebildiği görülebilir. Bu durum artıklık kullanımının başarıyla yapıldığının bir göstergesidir.

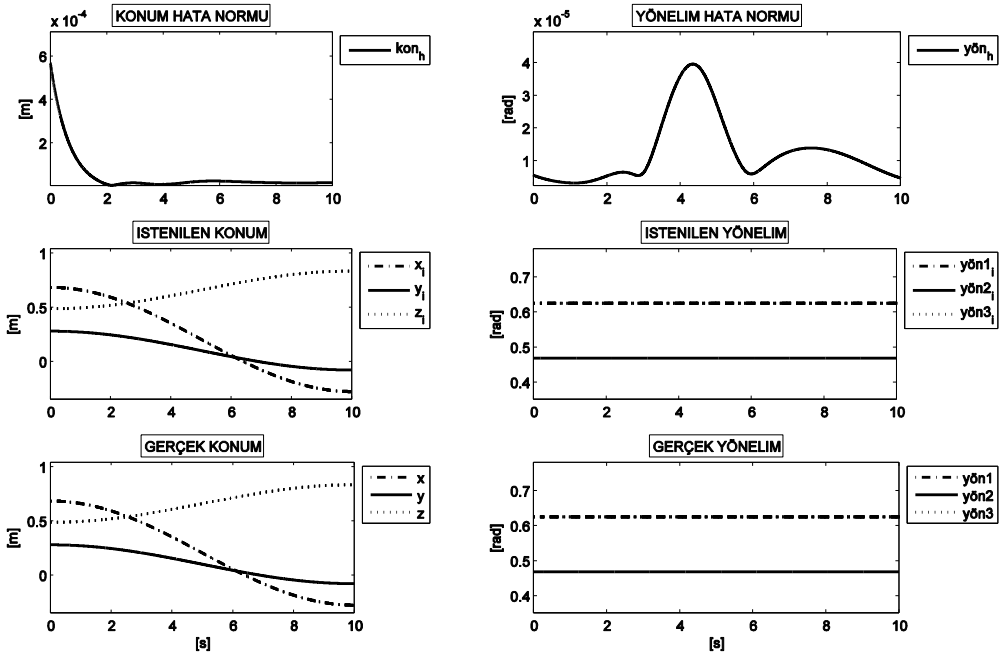
5.3.2 Engellerden Sakınma Uygulanan Benzetim Sonuçları

Engellerden sakınmak için yine adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi kullanılmıştır. Sakınma uygulayabilmek için uzayda konumu sabit bir engel ve sakınma uygulanmayan durumda linklerden birinin engele çarptığı bir yörünge (yörünge 6) belirlenmiştir. Sakınmada kullanılan engel modeli olarak küre, robot linki modeli olarak ise doğru parçası seçilmiştir. Bölüm 4.4'de anlatıldığı gibi benzetim boyunca her bir an için robotun engele olan uzaklığı hesaplanarak o andaki kritik link belirlenmiş ve belirlenen kritik linke engelden ters yöne bir hız atanmıştır. Yörünge 6 için sakınma uygulanmayan halde eklemlerin aldığı açı ve hız değerleri Şekil 5.45'de görülen şekildedir.



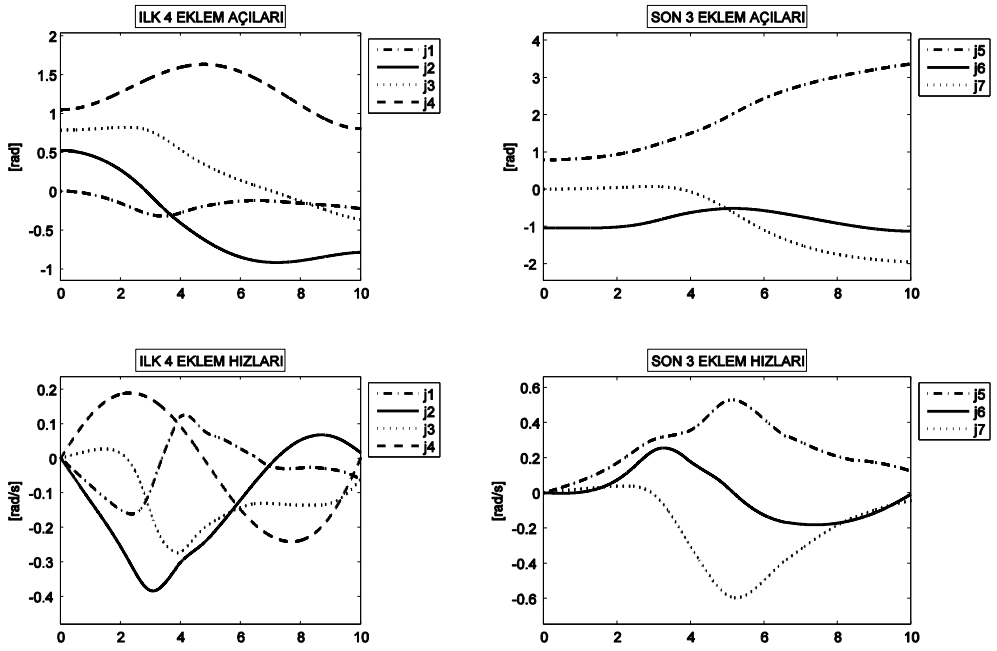
Şekil 5.45 Yörünge 6 İçin Engellerden Sakınma Uygulanmayan Halde Robotun Eklem Açı ve Hızları

Aynı yörünge ve yöntem için robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.46'da görülebilir.



