

Endüstriyel Robot Kollarıyla 3B Modelleme Amaçlı Benzetim Ortamı Tasarımı

Serkan Ergür

**YÜKSEK LİSANS TEZİ**

Elektrik Elektronik Mühendisliđi Anabilim Dalı

Haziran 2015

Simulation Environment Design For 3D Modeling Purpose with Industrial Robot Arm

Serkan Ergür

**MASTER OF SCIENCE THESIS**

Department of Electrical and Electronics Engineering

June 2015

Endüstriyel Robot Kollarıyla 3B Modelleme Amaçlı Benzetim Ortamı Tasarımı

Serkan Ergür

Eskişehir Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliği Uyarınca  
Elektrik Elektronik Mühendisliği Anabilim Dalı  
Kontrol ve Kumanda Sistemler Bilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır

Danışman: Yrd. Doç. Dr. Metin Özkan

Haziran 2015

## ONAY

Elektrik Elektronik Mühendisliđi Anabilim Dalı Yüksek Lisans öğrencisi Serkan Ergür'ün YÜKSEK LİSANS tezi olarak hazırladığı “Endüstriyel Robot Kollarıyla 3B Modelleme Amaçlı Benzetim Ortamı Tasarımı” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

**Danışman** : Yrd. Doç. Dr. Metin ÖZKAN

**İkinci Danışman** : -

**Yüksek Lisans Tez Savunma Jürisi:**

**Üye** : Yrd. Doç. Dr. Metin ÖZKAN

**Üye** : Prof. Dr. Osman PARLAKTUNA

**Üye** : Yrd. Doç. Dr. Ahmet YAZICI

**Üye** : Yrd. Doç. Dr. Kemal ÖZKAN

**Üye** : Yrd. Doç. Dr. Muammer AKÇAY

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve  
..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Hürriyet ERŞAHAN  
Enstitü Müdürü

## ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Yrd. Doç. Dr. Metin ÖZKAN danışmanlığında hazırlamış olduğum “ Endüstriyel Robot Kollarıyla 3B Modelleme Amaçlı Benzetim Ortamı Tasarımı ” başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 22/06/2015

Serkan ERGÜR

İmza

## ÖZET

Günümüzde gelişen teknoloji ile birlikte fabrikalarda endüstriyel robotların kullanımı artmıştır. Robot teknolojilerinde yapılan yoğun çalışmalara rağmen yeterli düzeye ulaşmış değildir. Bunun sebeplerinden biri ise gerçek bir robot üzerinde proje geliştirmenin maliyetli olmasıdır. Bu noktada, benzetim yazılımlarının kullanımı önem arz etmektedir. Bu çalışmada, 3B modelleme amaçlı benzetim ortamı tasarımı yapılması hedeflenmiştir. Cisimlerin, 3B robotik modellemesi sürecinde önemli bir problem üzerinde araştırmalar devam etmektedir. Bu problem, robot kolunun cisim yüzeyinden veri toplayacak olan algılayıcıyı dolaştıracağı pozisyon ve yönelimlerin belirlenmesidir. Bu çalışmalar robot üzerinde çok sayıda denemeler gerektirmekte ve gerçek robot ile bu zaman alıcı ve maliyetli bir süreç oluşturmaktadır. Ayrıca, tez çalışmasının bir başka hedefi bir firma tarafından döküm parçalarındaki çapakların temizlenmesi amacıyla geliştirilen ve prototip üretimi yapılan 5 eksenli SCARA ( Selective Compliance Assembly Robot Arm ) tipi robot için yörünge planlaması, kinematik hesaplamalar ve benzetim çalışmaları yapmaktır. 3B modelleme amaçlı benzetim ortamı da bu robot modeli için gerçekçi parametreler ile oluşturulmuştur. Yörünge planlaması, kartezyen uzayında robot uç noktasının doğrusal ve ara nokta üzerinden doğrusal hareketi için geliştirilmiştir.

**Anahtar Kelimeler:** 5 eksenli robot kolu, Kinematik, yörünge planlama, benzetim ortamı, 3B modelleme.

## SUMMARY

Today, the usages of industrial robots have increased with emerging technologies in the factories. Despite intensive studies on robot technology, it has not reached a satisfactory level. One reason for this is the high cost of the project development on a real robot. At this point, it is important to use the simulation software. In this study, simulation environment design for 3D modeling is planned. Researches for a problem about 3D modeling by the aim of robotic manipulators already continues. This problem is defined as the determination of the position and orientation of robot manipulator with laser profile sensor for the purpose of gathering 3D data from the object's surface. It requires a large number of experiments on the real robot. However, this development process is time-consuming and costly with real robots. In addition, another purpose of the thesis is to develop trajectory planning, kinematics calculation and simulation studies for 5 axis SCARA (Selective Compliance Assembly Robot Arm) type robot which is designed and prototyped by a company for the purpose of deburring on the casting. Simulation environment design for 3D modeling has been created with realistic parameters for this robot model. Trajectory planning in Cartesian space has been developed for the tip of the robot with linear and linear movement through via point.

**Keywords:** 5 axis robot arm, kinematics, trajectory planning, simulation environment, 3D modeling.

## TEŞEKKÜR

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak yaptığım bu çalışmamda, bana danışmanlık ederek, beni yönlendiren ve her türlü olanağı sağlayan danışmanım Yrd. Doç. Dr. Metin Özkan'a sonsuz teşekkürlerimi sunarım.

Tez çalışmalarım boyunca sonsuz bir güven ve büyük bir sabırla desteğini esirgemeyen aileme sonsuz teşekkürlerimi sunarım.

Tez çalışması için geliştirdikleri robot kolunun kullanılmasına izin veren Robotex firmasına sonsuz teşekkürlerimi sunarım.



## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET .....</b>	<b>vi</b>
<b>SUMMARY .....</b>	<b>vii</b>
<b>TEŞEKKÜR .....</b>	<b>viii</b>
<b>İÇİNDEKİLER .....</b>	<b>ix</b>
<b>ŞEKİLLER DİZİNİ .....</b>	<b>xi</b>
<b>ÇİZELGELER DİZİNİ .....</b>	<b>xiii</b>
<b>SİMGELER VE KISALTMALAR DİZİNİ .....</b>	<b>xiv</b>
<b>1. GİRİŞ VE AMAÇ .....</b>	<b>1</b>
<b>2. LİTERATÜR ÖZETİ .....</b>	<b>2</b>
<b>3. YÖRÜNGE PLANLAMASI .....</b>	<b>7</b>
3.1. Eklem Uzayında Yörünge Planlaması .....	8
3.1.1. Kübik polinomlar ile yörünge planlaması .....	8
3.1.2. Kübik polinomlar ile ara noktalar için yörünge planlaması .....	10
3.1.3. Yüksek dereceden polinomlar ile yörünge planlaması .....	10
3.1.4. Parabolik eklemeler ile doğrusal fonksiyon .....	11
3.2. Kartezyen Uzayında Yörünge Planlaması .....	13
<b>4. SİSTEM TASARIMI .....</b>	<b>17</b>
4.1. Endüstriyel Robot Kol Tasarımı .....	17
4.1.1. İleri kinematik .....	17
4.1.2. Ters kinematik .....	20
4.2. Benzetim Ortamı Tasarımı .....	23
4.2.1. Robot işletim sistemi (ROS) .....	23
4.2.2. Gazebo .....	24
4.2.3. Gazebo ile kullanmak üzere robot modeli oluşturma .....	26

## İÇİNDEKİLER (devam)

	<u>Sayfa</u>
<b>5. ÖNERİLEN YAKLAŞIM .....</b>	<b>32</b>
5.1. Robot Denetimi (ROS Düğümü) .....	34
5.1.1. Yörünge planlama .....	37
5.1.1.1. <u>Kartezyen uzayında doğrusal hareket</u> .....	37
5.1.1.2. <u>Kartezyen uzayında doğrusal hareket testleri</u> .....	40
5.1.1.3. <u>Kartezyen uzayında ara nokta ile doğrusal hareket</u> .....	46
5.1.1.4. <u>Kartezyen uzayında ara nokta ile doğrusal hareket testleri</u> .....	48
5.1.2. Yörünge planlama algoritması .....	54
5.2. Veri Toplama ve Kayıt (ROS Düğümü) .....	54
5.3. 3B Görselleştirme .....	56
<b>6. UYGULAMA .....</b>	<b>58</b>
<b>7. SONUÇ VE ÖNERİLER .....</b>	<b>63</b>
<b>KAYNAKLAR DİZİNİ .....</b>	<b>64</b>