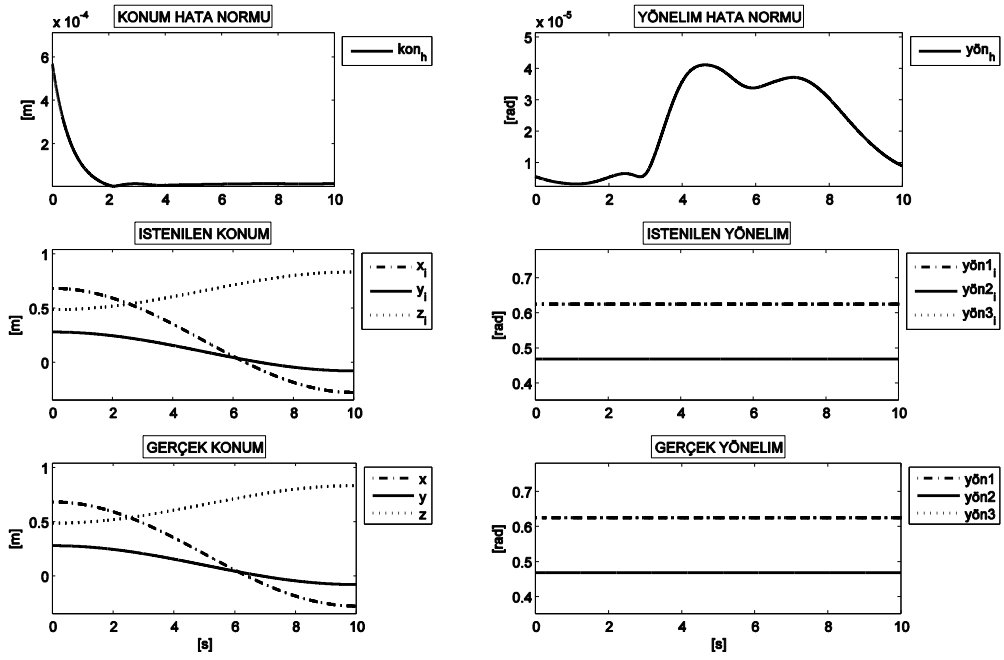
Şekil 5.46 Yörünge 6 İçin Engellerden Sakınma Uygulanmayan Halde Robotun Uç Noktasının İstenilen, Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Şekil 5.45 ve Şekil 5.46'da görülebildiği gibi robot istenilen yörüngeyi başarılı bir şekilde takip edebilmektedir. Ancak, elde edilen eklem konfigürasyonlarının "MATLAB Robotic Toolbox" kullanılarak benzetimi yapıldığında robot kolunun engele çarptığı görülmüştür. Engelden sakınmak için bölüm dörtte anlatılan tam çözüm ve yaklaşık çözüm modelleri kullanılarak sistem benzetimleri yapılmıştır. Yörünge altı için, engelden sakınmak amacıyla yaklaşık çözüm modeli kullanıldığında eklemlerin aldığı açı ve hız değerleri Şekil 5.47'de görülmektedir.



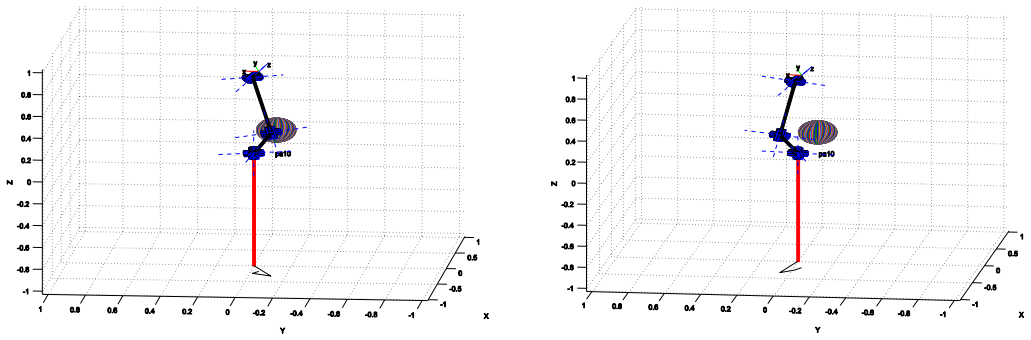
Şekil 5.47 Yörünge 6 İçin Yaklaşık Çözüm Modeli Kullanılarak Engellerden Sakınma Uygulandığında Robotun Eklem Açı ve Hızları

Aynı yörünge ve model için, robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.48'de görüldüğü şekilde elde edilmiştir.



Şekil 5.48 Yörünge 6 İçin Yaklaşık Çözüm Modeli Kullanılarak Engellerden Sakınma Uygulandığında Robotun Uç Noktasının İstenilen, Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

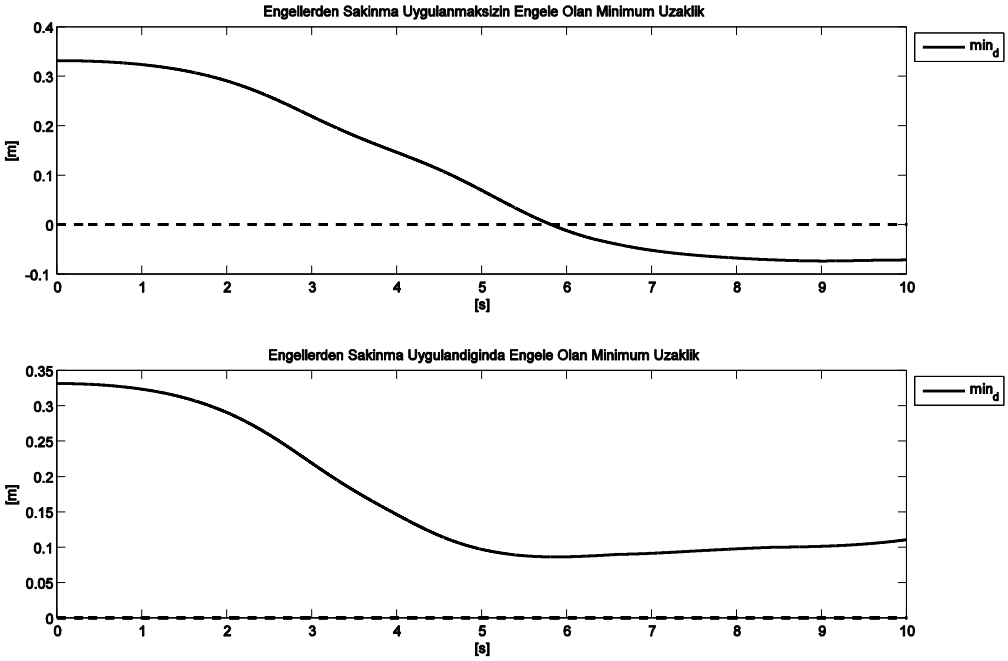
Elde edilen eklem konfigürasyonlarının “MATLAB Robotic Toolbox” kullanılarak benzetimi yapıldığında hareket süresince robot kolunun engele çarpmadığı görülmüştür. Hareketin herhangi bir anına ilişkin eklem konfigürasyonları aşağıda Şekil 5. 49’da görülebilir.



a)Sakınmasız Çözüm b)Sakınmalı Çözüm

Şekil 5.49 Sakınmasız ve Sakınmalı Çözümlerin Herhangi Bir Anına İlişkin Eklem Konfigürasyonları

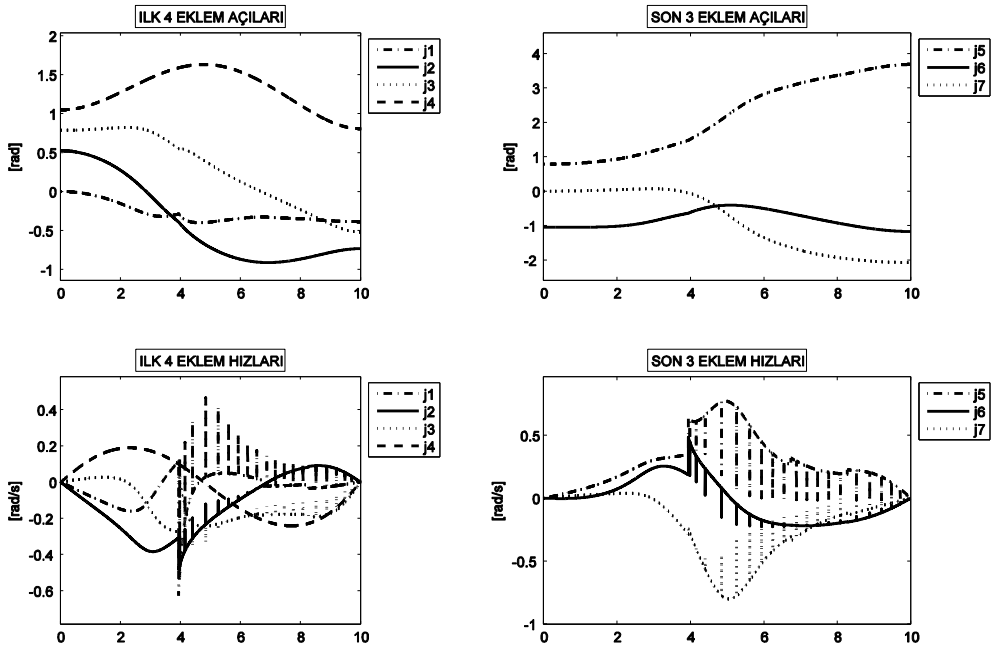
Bu durum, Şekil 5.45 ve Şekil 5.47 arasındaki farklılığa açıklık getirir. Artıklık kullanımıyla robot, istenilen uç noktası yörüngesini farklı eklem konfigürasyonlarıyla takip edebilmiştir. Bu şekiller dikkatli incelendiğinde, eklem açılarında meydana gelen küçük değişikliklerin robotun engelden sakınabilmesini sağladığı görülebilir. Benzetimle elde edilen eklem açı ve hızları, Mitsubishi PA 10 robot kolunun mekanik sınırları içinde kalmıştır. Bu sayede, sonuçlar gerçek zamanda da Mitsubishi PA 10 robot koluna uygulanabilmiş ve PA 10’ un engelden sakınarak istenilen yörüngeyi başarıyla yerine getirdiği görülmüştür. Aşağıdaki Şekil 5.50 sakınmasız ve sakınmalı (yaklaşık çözüm modeli ile) durumlara ilişkin robotun engele olan minimum uzaklıklarını göstermektedir.



Şekil 5.50 Sakınmasız ve Sakınmalı Durumlara İlişkin Robotun Engele Olan Minimum Uzaklıkları

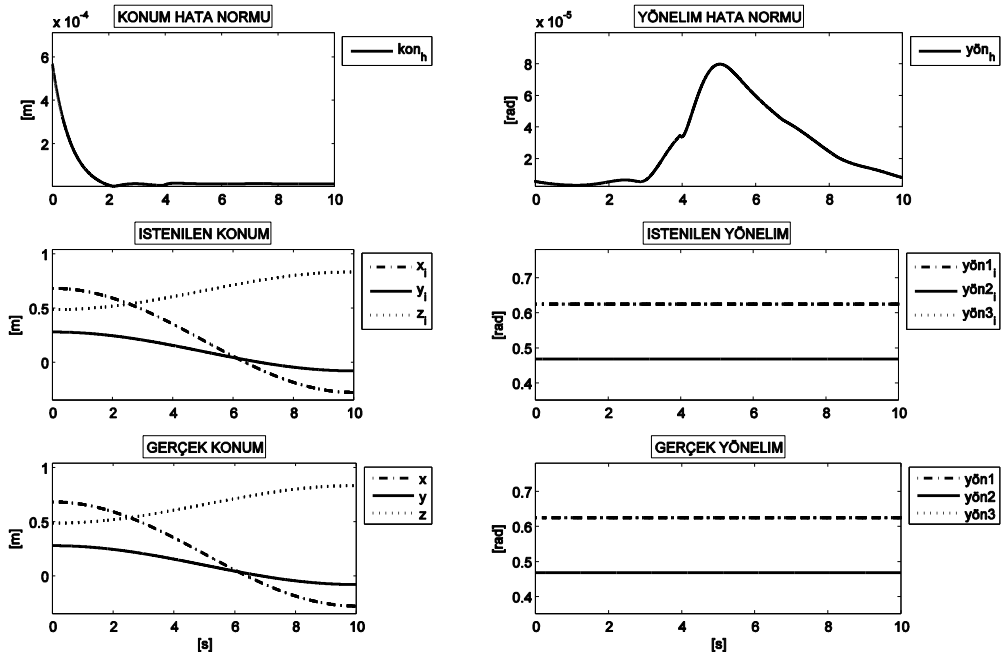
Şekilden de görüldüğü gibi sakınmasız halde altı saniye civarında minimum uzaklık sıfırın altına düşerken (yani robot engele çarparken), sakınma uygulanan durumda robotun engele en yakın olduğu uzaklık yaklaşık 10 cm civarındadır. Bu şekil engelden başarılı bir şekilde sakınıldığının bir göstergesidir.