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
3.1. Uç işlevcisinin a) eklem uzayında b) kartezyen uzayında takip ettiği yörüngeyi başlangıç, bitiş ve ara noktaları .....	7
3.2. Üçüncü dereceden kübik bir yörüngeye ait a) konum, b) hız, c) ivme grafiği .....	9
3.3. Doğrusal Yörünge .....	11
3.4. Parabolik kısımlar eklenmiş doğrusal yörünge .....	12
3.5. Parabolik kısımlar eklenmiş doğrusal yörüngeye ait a) konum, b) hız, c) ivme grafikleri .....	13
3.6. Kartezyen yolu sorunu tip 1 .....	14
3.7. Kartezyen yolu sorunu tip 2 .....	15
3.8. Kartezyen yolu sorunu tip 3 .....	16
4.1. Robot Kol Parametreleri .....	18
4.2. Örnek Link Görüntüsü .....	26
4.3. Örnek Link Modeli .....	27
4.4. Örnek Link Modelinin Gazebo Görüntüsü .....	27
4.5. Örnek Eklem Görüntüsü .....	28
4.6. Örnek Eklem Modeli.....	28
4.7. Robotun Gerçek Görüntüsü (Robotics Solutions Ltd.) .....	29
4.8. Linklerin Ayrı Ayrı Gazebo Görüntüleri a) link 0, b) link 1, c) link 2, d) link 3, e) link 4, f) link 5.....	30
4.9. Robot Modelin Link ve Eklemlerinin Gazebo Görüntüsü .....	31
5.1. Sistemin Genel Yapısı.....	33
5.2. Robot Denetimindeki Mesajlaşmalar .....	35
5.3. Robot Denetimi akış diyagramı .....	36
5.4. Kartezyen uzayında iki nokta arasında örnek doğrusal hareket .....	37
5.5. Doğrusal yörünge için yol, hız ve ivme grafiği .....	38
5.6. Doğrusal harekette yörünge planlamasının akış diyagramı .....	39
5.7. İki nokta arası doğrusal hareket için planlanan yörünge .....	41
5.8. İki nokta arası doğrusal hareket için eklem değişkenleri .....	42
5.9. İki nokta arası doğrusal hareket için eklem hızları .....	43

## ŞEKİLLER DİZİNİ (devam)

<u>Sekil</u>	<u>Sayfa</u>
5.10. İki nokta arası doğrusal hareket için eklem ivmeleri .....	44
5.11. a-k Matlab ortamında iki nokta arası yörünge planlaması esnasında elde edilen benzetim görüntüleri .....	45
5.12. Ara nokta ile doğrusal hareket .....	46
5.13. Ara nokta ile harekette a) $V_1 > V_2$ , b) $V_1 < V_2$ , c) $V_1 = V_2$ olduğu durumlar .....	47
5.14. Ara nokta ile doğrusal hareket için planlanan yörünge .....	49
5.15. Ara nokta ile doğrusal hareket için eklem değişkenleri .....	50
5.16. Ara nokta ile doğrusal hareket için eklem hızları .....	51
5.17. Ara nokta ile doğrusal hareket için eklem ivmeleri .....	52
5.18. a-f Matlab ortamında ara nokta ile yörünge planlaması esnasında elde edilen benzetim görüntüleri .....	53
5.19. Veri Toplama ve Kayıt'daki Mesajlaşmalar .....	55
5.20. Veri Toplama ve Kayıt akış diyagramı .....	55
5.21. Lazer verilerinin referans koordinat sistemine göre XYZ noktalarının belirlenmesi.	57
6.1. Kübik cismin ve robotun gazebo benzetim ortamı görüntüsü .....	58
6.2. Cismin taranması için izlenmesi beklenen yörünge .....	59
6.3. Tarama esnasından görüntüler (a-k) .....	60
6.4. Tarama sonrası elde edilen görüntü (Farklı Açılardan) (a-f) .....	61

**ÇİZELGELER DİZİNİ**

<b><u>Çizelge</u></b>	<b><u>Sayfa</u></b>
4.1. Denavit-Hartenberg (DH) Parametreleri .....	18
6.1. Yörünge Planlamasında Kullanılacak Parametreler .....	59

**SİMGELER VE KISALTMALAR DİZİNİ**

<b><u>Simge</u></b>	<b><u>Açıklama</u></b>
$c\theta_i$	Kosinüs teta i
$s\theta_i$	Sinüs teta i
$\theta_i$	Teta i (pozisyon)
$\dot{\theta}_i$	Teta i'nin türevi (hız)
$\ddot{\theta}_i$	Teta i'nin ikinci türevi (ivme)

<b><u>Kısaltma</u></b>	<b><u>Açıklama</u></b>
D-H	Denavit-Hartenberg
SCARA	Seçici uyum montaj robot kol
CAD	Bilgisayar Destekli Tasarım
ROS	Robot İşletim Sistemi
PID	Oransal-İntegral-Türevsel

## 1. GİRİŞ VE AMAÇ

Bu çalışmada, ocak üstü döküm ızgaralarda oluşan çapakları temizlemek amacıyla tasarlanmış ve prototip üretimi gerçekleştirilmiş olan 5 eksenli SCARA ( Selective Compliance Assembly Robot Arm ) tipi robot için, kinematik hesaplamalar, yörünge planlaması, benzetim ortamı tasarımı yapılmaktadır.

Bu çalışmadaki temel motivasyonu, Eskişehir Osmangazi Üniversitesi Teknoparkı bünyesinde bulunan Robotex Firmasının tasarımı ve prototip üretimini yaptığı, çapak alma amaçlı endüstriyel robot için kinematik hesaplamalara, yörünge planlaması ve benzetim ortamı tasarımına ihtiyaç olmasıdır. Böylece, bu çalışma sonuçları ile, üç kısımda literatür katkısı sağlanacağı düşünülmektedir. Bunlardan birisi, yaygın olarak bilinen SCARA tipi robotlardan farklı olarak 4 eksen yerine 5 eksene sahip robot kol için kinematik hesaplamaların sunulmasıdır. İkinci katkı, doğrusal ve ara noktalar üzerinden doğrusal yörünge planlamasının kodlama yapılabilecek şekilde algoritmik formda sunulmasıdır. Üçüncü katkı ise, tezin başlığını da oluşturan 3B modelleme amaçlı benzetim ortamı tasarımı ve uygulamasıdır. Son zamanlarda otonom robotik 3B modelleme yöntemlerinin geliştirilmesine yönelik yoğun çalışmalar yapılmaktadır. Bu çalışmalarda önerilen yöntemlerin gerçek robot üzerinde denenmesi, maliyetli ve zaman alıcı bir süreçtir. Geliştirilen benzetim ortamı ile bu süreçte araştırmacılara fayda sağlanacağı öngörülmektedir.

Tezin bundan sonraki kısmında, çalışmaların sunumu şu şekilde olacaktır: Bölüm 2'de literatür özeti sunulacaktır. Bu kısımda, robotun tasarım amacını oluşturan çapak alma çalışmaları, çapak alma amaçlı hareket planlama çalışmaları, genel hareket planlama çalışmaları ve robot kol ile 3B modelleme çalışmaları verilecektir. Bölüm 3'de yörünge planlama anlatılacaktır. Bölüm 4'de, robot kolun tasarımı ve robot kol için benzetim ortamının oluşturulması anlatılmaktadır. Bölüm 5'de 3B modelleme amaçlı benzetim ortamının mimari tasarımı ve ortam tarafından kapsanan Kartezyen uzayında doğrusal yörünge planlama yöntemleri anlatılmaktadır. Bölüm 6'da, uygulamalar verildikten sonra, son bölümde elde edilen sonuçlar yorumlanmaktadır.

## 2. LİTERATÜR ÖZETİ

Literatür incelendiğinde yörünge planlaması, 3 boyutlu model oluşturma konularında birçok çalışma mevcuttur. Projede tercih edilen robot çapak alma için geliştirildiğinden, robot kolları ile çapak alma konusundaki çalışmaların incelenmesinin katkı sağlayacağı düşüncesiyle bu konudaki çalışmalarda incelenmiştir. Literatürde bulunan bu çalışmalar;

- Çapak alma ile ilgili çalışmalar,
- Hem çapak alma hem hareket planlaması ile ilgili çalışmalar,
- Hareket planlaması ile ilgili çalışmalar,
- Robot kullanımı ile 3 Boyutlu model oluşturma ile ilgili çalışmalar,

olarak 4 grupta sunulmaktadır.