İkinci aşamada yörünge altı için, engelden sakınmak amacıyla tam çözüm modeli kullanıldığında eklemlerin aldığı açı ve hız değerleri Şekil 5.51'de görülen şekildedir.



Şekil 5.51 Yörünge 6 İçin Tam Çözüm Modeli Kullanılarak Engellerden Sakınma Uygulandığında Robotun Eklem Açı ve Hızları

Aynı yörünge ve model için, robot kolu uç noktasının istenilen konum ve yönelimleri, gerçek konum ve yönelimleri ile konum ve yönelim hatalarının normları Şekil 5.52'de görülen şekilde elde edilmiştir.



Şekil 5.52Yörünge 6 İçin Tam Çözüm Modeli Kullanılarak Engellerden Sakınma Uygulandığında Robotun Uç Noktasının İstenilen, Gerçek Konum ve Yönelimleri İle Oluşan Hataların Normları

Tam çözüm modelinde edilen sonuçlarla da yüksek hata başarımı ile engelden sakınarak istenilen yörünge takip edilmektedir. Ancak Şekil 5. 51'de görülen ani hız değişimleri bu çözüm modelinin gerçek zamanda robota uygulanmasını imkansız kılmaktadır.

5.4 Genel Değerlendirmeler

Mitsubishi PA 10 robot kolunun sınırlamasız ters kinematik çözümleri, sırasıyla en küçük kareler, sabit ve adaptif çarpanlı sönümlendirilmiş en küçük kareler ve jakobian devriği yöntemleri kullanılarak elde edilmiştir. Test edilen yöntemler arasında hata başarımı en yüksek olan yöntem adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi olmuştur. En küçük kareler yöntemi tekil noktalar ve komşuluğunda yüksek eklem hızlarına neden olurken, sabit çarpanlı sönümlendirilmiş en küçük kareler yöntemi hareket sonunda eklem hızlarının sıfır olmamasına ve hata normlarının artmasına yol açmıştır. Jakobian devriği yöntemi ise, ancak bazı durumlarda ve kısmen doğru sonuçlar vermiştir.

Sonraki aşamada, Mitsubishi PA 10 robot kolunun artıklığını kullanmak amacıyla sınırlamalı ters kinematik çözümleri elde edilmiştir. Sınırlamalı çözümlerde, adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi kullanılmıştır. Eklemlerden birinin mekanik sınırları dışında çalıştığı bir yörünge için değişik ağırlık katsayılarıyla eklem mekanik sınırları içine çekilmiştir. Robotun istenilen hareketi gerçekleştirirken konumu belirlenen bir engele çarptığı bir başka yörünge için, yaklaşık ve tam çözüm modelleriyle engelden sakınma uygulanmış ve robotun istenilen hareketi engele çarpmadan gerçekleştirmesi sağlanmıştır. Benzetim sonucunda yaklaşık çözüm modelinin daha iyi sonuçlar verdiği görülmüştür.

Mitsubishi PA 10 robot kolunun artıklığı, seçilen yörüngeler için sınırlamalı çözüm hata başarımı ile sınırlamasız çözüm hata başarımı arasında bir fark doğurmamıştır. Ancak robotun sahip olduğu serbestlik derecesinden daha fazla serbestlik derecesi gerektiren görevlerde hata başarımları arasında bu farkın oluşacağı açıktır.

6 SONUÇLAR VE GELECEK ÖNERİLERİ

Günümüzde, robotların kullanım alanları oldukça genişlemiş ve genişlemesini sürdürmektedir. İnsan-robot etkileşiminde, eklemlerin hız yörüngeleri yerine, robot uç noktası hızı daha anlaşılır bir fiziksel büyüklük olarak ortaya çıkmaktadır. Bunun için gerekli olan ters kinematik çözümleri üzerine yapılmış ve yapılmakta olan pek çok çalışma vardır. Bu çalışma sırasında da bir artık robot olan Mitsubishi PA 10 için ters kinematik çözümleri iteratif olarak elde edilmiştir.

İteratif çözümde kullanılan yöntemlerin benzetim yoluyla sonuçları test edilmiştir. İlk aşamada, en küçük kareler yöntemi ile sonuçlar elde edilmiştir. Robot kolu tekil noktalardan uzakta hareket ettiğinde bu yöntem ile gerçek zamanda robot koluna uygulanabilir sonuçlar elde edilirken, robot kolu hareketi sırasında tekil noktalardan veya yakınlarından geçtiğinde, çok yüksek eklem hızları meydana gelmiş ve bu sonuçların gerçek zamanda robot koluna uygulanabilir olmadığı görülmüştür. Tekil noktalardaki sorunu aşmak için ikinci aşamada sabit çarpanlı sönümlendirilmiş en küçük kareler yöntemi kullanılarak sistem benzetimi yapılmıştır. Bu yöntemle, tekil noktalar ve komşuluğundaki bölgelerdeki yüksek eklem hızları gerçek zamanda robot koluna uygulanabilir düzeye çekilmiş, ancak benzetim sonunda eklem hızları sıfıra gitmemiştir. Ayrıca, robotun tekil noktalardan geçmediği yörüngelerde en küçük kareler yöntemi ile elde edilen hata normlarından daha büyük hata normlarının oluştuğu gözlenmiştir. Üçüncü aşamada, bu sorunun üstesinden gelmek için adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi kullanılarak sistem benzetimi yapılmıştır. Bu yöntemle, tekil noktalar ve komşuluğundaki bölgelerdeki yüksek eklem hızları önlenmiş ve eklem hızlarının benzetim sonunda sıfıra gitmesi sağlanmış, ayrıca robot tekil noktalardan uzakta hareket ederken en küçük kareler yöntemi ile aynı hata başarımına ulaşılarak hata normları istenilen seviyelere çekilmiştir. Bir sonraki aşamada hesaplama kolaylığı nedeniyle jakobian devriği yöntemi kullanılarak sistem benzetimi yapılmış, robot kolu istenilen yörüngeyi ancak bazı durumlarda ve büyük hata normları ile takip edebilmiştir. Yapılan karşılaştırma sonunda adaptif çarpanlı sönümlendirilmiş en küçük kareler yönteminin hata

başarımının en iyi olduğu görülmüştür. Benzetimle elde edilen sonuçların teoriyle tamamen uyumlu olduğu tespit edilmiştir.

Sonraki aşamada artıklık kullanılarak, hata başarımı en iyi olan adaptif çarpanlı sönümlendirilmiş en küçük kareler yöntemi ile mekanik eklem sınırlarından sakınma ve engellerden sakınma benzetimleri yapılmıştır. Sakınma uygulanmayan halde mekanik sınırları dışına çıkan eklem uzayı yörüngesi, sakınma uygulandığında bu sınırların içine çekilmiştir. Engellerden sakınma uygulaması için ise, konumu sabit küresel bir engele, sakınma uygulanmayan halde robot eklemlerinden birinin çarptığı bir yörünge seçilmiş ve benzetimi yapılmıştır. Bu yörünge için engellerden sakınma uygulandığında robot kolunun engele çarpmadan başka bir eklem konfigürasyonu ile istenilen yörüngeyi takip ettiği görülmüştür. Engellerden sakınma uygulaması literatürde genellikle düzlemsel robotlar üzerinde gerçeklenmektedir. Bu çalışmada elde edilen sonuçlarla, üç boyutlu uzayda hareket eden robotlarda da gerçeklenmesi oldukça anlamlıdır.

Son olarak, anlatılan yöntemler için benzetimle elde edilen sonuçlar, mekanik sınırlarının içinde olması şartıyla Mitsubishi PA 10 robot koluna uygulanmış ve gerçek zamanda da robotun istenilen yörüngeyi başarıyla takip ettiği görülmüştür.

Bu tezi takip edecek çalışmalarda, artıklık derecesi yüksek robot kollarının ters kinematik çözümleri elde edilebilir. Farklı optimizasyon ölçütleri için yapay sinir ağları ve bulanık mantık uygulamaları da bu amaç için kullanılabilir. PA-10' un dinamik modeli göz önüne alınarak ters dinamik problem çözümleri ileride yapılacak çalışmalar içerisindedir.

KAYNAKLAR

[1] Alfs, S. and Ivlev, O. and Martens, C. and Graser, A. 1999. Simulation Tool for Kinematic Configuration Control Technology for Dexterous Robots, *A Total Networked Industry Environment.* IECON'99, San Jose, CA, USA, vol. 1, p. 430-435.

[2] Barhen, J. and Gulati, S. and Zak, M. 1989. Neural Learning of Constrained Nonlinear Transformation, *IEEE Computer 22(6):* pp. 67-76

[3] Baerlocher, P. and Boulic, R. 1998. Task- Priority Formulation for the Kinematic Control of Highly Redundant Articulated Structures, *IEEE IROS '98*, pp 323–329,

[4] Behesti, M.H.T. and Tehrani, A. K. 1999. Obstacle avoidance for kinematically Redundant Robots using an Adaptive Fuzzy Logic Algorithm . *Proceedings of the American Control Conference.* Vol. 2, pp.1371-1375,

[5] Ben-Israel, A. 1965. A modified Newton-Raphson method for the solution of systems of equations, *Israel Journal of Mathematics*, Vol. 3, pp. 94 – 99,

[6] Ben-Israel, A. 1966. A Newton-Raphson method for the solution of systems of equations, *Journal of Mathematical Analysis and Applications*, Vol. 15, pp. 243 – 252,

[7] Ben-Israel, A. and Greville, T.N.E. 1974. Generalized inverses: theory and applications, *John Wiley & Sons*,

[8] Bjröck, A.1996. Numerical Methods for Least-Squares Problems, *SIAM, Philadelphia*,

[9] Boddy, C. L. and Taylor, J. D. 1993. Whole-arm reactive collision avoidance control of kinematically redundant manipulators, in *Proc. IEEE Int. Conf. Robotics Automat.*, Vol. 3, pp. 382–387.

[10] Cao, B.and Dodds, I.and Irwin, W. 1999. Redundancy resolution and obstacleavoidance for cooperative industrial robots. *Journal of Robotic Systems*, 16(7):405-417,

[11] Chang, P.H. 1987. A Closed-Form Solution for Inverse Kinematics of Robot Manipulators with Redundancy, *IEEE Trans. on Robotics and Automotion* Vol. RA-3, No.5, pp. 393-403.

[12] Chen, Y. and Vidyasagar, M. 1990. Optimal control of robotic manipulators in the presence of obstacles. *J. of Robotic Systems*, 7(5) pp. 721-740.

[13] Chen, Y. and Walker, I.D. 1993 A Consistent Null-Space Based Approach to Inverse Kinematics of Redundant Robots, *ICRA (3) 1993:* pp.374-381.

[14] Cheng, F.T.and Lu,Y.T. and Sun,Y.Y. 1998. Window-shaped obstacle avoidance for a redundant manipulator, *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 806–815.

[15] Cherif, A.R. and Daachi, B. and Benallegue, A. and Levy, N. 2002. Kinematic Inversion, *Proc. Intl. Conf. IEEE Intelligent Robots and System***,** vol.2 pp 1904- 1909.

[16] Cheung, E. and Lumelsky, V. 1988.Motion planning for robot arm manipulators with proximity sensing. In *Proc. Intl. Conf. On Robotics and Automation*, pages 740 . 745.