***Robotik sistemler kullanılarak yapılan çapak alma ile ilgili çalışmalar, üretim süreçlerinde önemli bir problem oluşturması nedeniyle, literatürde yaygın olarak bulunmaktadır.***

Kazerooni (1988), empedans kontrolü kullanarak, otomatik çapak alma gerçekleştirmektedir. Kuvvet sensörü yardımıyla tanjant ve normaldeki kuvvetlere göre çapakların yerleri belirlenerek ve hız azaltılarak çapak temizleme yapılmaktadır. Hsu ve Fu (2000), uyarlamalı bulanık hibrit pozisyon/kuvvet kontrolü kullanarak çapak alma için öneride bulunmuşlardır. Öneriye göre robotun hareketlerine otomatik karar veren dışsal bir döngü ile çapağın temizlenmesine karar veren içsel döngüden oluşan bir sistemdir. Önerilen uyarlamalı bulanık denetleyici köşe noktalarındaki belirsizlikleri giderebilen B-Spline fonksiyonu kullanarak kapalı döngü sistemin kararlılığını garanti etmektedir. Asakawa vd. (2002), CAD (Bilgisayar Destekli Tasarım) veri temeline serbest eğri yüzey üzerinde otomatik çapak alma ile ilgili bir çalışma gerçekleştirmiştir. Dumitras (2007), kesme işlemi parametreleri ve yüzey kalitesi, kesme kuvvetleri ve çapak alma oranı arasındaki ilişkiyi kuran bir matematiksel model sunmaktadır. Buckmaster vd. (2008), yeni bir hibrid doğal giriş kontrol yöntemi sunmaktadır. CAD modelden yararlanılmaktadır. Parçanın yönelimini ayarlamak için parçanın belli noktalarına



dokunarak düzeltmeler yapılmaktadır. Liao vd. (2008), parlatma ve cilalama için yaklaşım sunmaktadır. Bir basınç sensörü ve doğrusal enkoder kullanılmıştır. Parlatma kontrolü için verilen parçanın geometrisine göre önceden planlanmış takım basıncı ve bir Proportional - Integral - Derivative (Oransal - İntegral - Türevsel) denetleyici basınç takibi için kullanılmaktadır. Çapak alma kontrolü için başka bir PID denetleyici algılama aracı uzantısı ile takım uzunluğunu düzenlemek için uygulanmaktadır. Mohammad vd. (2008), DELMIA (R) robotu kullanılarak çapak alma işlemi benzetim ortamında anlatmıştır. Lee vd. (2009), uyarlamalı kuvvet kontrol kullanarak robot çapak alma sistemi için doğrudan öğretim ve yeniden oynatma metodu kullanmıştır. İlk önce doğrudan öğretim metodu ile yörünge kaydedilir, sonra Douglas-Peucker (DP) algoritması ile bu yörünge yumuşatmak için sıkıştırılır. Daha sonra yeniden oynatma metodu ile kaydedilen yörünge otomatik olarak gerçekleştirilmektedir. Adaptif kuvvet kontrol yardımı ile de çapaklar tespit edilip temizlenmektedir.

***Çapak temizleme amaçlı hareket planlamasını kapsayan çalışmalarda literatürde bulunmaktadır.***

Murphy vd. (1998), CAD verilerinden yararlanarak hem yörünge planlaması yapıp hem de çapak alma işlemi gerçekleştirmiştir. CAD verisi bir grafik ara yüzünden geçirilerek bir yörünge planlanır. Daha sonra robot bu yörüngede ilerlerken kuvvet geri bildirimini ile yörünge düzeltmeleri yapılır ve son yörünge belirlenmiş olur. Kuvvet geri bildirimini ile ayrıca çapakların hangi tür uç ile temizlenmesi gerektiği de belirlenir. Jinno vd. (1999), şekli bilinmeyen bir cismin kamera ile şeklini belirleyip, kuvvet sensörü yardımıyla da yörünge planlaması yapmaktadır. Daha sonra ise çapak alma işlemi yapılmaktadır. Jayaweera ve Weeb (2011), MXS algılayıcı ile malzeme taranıyor ve elde edilen veriler MATLAB da işleniyor ve yörünge planlaması oluşturulmaktadır. Daha sonra kesme ucu ile çapak alma işlemi yapılıyor. Rososhansky ve Xi (2011), kapsama tabanlı bir yol planlaması üzerine çalışmışlardır. Amaç, cilalanacak veya temizlenecek yüzeyin tamamının temizlenmesidir. Temizlenmemiş nokta kalmamalıdır veya aynı noktadan iki defa geçilmemelidir. Bunun için temizleme aparatının çapı ile temizlenecek yüzeyin çapı arasında en uygun oranın belirlenip, temizleme yapılmaktadır. Song vd. (2012), CAD model ve öğretim noktaları yöntemiyle önce yörünge planlaması yapıp sonra çapak alma yapmaktadır. Çalışmada öncelikle CAD modelden yörünge belirlenir, daha sonra öğretim

noktalarından elde edilen yörünge ile karşılaştırılıp oryantasyon hataları düzeltilir. Kuvvet sensörü yardımı ile de çapakların yerleri belirlenir. Sonraki aşamada ise temizleme işlemi gerçekleştirilir.

**Literatürde çok sayıda, hareket planlaması ile ilgili çalışmalar bulunmaktadır.**

Jeon vd. (1993), kolayca değiştirilebilir hız profilleri için etkin bir yörünge oluşturma tekniği önermiştir. Bu tekniğe göre bir zaman için en iyi hız profilli bir hareket oluşturulur. Daha sonra pürüzsüz bir hız profilinin en iyi hız profili ile her örnekleme zamanında çarpılması ile elde edilir. Önerilen tekniğin karmaşıklığı ve mevcut tekniklerle karşılaştırılması, tekniğin daha verimli olduğunu göstermiştir. Wurll ve Henrich (2011), endüstriyel robotlar için yol planlama alanında katkı sunar. A\* araması ve kesikli yapılandırma alanı (Discretized C-space) metodları kullanılmıştır. Zhang vd. (2006), çalışmasında kamera ve kuvvet sensörü kullanarak çapak alma için yörünge planlaması yapmıştır. Öncelikle manipülatörün izleyeceği yol işaretleniyor. Daha sonra kamera ile özellik çıkarma işlemi yapılır. İstenen özellik ile çıkarılan özellik karşılaştırılır ve aradaki hataya göre manipülatörün XYZ koordinat sistemine göre yönelimi düzeltilir. Sonra manipülatör malzeme üzerinde işaretlemeyi takip edecek şekilde ilerletilirken kuvvet sensörü ile de Z eksen yönündeki değişimler belirlenir. Santos vd. (2008), en uygun kontrol teknikleri kullanarak kısıtlı çalışma alanında robot yol planlaması yapmıştır. Pareto en iyi ağırlık hedefi metodu kullanılmıştır. Ayrıca her eklemin engele olan uzaklığı ölçülür ve bu her adımda güncellenir. Böylece robot engellere çarpmaz. Ramabalan vd. (2009), engellerin varlığında robot kol mekanizmalarında çok amaçlı dinamik yörünge planlamasında çalışmıştır. Çalışmada, iki çok amaçlı evrimsel algoritma yani seçkin olmayan egemen sıralama genetik algoritma (NSGA - II) ve çok amaçlı diferansiyel evrim (MOD) algoritmaları kullanılmıştır. Zheng vd. (2009), endüstriyel mekatronik sistem için çevrimiçi pürüzsüz yörünge ortaya konulmuştur. (İki modifiye doğrusal olmayan izleme diferansiyel sınırlı hız ve ivme ile kaba bir referans (adım ve rampa, vb.) düzgün yörünge oluşturmak için önerilmiştir). Graaf vd. (2010), bu çalışmasında lazer sensör kullanarak gerçek zamanlı bağlantı noktası takibi yapmıştır. Robotun uç noktasına bağlı lazer sensör belli bir görüş açısına sahip ve bağlantı noktası takibinin yapılabilmesi için bağlantı noktasının sensörün görüş açısının içinde kalması gereklidir. Bunun için lazer sensörden gelen veriler işlenerek robotun hareketi için gerekli veriler elde edilir ve bağlantı noktası

lazer sensörün görüş açısında kalır ve takip yapılmış olur. Guigue vd. (2010), çok amaçlı robot yörünge planlama sorunlarını çözmek için yeni bir yaklaşım sunmaktadır. Pareto metodu en iyi seti yerine tek bir çözüm ağırlık amaç fonksiyonları bulmak için önermektedir. Ayrık Dinamik Programlama (DDP) yaklaşım yöntemi ise pareto optimum seti temsil etmek için umut verici bir yaklaşım olarak gösterilmektedir. Kröger vd. (2010), beklenmedik ani olaylar için çevrimiçi yörünge üretimi konusunda yöntem önermiştir. Yönteme göre hareket yörüngeleri her milisaniye, her kontrol döngüsü içinde çevrimiçi üretilmektedir. Herhangi bir durumda beklenmedik olaylara anında tepki sağlamaktadır.. Sonuç olarak robotik sensör entegrasyonu özellikle kontrol sistemlerinin geliştirilmesi büyük ölçüde basitleştirilmektedir. Bonilla vd. (2012), kalibresiz görüş kullanarak endüstriyel robot manipülatör ve empedans kontrolü ile yol izleme manevraları yapmıştır. Yörünge veri tabanında kayıtlıdır sonra kameradan gelen veri ile karşılaştırılarak oryantasyon hatası giderilir. Daha sonra robot belirlenen yörüngede ilerlerken kuvvet sensörü ile kontrol yapılarak robotun zarar görmesi önlenir. Vakanski vd. (2012), öğrenme ve yeniden gösterim ile robot programlama için karmaşık yörüngelerin çözülmesi için etkili bir yöntem geliştirmektedir. Kodlama gizli Markov modeli ile yapılmaktadır. Jetchev ve Toussaint (2013), yerel hareket planlaması yöntemini geliştirmek için yöntem açıklamışlardır. Ana fikir yeni durumlar için uygun yörüngeleri hızlı bir şekilde tahmin etmektir. Yöntemin iki temel maddesi vardır. Birincisi önceki durumlardan yeni duruma genellemek gereklidir. Bunun için durumlar tanımlanmalıdır. Bu durumlar için özellikler oluşturmak ve genelleme geliştirmek için düzgünleştirilmiş özellik seçimi yaklaşımı kullanılmaktadır. İkinci madde ise önceki optimize yörüngelerin yeni duruma transferi eklem açısı uzayında yapılmamalıdır. Gasparetto vd. (2012), yörünge planlaması konusunda yapılmış çalışmaların derlemesini yapmıştır.