[17] Chiaverini, S. and Siciliano, B. and Egeland, O. 1994. Review of the Damped Least-Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator, *IEEE Trans. on Control Systems Techonology*, Vol. 2, No. 2, pp. 123-134.

[18] Chiaverini, S. 1997. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Trans. on Robotics and Automation*, 13(3):398. 410.

[19] Chiaverini, S. and Egeland, O. and Kanestrom, R.K. 1991. Achieving user-defined accuracy with damped least-squares inverse kinematics, *Int. Conf. Advanced Robotics (ICAR),* ,Vol.1 pp. 672-677.

[20] Chung, W. J. and Youm, Y. 1993.Null torque-based dynamic control for kinematically redundant manipulators. *Journal of Robotic Systems*, 10(6): pp. 811-833.

[21] Cleary, K. 1990. Incorporating Multiple Criteria in the Operation of RedundantManipulators, *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 618 – 624.

[22] Colbaugh, R and Seraji, H. and Glass, K.L. 1989.Obstacle avoidance for redundant robots using configuration control. *Journal of Robotic Systems*, 6(6):721-744.

[23] Das, H. and Slotine, J-J. and Sheridan, T.B. 1988. Inverse kinematic algorithms for redundant systems, *Proceeding of IEEE ICRA (International Conference on Robotics and Automation),* pp. 43 – 48.

[24] Delgado, K.K. and Long, M. and Seraji, H. 1990. Kinematic Analysis of 7-DOF Manipulators, *The Int. J. of Robotics Research*, Vol. 11, No. 5, pp. 469-481.

[25] Deo, A.S. and Walker, I.D. 1997. Minimum Effort Inverse Kinematics for Redundant Manipulators, *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 5, pp. 767 – 775.

[26] Ding, H. 1999. Recurrent Neural Networks for Minimum Infinity- Norm Kinematic Control of Redundant Manipulators. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 29, pp. 269-276.

[27] Driscoll, J.A. 2000. Comparision of Neural Network Architectures for the Modeling of Robot Inverse Kinematics. *Proc.eedings of the IEEE Southeastcon* pp. 44-51.

[28] Dordevic, G.S. and Rasic, M. and Kostic, D. and Surdilovic, D. 1999. Learning of Inverse Kinematics Behavior of Redundant Robot. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 3165-3170.

[29] Fiorini, P. and Seraji, H. and Long, M. 1997. A PC-based Configuration Controller for Dexterous 7-DOF Arms. *IEEE Robotics & Automotion Magazine*, pp. 30-38.

[30] Galicki, M. 1992. Optimal planning of a collision-free trajectory of redundant manipulators. *The International Journal of Robotics Research*, 11(6):549-559.

[31] Glass, K. and Colbaugh, R.and Lim, D. and Seraji, H. 1995.Real-time collision avoidance for redundant manipulators, *IEEE Trans. Robotics Automat* vol. 11, pp. 448–457.

[32] Garziera, R. 1994. Recursive Formulation for the Inverse Kinematics of Redundant Robots Performing Tasks with Priority Order, *ASME Journal of Mechanical Design*, Vol. 116, No. 1, pp. 337 – 338.

[33] Golub, G.H. and Van Loan, C.F. 1996. Matrix computations, *Third edition, Johns Hopkins University Press,*

[34] Guo, Z. Y. and Hsia,T. C. 1993.Joint trajectory generation for redundant robots in an environment with obstacles. *Journal of Robotic Systems*, 10(2):199-215.

[35] Hanafusa, H. and Yoshikawa, T. and Nakamura, Y. 1981. Analysis and Control of Articulated Robot with Redundancy, *IFAC, 8th Triennal World Congress*, Vol. 4, pp. 1927 – 1932.

[36] Ivlev, O. and Graser, A. 1997. An Analytical Method for the Inverse Kinematics of Redundat Robots. *Proc. Of the Third ECPD Intrnational Conference on advanced Robots, Intelligent Automation and Active Systems*. Bremen, 15-17, pp. 416-421.

[37] Ivlev, O. and Graser, A. 1998. A Resolving Redundancy of Series Kinematic Chains through Imaginary Links. *CESA298 IMACS Multiconference: Computational Engineering in Systems Applications.* Nabeul-Hammamet, Tunisia, 1-4, 1998, pp. 477-482.

[38] Kang. H.-J. and Freeman, R. A. 1993.Null space damping method for local joint torque optimization of redundant manipulators. *Journal of Robotic Systems*, 10(2):249-270.

[39] Kazerounian, K.and Wang, Z. 1988. Global versus local optimization in redundancy resolution of robotic manipulators. *The International Journal of Robotics Research*, 7(5)

[40] Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. of Robotic Research*, 5: pp. 90-98,

[41] Kim, J.O. and Khosla, P. 1992. Real-time obstacle avoidance using harmonicpotential functions, *IEEE Trans. Robotics Automat.*, vol. 8, pp. 338–349.

[42] Kircanski, M.and Vukobratovic, M. 1986. Contribution to control of redundant robotic manipulators in an environment with obstacles. *The International Journal of Robotics Research*, 5(4):112-119.

[43] Kircanski, M. 1993. Inverse Kinematic Problem Near Singularities for Simple Manipulators: Symbolical Damped Least- Squares Solution, *IEEE International Conference Robotics and Automation,. Proceedings* Vol.1, pp. 974-979.

[44] Klein, C.A. and Huang,C.H.1983. Review of Pseudoinverse control for Use with Kinematically Redundant Manipulators, *IEEE Trans. Systems, Man, Cybernetics*, Vol. SMC-13, pp. 245 – 250.

[45] Klein, C. A. and Blaho, B. E. 1987. Dexterity measures for the design and control of kinematically redundant manipulators. *The International Journal of Robotics Research*, 6(2):72-83.

[46] Kohonen, T. 1989. Self-Organization and Associative Memory, *Springer Series in Information Sciencies, Third Edition*,

[47] Liang, T.C. and Liu, J.S. 1999. An improved trajectory planner for redundant manipulators in constrained workspace. *Journal of Robotic Systems*, 16(6):339-351.

[48] Liégeois, A. 1977. Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms,*IEEE Trans. On Systems, Man, and Cybernetics,* Vol. SMC-7, No. 12, pp. 868 – 871.

[49] Luetkepohl, H. 1996. Handbook of Matrices, *John Wiley & Sons.*

[50] Mao, Z. and Hsia, T. C. 1997. Obstacle avoidance inverse kinematics solution of redundant robots by neural networks, *Robotica*, Vol. 15, pp. 3–10.

[51] Maciejewski, A.A. and Klein, C.A. 1985 Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments, *The International Journal of Robotics Research*, Vol. 4, No. 3, pp. 109 – 117.

[52] Maciejewski, A.A. and Klein, C.A. 1988. Numerical Filtering for the Operation of Robotic *Manipulators through Kinematically Singular Configurations, Journal of Robotic Systems,* Vol. 5, No. 6, pp. 527 – 552.

[53] Maciejewski, A.A. and Klein, C.A. 1989. The SVD Computation and Applications to Robotics, *International Journal of Robotics Research*, Vol. 8, No. 6, pp. 63 – 79.

[54] Maciejewski, A.A. 1990. Dealing with the Ill-Conditioned Equations of Motion for Articulated Figures, *IEEE Computer Graphics and Applications*, Vol. 10, No. 3, pp. 63 – 71.

[55] Mayorga, R. V.and Janabi-Shari, F. and Wong, A. K. C. 1995. A fast approach for the robust trajectory planning of redundant robot manipulators. *Journal of Robotic Systems,*12(2):147-161.

[56] Mayorga, R. V. and Milano, N. and Wong, A. K. C. 1993.A fast procedure for manipulator inverse kinematics computation and singularities prevention. *Journal of Robotic Systems*, 10(1):45-72.

[57] McGhee, S. and Chan, T. F. and Dubey, R. V. 1994. Probability-Based Weighting of Performance Criteria for a Reduntant Manipulator, *IEEE Conference*, pp. 1887 – 1894.

[58] Moeslund, T.B. and Granum, E. 2001. Pose Estimation of a Human Arm Using Kinematic Constraints, *Proc. 12th Scandinavian conference on image analysis (SCIA'01),Bergen, Norway, June 11-14.*

[59] Mussa, F.A. and Ivaldi and Hogan, N. 1991. Integrable solutions of kinematic redundancy via impedance control. The *International Journal of Robotics Research*, 10(5):481-491.

[60] Omidvar, O. 1997. Neural Systems for Robotics. *Academic Press.*

[61] Oyama, E. and Tachi, S. 1999. Inverse Kinematics Learning by Modular Architecture Neural Networks, *IEEE International Joint Conference on Neural Networks.*

[62] Nair, D. and Aggarwal, J. K. 1998. Moving obstacle detection from a navigating robot," *IEEE Trans. Robotics Automat.*, Vol. 14, pp. 404–416.

[63] Nakamura, Y. 1991. Advanced robotics redundancy and optimization. *Addison-Wesley Publishing Company.*

[64] Nakamura, Y. and Hanafusa, **H**. 1986. Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control, *Journal of Dynamic Systems, Meas., and Control*, Vol. 108, pp. 163 – 171.

[65] Nakamura, Y. and Hanafusa, Yoshihawa, T. 1987. Task-Priority Based Control of Robot Manipulators, *The International Journal of Robotics Research*, Vol. 6, No. 2, pp. 3 -15.

[66] Nanayakkara, T. 2003. Dynamic Controlling of an Industrial Seven-Link Manipulator by Evolving Runge-Kutta- Gill RBF Networks, *In Proc. of Robotics and Mechatronics Conference*, 1A1-29-038.

[67] Nanayakkara, T. and Watanabe, K. and Kıguchı, K. and Izumi, K. 2000. Fuzzy self- adaptive radial basis function neural network- based control of a seven- link redundant industrial manipulator. Advanced Robotics, Vol.15, pp.17-43.

[68] Nashed (ed), Z.1976. Generalized Inverses and Applications, *Academic Press.*

[69] Nenchev, D.N. 1992. Restricted Jacobian matrices of redundant manipulators in constrained motion tasks. *The International Journal of Robotics Research*, 11(6):584-597.

[70] Nenchev, D.N.and Tsumaki, Y. 2003. Motion Analysis of a Kinematically Redundant Seven-DOF Manipulator Under the Singularity-Consistent Method. *Proceedings of the 2003 IEEE International Conference on Robotics & Automation Taipei, Taiwan,*

[71] Nenchev, D.N.and Tsumaki, Y. **and Uchiyama, M.** 2000. Singularity-Consistent parameterization of robot motion and control, *The Int. Journal of Robotics Research*, Vol. 19, No. 2, pp. 159-182.

[72] Neumair, A. 1998. Solving ill-conditioned and singular linear systems: A tutorial on regularization, *SIAM Review*, Vol. 40, pp. 636 – 666.

[73] Ortega, J.M. and Rheinboldt, W.C 1970. Iterative solution of nonlinear equations in several variables, Academic Press, New York,

[74] Paul, R. 1981. Robot Maniputors: Mathetamitcs, Programming and Control, *MIT Press*,

[75] Pourazady, M, and Ho, L. 1991.Collision avoidance control of redundant manipulators. Mechanism and Machine Theory, 26(6):603-611.

[76] Pourboghrat, F. 1990. Neural Networks for Learning Inverse-Kinematics of Redundant Manipulators. Proceedings of Circuits and Systems, the 32nd Midwest Symposium Vol.2, pp**.** 760-762.

[77] Press, W.H. and Teukolsy, S.A. and Vetterling, W.T. and Flannery B.P. 1992. Numerical Recipes in C, *Second Edition, Cambridge University Press.*

[78] Rubi, J. and Rubio, A. and Avello, A. 2002. Involving the Operation in a Singularity Avoidance Strategy for a Redundant Slave Manipulator in a Teleoperated Application, *Int. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland.

[79] Sciavicco, L. and Siciliano, B. 1988. A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators, *IEEE Journal of Robotics and Automation*, Vol. 4, pp. 403 – 410.