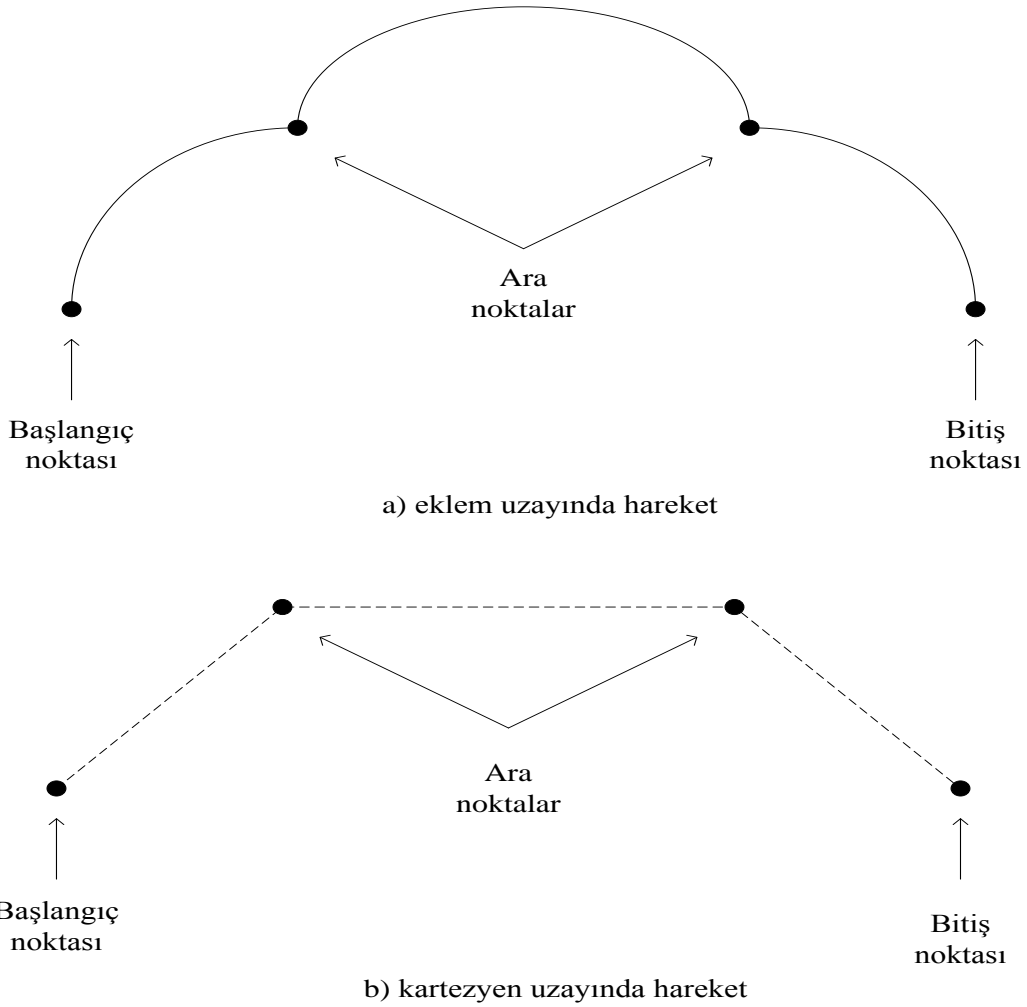
### ***3 Boyutlu görüntü oluşturma ile ilgili çalışmalar.***

Tognolo vd. (2003), biyomedikal uygulamalar için anatomik parçaların 3 boyutlu modelini elde etmeyi amaçlamıştır. Nesne yüzeyinden nokta bulutu elde etme ile yeni yüzey kurma algoritması 3 boyutlu modeli elde etmek için birlikte kullanılmıştır. Prototip 3 boyutlu tarama sistemi sunulmuştur. Lanman vd. (2009), tek ortografik yapısal ışık projektörü, düzlemsel aynalar ve senkronize kameralar kullanılarak 3 boyutlu yüzey modelleri elde etmek için yeni bir sistem sunmaktadır. Tasarımın temel yararı nesne

yüzeyinde her noktanın birden fazla yandan aydınlatılmış olma olasılığı olmasına rağmen kesin bir “gray code” dizisi atanabilmesini sağlamaktadır. Niola vd. (2010), lazer tarayıcı ve webcam ile gerçek zamanlı şekil edinimi için yöntem sunmuştur. Lazer görüntü hattı saptanması için bir dijital filtre parametreleri tanımlaması incelenmiştir. Yöntemin aynı zamanda robot, kinematik kalibrasyon gibi robot uygulamaları için kullanılabilir olduğunu sunmaktadır. Chan vd. (2001), sinir ağı tabanlı stereo görüntü işleme CMM (koordine ölçüm makinesi) çalışma alanı nesneyi bulmak için ve CMM tarama sistemini oluşturmak için kullanılmıştır. CMM tarama sisteminden elde edilen verilerle daha sonra bir CAD/CAM modeli inşa edilebilmektedir. Torabi ve Gupta (2010), 3 boyutlu nesne modelleme için entegre ve tam özerk “eye-in hand” (göz-elde) sistemini sunmaktadır. Sistem 6 serbestlik dereceli manipülatör kolunun üzerine monte edilmiş bir lazer menzil sensöründen oluşmaktadır. Sistem bilinmeyen bir ortamda bir nesnenin 3 boyutlu modelini oluşturmayı göstermektedir. Khalfaoi vd. (2013), nesne şekli ile ilgili asgari ön bilgiler ile tam ve otomatik tarama stratejisi geliştirmeyi hedeflemiştir. 3 boyutlu modelleri yeniden oluşturmak için NBV (sonraki eniyi görüntü) algoritmasını belirlemek üzere yeni bir yaklaşım sunmaktadır. Metod yüzeylerin sınıflandırılmasına dayanmaktadır.

### 3. YÖRÜNGE PLANLAMASI

Robot kollarının uç işlevcisinin bir başlangıç noktasından bitiş noktasına hareketi esnasında hangi ara noktalardan, hangi hız ve ivme ile hareket edeceği ve o noktalardaki yöneliminin hesaplanmasına yörünge planlaması denilmektedir. Robot kolları için eklem uzayında ve kartezyen uzayında olmak üzere iki farklı yörünge planlaması yapılmaktadır.



**Şekil 3.1.** Uç işlevcisinin a) eklem uzayında b) kartezyen uzayında takip ettiği yörüngeyi başlangıç, bitiş ve ara noktaları.

Şekil 3.1’de görüldüğü gibi, Kartezyen ve eklem uzayında iki nokta arasında hareket planlaması yapıldığında uç işlevcinin hareketi birbirinden farklıdır. Şekil 3.1(a)’da eklem uzayında yörünge planlaması sonucu uç işlevcin hareketi görülmektedir. Bu hareket aynı zamanda noktadan noktaya hareket olarak da adlandırılmaktadır. Robotun uç işlevcisinin, başlangıçta bulunduğu ve ulaşması istenen noktalardaki eklem açıları belirlenmektedir. Her bir eklemin eşit zamanda aralığında hedef pozisyona ulaşması için eklem hareketleri planlanmaktadır.

Şekil 3.1(b)’de ise, Kartezyen uzayda hareket planlaması sonucu uç işlevcinin hareketi görülmektedir. Burada, uç işlevcinin başlangıç ve hedef pozisyonları arasında doğrusal hareketi sağlanmaktadır. Robot uç işlevci için tanımlanan doğrusal yol üzerinde, birbirine yeterince yakın ara noktalar tanımlanmakta ve robot uç işlevcisinin bu ara noktalardaki pozisyon ve hızları belirlenmektedir. Böylece, eklemler bu hız profilini sağlayacak şekilde sürülmektedir.

### 3.1. Eklem Uzayında Yörünge Planlaması

Bu yörünge planlamasında, genellikle en kolay hesaplama söz konusudur. Robot üzerindeki tekil noktaları problem teşkil etmez. Planlama için bazı düzgün fonksiyonlar kullanılmaktadır.

#### 3.1.1. Kübik polinomlar ile yörünge planlaması

Zamana bağlı kübik bir yörünge oluşturan üçüncü dereceden polinom olarak;

$$\theta(t) = s_0 + s_1t + s_2t^2 + s_3t^3 \quad (3.1)$$

denkleminde ifade edilir. Eklem hızı (3.2)’deki gibi  $\theta(t)$  denkleminin birinci türevi, ivmesi (3.3) ise  $\theta(t)$  denkleminin ikinci türevinin alınması ile elde edilir.

$$\dot{\theta}(t) = s_1 + 2s_2t + 3s_3t^2 \quad (3.2)$$

$$\ddot{\theta}(t) = 2s_2 + 6s_3t \quad (3.3)$$

$$\begin{aligned} \theta(0) &= \theta_0 & : \text{Başlangıç Açısı} \\ \theta(t_f) &= \theta_f & : \text{Hedef Açısı} \\ \dot{\theta}(0) &= 0 & : \text{Başlangıç Hızı} \\ \dot{\theta}(t_f) &= 0 & : \text{Bitiş Hızı} \end{aligned} \quad (3.4)$$

(3.4)'deki bilinen koşullar, (3.1) ve (3.2)'deki denklemler kullanılarak zaman katsayıları (3.5)'deki gibi bulunur ve bu katsayılar başlangıç ve bitiş hızlarının sıfır olduğu durumlarda geçerlidir.  $\theta(t)$ ,  $\dot{\theta}(t)$  ve  $\ddot{\theta}(t)$  denklemlerine ait grafikler şekil 3.2 gösterilmiştir.

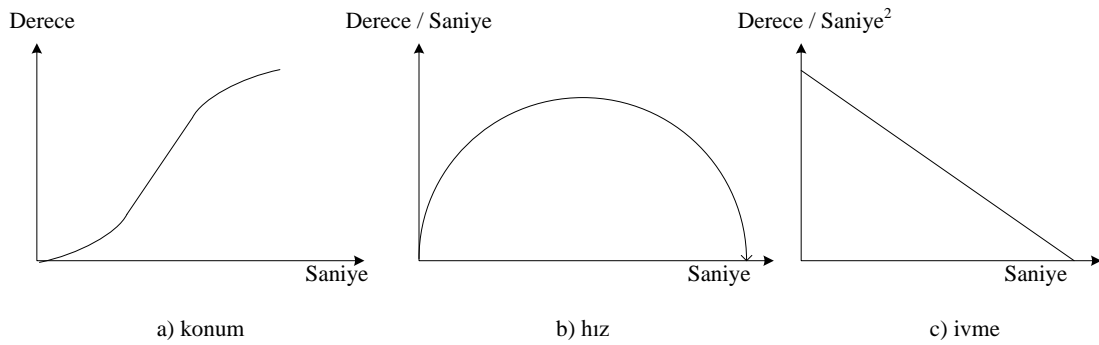
$$s_0 = \theta_0$$

$$s_1 = 0$$

$$(3.5)$$

$$s_2 = \frac{3}{t_f^2} (\theta_f - \theta_0)$$

$$s_3 = -\frac{2}{t_f^3} (\theta_f - \theta_0)$$



**Şekil 3.2.** Üçüncü dereceden kübik bir yörüngeye ait a) konum, b) hız, c) ivme grafiği

### 3.1.2. Kübik polinomlar ile ara noktalar için yörünge planlaması

Robot başlangıç noktasından hedef noktasına giderken bir takım ara noktalardan geçer. Bu noktalarda başlangıç ve bitiş hızları sıfır değildir ve (3.5)'deki denklem katsayıları geçerli değildir. (3.6)'daki yeni bilinenler ile katsayılar yeniden hesaplanmalıdır.

$$\begin{aligned}
 \theta(0) &= \theta_0 & : \text{Başlangıç Açısı} \\
 \theta(t_f) &= \theta_f & : \text{Hedef Açısı} \\
 \dot{\theta}(0) &= \dot{\theta}_0 & : \text{Başlangıç Hızı} \\
 \dot{\theta}(t_f) &= \dot{\theta}_f & : \text{Bitiş Hızı}
 \end{aligned} \tag{3.6}$$

(3.1)'deki denklem (3.6)'daki bilinenler ile çözüldüğünde (3.7)'deki katsayılar elde edilir.

$$\begin{aligned}
 s_0 &= \theta_0 \\
 s_1 &= \dot{\theta}_0 \\
 s_2 &= \frac{3}{t_f^2} (\theta_f - \theta_0) - \frac{2}{t_f} \dot{\theta}_0 - \frac{1}{t_f} \dot{\theta}_f \\
 s_3 &= -\frac{2}{t_f^3} (\theta_f - \theta_0) + \frac{1}{t_f^2} (\dot{\theta}_f - \dot{\theta}_0)
 \end{aligned} \tag{3.7}$$

### 3.1.3. Yüksek dereceden polinomlar ile yörünge planlaması

Robotun takip edeceği yörünge hem başlangıç hem de bitişindeki eklem konum, hız ve ivmelerinin baştan belirlenmesi istenirse (3.8)'deki 5.dereceden bir polinom kullanılır.

$$\theta(t) = s_0 + s_1 t + s_2 t^2 + s_3 t^3 + s_4 t^4 + s_5 t^5 \tag{3.8}$$



$$s_0 = \theta_0$$

$$s_1 = \dot{\theta}_0$$

$$s_2 = \frac{\ddot{\theta}_0}{2}$$

(3.9)

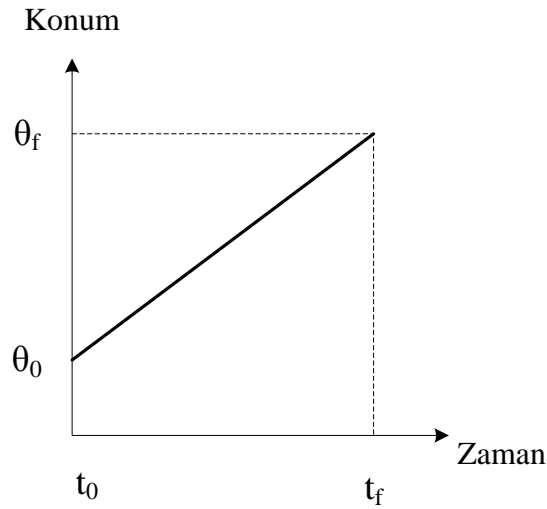
$$s_3 = \frac{20(\theta_f - \theta_0) - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f + (\ddot{\theta}_f - 3\ddot{\theta}_0)t_f^2}{2t_f^3}$$

$$s_4 = \frac{30(\theta_0 - \theta_f) + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4}$$

$$s_5 = \frac{12(\theta_f - \theta_0) + 6(\dot{\theta}_f + \dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5}$$

Başlangıç ve bitiş açılarına göre katsayılar (3.9)'daki gibi bulunur.

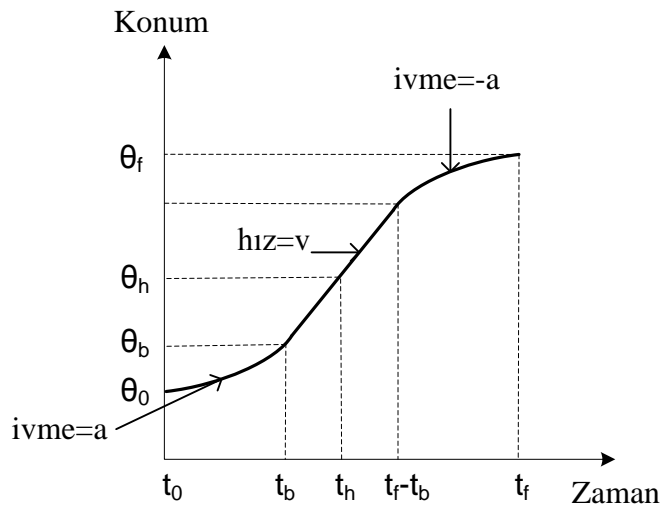
### 3.1.4. Parabolik eklemeler ile doğrusal fonksiyon



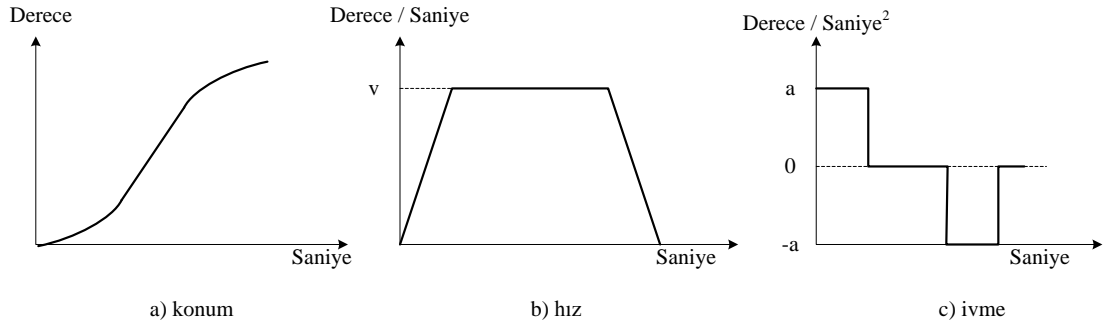
Şekil 3.3. Doğrusal Yörünge.