[80] Schafer, B. and Krenn, R. and Rebele, B. 2002. On Inverse Kinematics and Kinetics of Redundant Space Manipulator Simulation, *Journal of Computational and Applied Mechanics*, Vol. 4., No. 1, pp. 53—70.

[81] Schreiber, G. and Hirzinger, G. 1998. Singularity Consistent Inverse Kinematic By Enhancing The Jacobian Transpose, *In: ARK 1998 Strobl/Wolfgangsee Advances in Robot Kinematics: Analysis and Control, Kluwer Academic Publishers, Dordrecht*, pp 475-483.

[82] Seraji H. 1989. Configuration Control of Manipulators: Theory and Implementation. *IEEE Trans. on Robotics and Automotion* Vol. 5, No. 4, pp. 472-490.

[83] Seraji, H. and Colbaugh, R. 1990. Improved configuration control for redundant robots. *Journal of Robotic Systems*, 7(6): pp. 897-928.

[84] Seraji, H. and Bon, B. 1999.Real-time collision avoidance for position-controlled manipulators. *IEEE Trans. on Robotics and Automation*, 15(4): pp. 670-677.

[85] Siciliano, B. and Slotine J-J. 1991. A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems, *ICAR '91*, pp. 1211 – 1216.

[86] Sims, K. and Zeltzer, D. 1988. A Figure Editor and Gait Controller for Task Level Animation, *SIGGRAPH Course notes (Synthetic Actors: the Impact of Artificial Intelligence and Robotics on Animation).*

[87] Spong, M.W. and Vidyasagar, M. 1989. Robot Dynamics and Control. *: John Wiley, New York.*

[88] Tikhonov, A.N. 1963. Solution of incorrectly formulated problems and the regularization method, *Soviet Mah. Dokl.*, Vol. 4, pp. 1035 – 1038.

[89] Tsmaki, Y. 2001. A Numerical SC approach for a teleoperated 7-DOF manipulator, *in Proc. 2001 IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, pp. 1039-1044.

[90] Wampler, C.W. 1986. Manipulators inverse kinematic solutions based on vector formulations and damped least-squares method, *IEEE Trans. Syst., Man, Cybernetics*, Vol. 14, pp. 93 – 101.

[91] Wang, D. and Hamam, Y. 1992.Optimal trajectory planning of manipulators with collision detection and avoidance. *The International Journal of Robotics Research*, 11(5):460-468.

[92] Wang, Y.J. and Lin, C.T. 1998. Runge- Kutta Neural Network for Identification of Dynamical Systems in High Accuracy. *IEEE Transactions on Neural Networks*, Vol. 9., No. 2, pp. 294-307.

[93] Welman, C. 1993. Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation, *Master Thesis, Simon Fraser University*.

[94] Xia, Y. and Wang, J. 2001. A Dual Neural Network for Kinematic Control of Redundant Robot Manipulators**.** *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* , Vol. 31, no. 1, pp. 147-154.

[95] Xie, H.P. and Patel, R.V. and Kalaycioglu, S. and Asmer, H. 1998. Realtime collision avoidance for a redundant manipulator in an unstructured environment. In *Proc. Intl. Conf. On Intelligent Robots and Systems IROS'98 Victoria, Canada*, pp. 1925-1930.

[96] Xu, Y. and Nechyba, M.C. 1993. Fuzzy Inverse Kinematic Mapping: Rule Generation, Efficiency, and Implementation. *Proceedings of International Conference on Intelligent Robots and Systems*, Vol. 2, July. pp. 911 - 918.

[97] Yoshikawa, T. 1984. Analysis and Control of Articulated Robot with Redundancy.*The First International Symposium*, ed. M. Brady and R. Paul (MIT Press, 1984), pp. 735 – 747.

[98] Yoshikawa, T. 1985. Manipulability and Redundancy Control of Robotic Mechanism, *The international Journal of Robotics Research*, Vol.4, pp.3-9.

[99] Wampler, C.W. 1987. Inverse Kinematic Functions for Redundant Manipulators. Proc. *IEEE Intl. Conf. on Robotics and Automotion*, Raleigh, NC, pp. 393 – 403.

[100] Whitney, D.E.1969. Resolved Motion Rate Control of Manipulators and Human Prostheses, *IEEE Trans. On Man-Machine Systems*, Vol. MMS-10, No. 2, pp. 47 – 53.

[101] Wolovich, W.A. and Elliot, H.1984. A computational technique for inverse kinematics, *Proc. Of 32nd Conference on Decision and Control*, pp. 1359 – 1362.

[102] Zlajpah, L. 1998.Simulation of *n*-r planar manipulators. *Simulation Practice and Theory*, 6(3):305 . 321.

[103] Zlajpah, L. and Nemec, B. 2002. Kinematic Control Algorithms for On -Line Obstacle Avoidance for Redundant Manipulators, Intelligent Robots and System, 2002. IEEE/RSJ International Conference on ,Vol.2 pp. 1898-1903.

[104] Zhang, Y. 2002. A Dual Neural Network for Bi-Criteria Kinematic Control of Redundant Manipulators. *IEEE Transactions on Robotics and Automotion* 18 (6): 923-931.

[105] Zhang, Y. 2003.. A Dual Neural Network for Redundancy Resolution of Kinematically Redundant Manipulators Subject to Joint Limits and Joint Velocity Limits, *IEEE Transactions on Neural Networks* 14 (3): 658-667.

[106] Zhang, Y and Wang, J. 2004. Obstacle Avoidance for Kinematically Redundant Manipulators Using a Dual Neural Network. *IEEE Transactions on Syst. Man. Cy. B* 34 (1): 752-759.

ÖZGEÇMİŞ

Bülent ÜNVER 15 Ağustos 1978 tarihinde Akşehir'de doğdu. İlk ve orta öğrenimini Akşehir'de tamamladı. 1997 yılında İstanbul Teknik Üniversitesi Elektronik ve Haberleşme bölümünde lisans öğrenimine başladı ve 2002 yılında mezun oldu. 2002 yılında İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Programında lisansüstü öğrenimine başladı.