Diğer bir yörünge yöntemi parabolik kısımların yerleştirildiği doğrusal yörünge dir. Bu yöntemde eklemelerin tamamı doğrusal bir yörünge takip etse dahi, uç işlevci iki nokta arasında doğrusal geçemez. Bu durumda hareketin başlangıcında ve bitişinde sürekli olmayan hız problemleriyle karşılaşılır. Bunu önlemek için yörünge nin başlangıç ve bitişine parabolik kısımlar eklenerek konum ve hızın sürekliliği sağlanır. Ayrıca, parabolik bir yörünge de sabit ivmeli bir hareket kullanılarak hızın sabit pürüzsüz bir şekilde sürekli olması sağlanır.

Şekil 3.4' de parabolik kısımlar eklenmiş doğrusal yörünge görülmektedir ve  $t_0 - t_b$  ve  $t_f - t_b$  ile  $t_f$  arası doğrusal yörünge nin başlangıcına ve bitişine eklenmiş parabolik kısımları  $t_b$  ile  $t_f - t_b$  arası ise doğrusal bölgeyi göstermektedir.  $t_h$  ile gösterilen noktanın sağ ve sol tarafında kalan bölümler simetri olacak şekilde tasarlanır. Yani  $t_h = t_f/2$  ve  $\theta_h = (\theta_0 + \theta_f)/2$ 'dir. Bunun için parabolik kısımlarda gerçekleşen zamanların bir birine eşit olmasının yanında ivmelerde sabit ve aynı olmalıdır. Aynı zamanda, sürekli bir hız elde edebilmek için parabolik kısımların sonundaki hızla doğrusal bölümdeki hızın bir birine eşit olması gerekir. Şekil 3.5 de ise konum, hız ve ivme grafikleri gösterilmektedir.



Şekil 3.4. Parabolik kısımlar eklenmiş doğrusal yörünge.



**Şekil 3.5.** Parabolik kısımlar eklenmiş doğrusal yörüngeye ait a) konum, b) hız, c) ivme grafikleri.

### 3.2. Kartezyen Uzayında Yörünge Planlaması

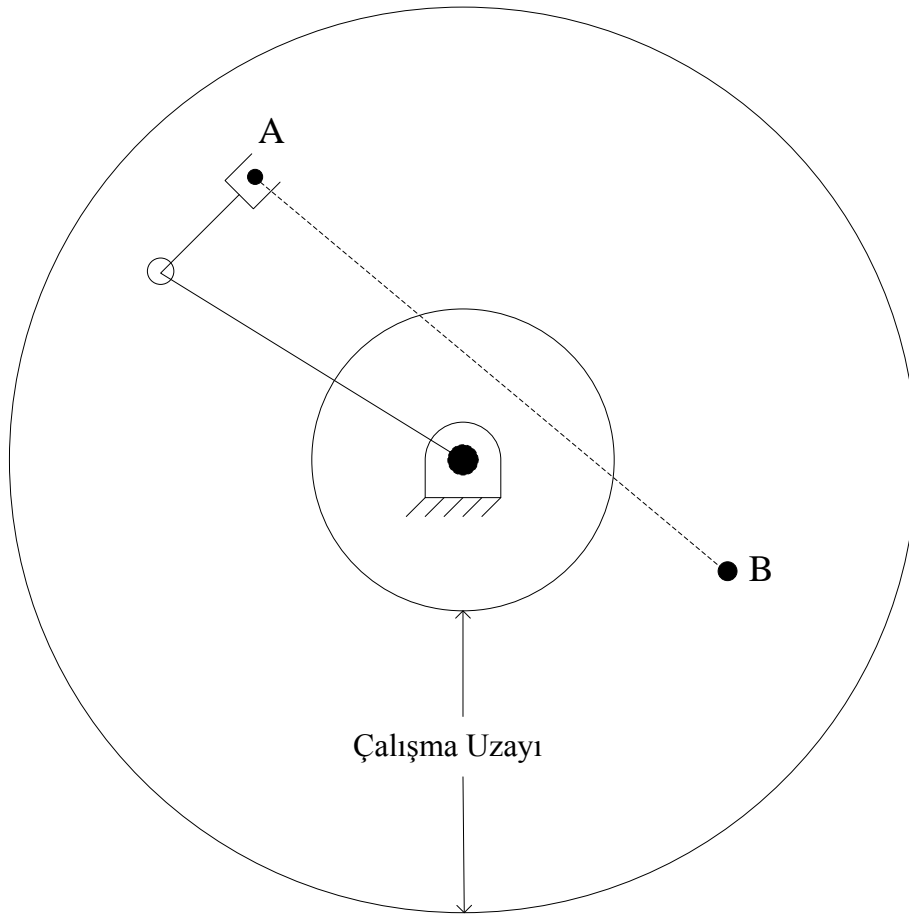
Yörünge planlaması eklem uzayında yapılabildiği gibi kartezyen uzayında da yapılabilmektedir. Uygulanan metot ise yörünge şeklini kartezyen konumu ve yönelimine bağlı fonksiyon şeklinde tanımlamaktır. Bu konum ve yönelimler robotun seçilen ana çerçevesine göre olmalıdır. Ters kinematik yöntemiyle bu konum ve yönelimler eklem açılarına çevrilmektedir ve bu açıların elde edildiği sıra ile uygulanmasıyla istenen kartezyen yörünge planlaması elde edilir. Kartezyen yörünge üzerinden seçilen konum bilgilerinin çokluğu istenen yörünge şeklinin daha gerçekçi olmasını sağlamaktadır. Kullanılan yörünge şekillerine örnek düz çizgi, dairesel ve sinüzoidal verilebilir. En çok tercih edilen ise düz çizgidir.

#### Kartezyen yörüngede geometrik sorunlar

Kartezyen yörüngeleri çalışma alanı ve tekillik ile ilgili çeşitli sorunlara yatkındır.

### Tip 1 Sorunlar : Erişilemeyen ara noktalar

Robot kolun ilk konumu ve hedef konumu çalışma alanı içinde olmasına rağmen, bu iki noktayı birleştiren düz çizgi üzerindeki tüm noktalar çalışma alanı içinde olmayabilir.

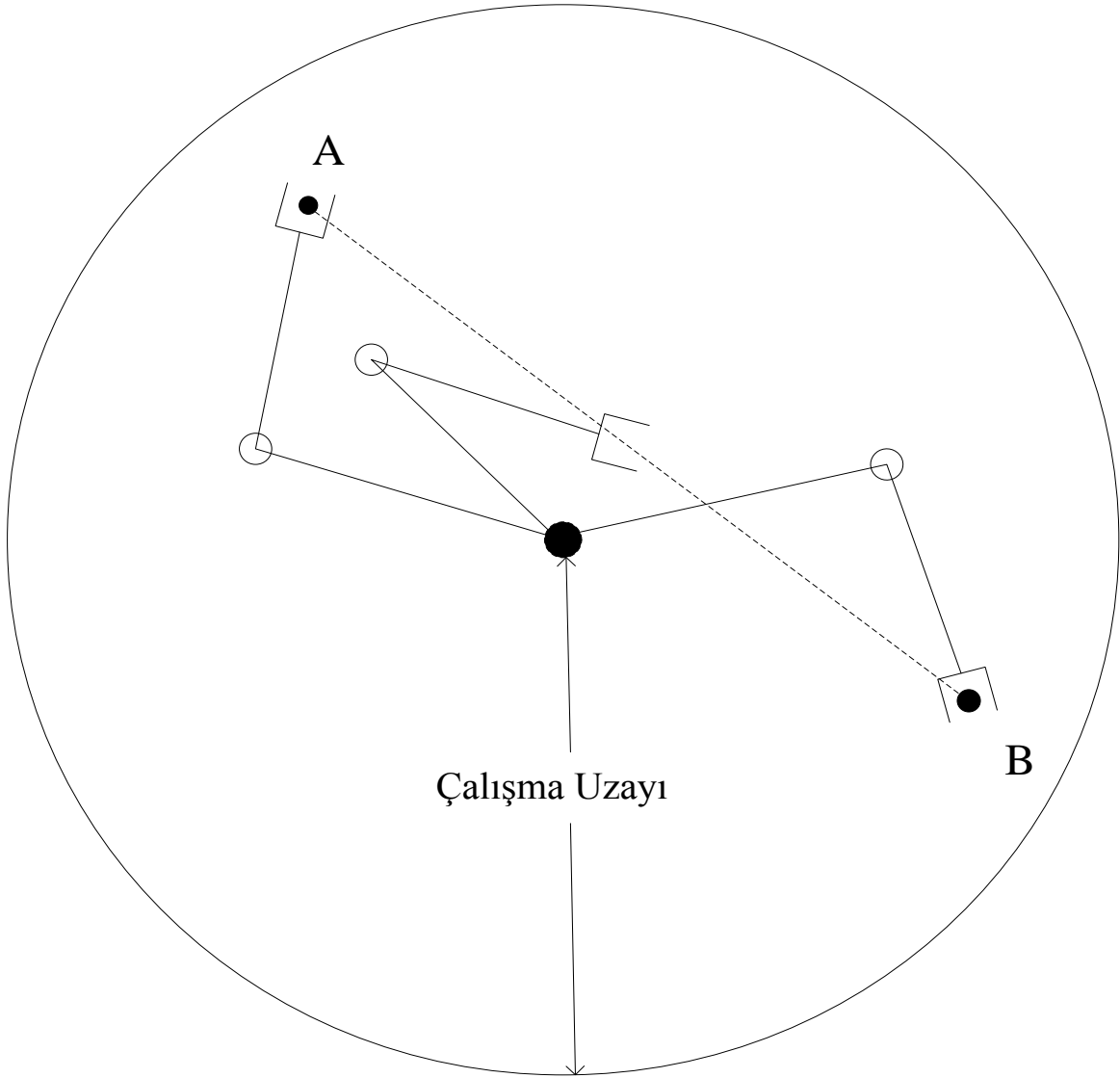


**Şekil 3.6.** Kartezyen yolu sorunu tip 1.

Şekil 3.6 de görüldüğü gibi 2 kol birbirinden farklı uzunluktadır ve çalışma uzayı kolların farkı olan bir delik içerir. Böylece A noktasından B noktasına giderken eklem uzayında sorun yoktur. Fakat kartezyen uzayında manipülâtörün ulaşamayacağı ara noktalar vardır.

### Tip 2 Sorunlar : Tekillik Yakınında Yüksek Eklem Oranları

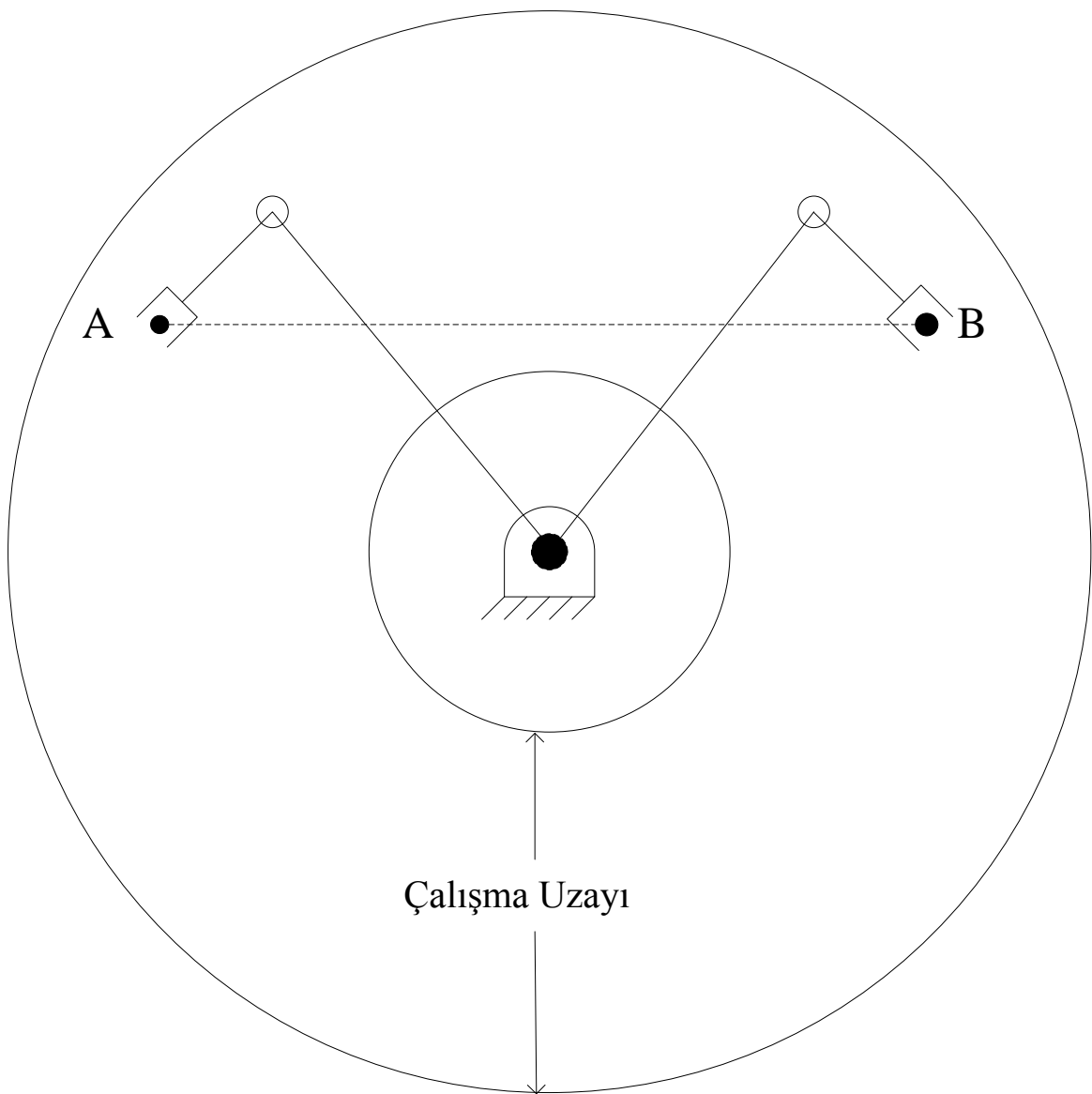
Kartezyen uzayda uç işlevci için istenen hızlar bazen seçilememektedir. Bu manipülâtörün çalışma alanında seçilemeyen sonlu eklem oranlarından kaynaklanmaktadır. Örnek olarak Şekil 3.7'a bakıldığında eşit kol uzunluklu manipülâtör A noktasından B noktasına gitmektedir. İstenen yörünge boyunca düz çizgide sabit doğrusal hız ile uç işlevci hareket etmektedir. Yol boyunca tüm noktalar ulaşılabilir fakat robot yolun orta noktasını geçerken eklem hızları çok yüksektir.



Şekil 3.7. Kartezyen yolu sorunu tip 2.

**Tip 3 Sorunlar: Başlangıç ve hedef noktalarına farklı çözümler ile ulaşılabilir**

Bu sorun robot hedef noktasına, başlangıç noktasındaki fiziksel şekli ile ulaşamadığında oluşur. Şekil 3.8 bu durum için bir örnek verilmektedir. Resimde robot bütün noktalara ulaşabilmektedir fakat tek bir çözüm olarak değil birkaç çözüm şeklinde bu mümkün olabilmektedir. Bu problemten dolayı bir çok endüstriyel robot yörünge planlaması için hem eklem hem kartezyen uzayını destekler.



**Şekil 3.8.** Kartezyen yolu sorunu tip 3.

## 4. SİSTEM TASARIMI

Sistem tasarımı 3 kısımdan oluşmaktadır. Bu kısımlar, endüstriyel robot kol tasarımı, benzetim ortamı tasarımı ve denetim yazılım tasarımından oluşmaktadır. *Endüstriyel robot tasarımı*, Robotes Robotik Çözümler Ltd. Şti. tarafından gerçekleştirilmiştir. Bu tez kapsamında, kinematik ve dinamik denklemlerin Matlab ortamında ve C/C++ programlama dili kullanılarak kodlamaları ve testleri yapılmıştır. *Benzetim ortamı tasarımında*, robotun Gazebo olarak adlandırılan açık kaynak kodlu benzetim ortamı oluşturma yazılımında 3B benzetimi gerçekleştirilmiştir. Bunun için, XML tabanlı özel bir formata sahip dosya olarak, robotun fiziksel özelliklerinin tanıtımı gerçekleştirilmiştir. Ayrıca, benzetim yazılımının komutları alması ve durum bilgilerini iletmesini sağlayacak eklenti (plug-in) yazılmıştır. *Denetim yazılım tasarımında* ise, cisimlerin 3B modelinin robot kolu uç işlevcisine bağlanmış lazer profil algılayıcı ile yapılabilmesi için yazılım tasarımı gerçekleştirilmektedir. Bu kısımda, hareket planlaması ve kodlaması gerçekleştirilmektedir. Bu kısım, yörünge planlaması ve kodlamasını içermekte ve Bölüm 5.1’de detaylı olarak verilmektedir.

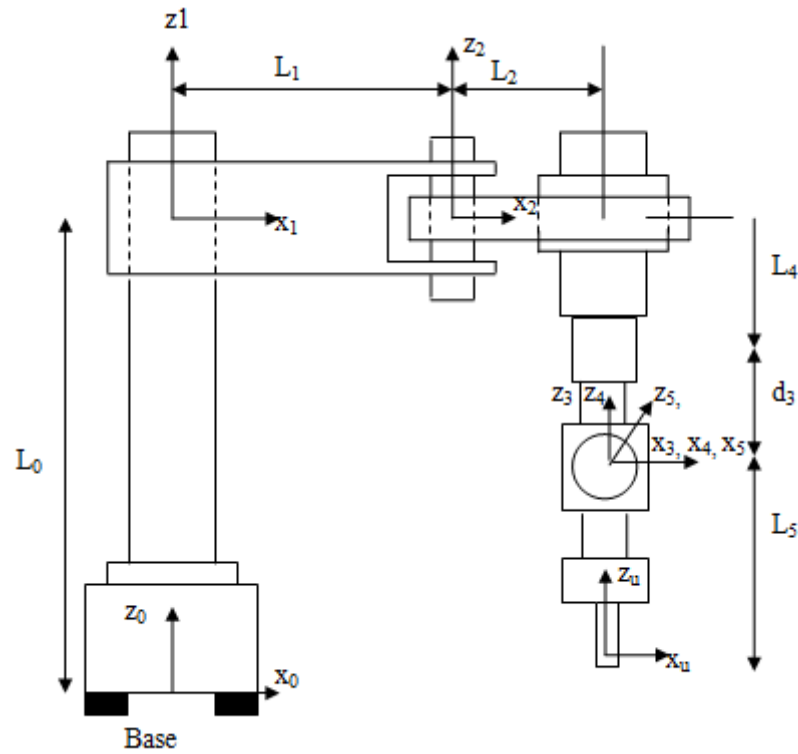
### 4.1. Endüstriyel Robot Kol Tasarımı

Endüstriyel robot kolun mekanik ve elektronik tasarımları, Robotes Robotik Çözümler Ltd.Şti. tarafından yapılmıştır. Robotun teknik çizimleri ve parametreleri, tez kapsamında kinematik denklemlerin elde edilmesi ve kodlanması, yörünge planlamaları ve benzetim modelinin oluşturulmasında kullanılmıştır.

#### 4.1.1. İleri kinematik

İleri kinematik, eklem değişkenleri bilindiğinde takım ucunun pozisyon ve yönlenmesinin referans koordinat sistemine göre belirlenmesi için gerekli denklemleri kapsamaktadır.

Tez kapsamında, çalışmaların yürütüldüğü endüstriyel robot kolun parametreleri Şekil 4.1’de verilmektedir. Burada  $L_i$ ’ler ilgili linklerin uzunluklarını temsil etmektedir.



Şekil 4.1. Robot Kol Parametreleri

Çizelge 4.1. Denavit-Hartenberg (DH) Parametreleri

$\dot{i}$	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	$L_0$	$\theta_1$
2	0	$L_1$	0	$\theta_2$
3	0	$L_2$	$-(d_3 + L_4)$	0
4	0	0	0	$\theta_4$
5	$-90^\circ$	0	0	$\theta_5$
6	$90^\circ$	0	$-L_5$	0



Referans koordinat sistemi (Base)  $\theta_0$  sıfır olduğunda ilk eklemdaki koordinat sistemi ile çakışık şekilde seçildiğinde sistemin Denavit-Hartenberg parametreleri Çizelge 4.1'deki gibi olacaktır.

Kinematik denklemleri türetirken 4x4 boyutlu homojen dönüşüm matrislerinden faydalanılmaktadır. Bu matrislerin genel yapısı;

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & \alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

şeklinde ve i. ekleme iliştilen koordinat sisteminin i-1. ekleme iliştilen koordinat sistemine göre pozisyon ve yönlenmesini belirtmektedir. Burada  $c\theta_i$  ve  $s\theta_i$  sırasıyla ilgili açının kosinüs ve sinüsüdür. Verilen D-H parametreleri ve homojen matris formülüne göre aşağıdaki matrisler yazılabilir;

$$T_1^0 = \begin{pmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & L_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.2)$$

$$T_2^1 = \begin{pmatrix} c\theta_2 & -s\theta_2 & 0 & L_1 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.3)$$

$$T_3^2 = \begin{pmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -(d_3 + L_4) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.4)$$

$$T_4^3 = \begin{pmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.5)$$

$$T_5^4 = \begin{pmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.6)$$

$$T_6^5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & L_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.7)$$

Uç işlevcinin referans koordinat sistemine göre pozisyon ve yönlenmesini bulmak için yukarıdaki altı matris aşağıdaki şekilde çarpılmalıdır;

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 = \begin{pmatrix} c_5 c_{124} & -s_{124} & s_5 c_{124} & L_1 c_1 + L_2 c_{12} - L_5 s_5 c_{124} \\ c_5 s_{124} & c_{124} & s_5 s_{124} & L_1 s_1 + L_2 s_{12} - L_5 s_5 s_{124} \\ -s_5 & 0 & c_5 & L_0 - L_4 - d_3 - L_5 c_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.8)$$

Burada  $c_{12} = \cos(\theta_1 + \theta_2)$  tür. Diğer değişkenlerde benzer şekilde tanımlanmaktadır.

#### 4.1.2. Ters kinematik

Ters kinematik ise, takım ucunun bağlandığı bilek noktasının istenilen pozisyon ve yönlenmesini sağlayacak eklem değişkenlerinin belirlenmesidir. Bu noktanın pozisyon ve yönlenmesi

$$T_u^0 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

olarak verilsin. Denklem 4.8 ve 4.9'da verilen matrislerin elemanları eşleştirilerek eklem açıları bulunmaktadır.

### $\theta_5$ açısının bulunması

$T_6^0$  matrisinin 3.satır, 1.sütunundaki eleman ve 3.satır, 3.sütundaki elemanlardan  $s_5$  ve  $-c_5$  bilinmektedir. İkisinin oranından  $\tan(\theta_5)$  bulunur.

$$\theta_5 = \tan^{-1} - \frac{T_6^0(3,1)}{T_6^0(3,3)} \quad (4.10)$$

$$d_3 = L_0 - L_4 - (L_5 - \cos\theta_5) - P_z \quad (4.11)$$

$\theta_5$  bilindiğinden formül (4.11)'e göre  $d_3$  uzunluğu bulunur.  $T_6^0$  matrisinin 1.satır, 2.sütunundaki eleman ve 2.satır, 2.sütundaki elemanlardan  $-s_{124}$  ve  $c_{124}$  bilinmektedir. İkisinin oranından  $\tan(\theta_{124})$  bulunur.

$$\theta_{124} = \tan^{-1} - \frac{T_6^0(1,2)}{T_6^0(2,2)} \quad (4.12)$$

### $\theta_2$ açısının bulunması

(4.13) ve (4.14) düzenlenerek (4.15) ve (4.16) elde edilir. Eşitsizliğin çözümü (4.17), (4.18) ve (4.19)'daki gibi çözümlenerek  $\theta_2$ 'nin kosinüs değeri (4.20)'de, sinüs değeri ise (4.21) de bulunur. (4.22) ve (4.23)'e göre de  $\theta_2$ 'nin iki farklı çözümü hesaplanır.  $\theta_i^j$ 'de i. eklem açısının j. çözümünü ifade eder. Örneğin  $\theta_2$ 'nin iki çözümü vardır. Bunlar  $\theta_2^1$  ve  $\theta_2^2$ 'dir.

$$P_x = L_1 c_1 + L_2 c_{12} - L_5 s_5 c_{124} \quad (4.13)$$

$$P_y = L_1 s_1 + L_2 s_{12} - L_5 s_5 s_{124} \quad (4.14)$$

$$P_x + L_5 s_5 c_{124} = A = L_1 c_1 + L_2 c_{12} \quad (4.15)$$

$$P_y + L_5 s_5 s_{124} = B = L_1 s_1 + L_2 s_{12} \quad (4.16)$$

$$A^2 = L_1^2 c_1^2 + L_2^2 c_{12}^2 + 2L_1 L_2 c_1 c_{12} \quad (4.17)$$

$$B^2 = L_1^1 s_1^2 + L_2^2 s_{12}^2 + 2L_1 L_2 s_1 s_{12} \quad (4.18)$$

$$A^2 + B^2 = L_1^2 + L_2^2 + 2L_1 L_2 c_2 \quad (4.19)$$

$$c_2 = \frac{A^2 + B^2 - L_1^2 - L_2^2}{2L_1 L_2} \quad (4.20)$$

$$s_2 = \mp \sqrt{1 - c_2} \quad (4.21)$$

$$\theta_2^1 = \text{Atan2}(s_{21}, c_2) \quad (4.22)$$

$$\theta_2^2 = \text{Atan2}(s_{22}, c_2) \quad (4.23)$$

### $\theta_1$ açısının bulunması

(4.24) ve (4.25)'deki eşitsizlik, (4.26), (4.27) ve (4.28)'deki düzenlenen eşitsizlikler de kullanılarak (4.29) ve (4.30) da  $\theta_1$ 'in iki farklı çözümü bulunur.

$$A = L_1 c_1 + L_2 c_{12} = L_1 c_1 + L_2 c_1 c_2 - L_2 s_1 s_2 = c_1(L_1 + L_2 c_2) - s_1(L_2 s_2) \quad (4.24)$$

$$B = L_1 s_1 + L_2 s_{12} = L_1 s_1 + L_2 s_1 c_2 - L_2 c_1 s_2 = s_1(L_1 + L_2 c_2) - c_1(L_2 s_2) \quad (4.25)$$

$$k_1 = L_1 + L_2 c_2 \quad (4.26)$$

$$k_{21} = L_2 s_{21} \quad (4.27)$$

$$k_{22} = L_2 s_{22} \quad (4.28)$$

$$\theta_1^1 = \text{Atan2}(B, A) - \text{Atan2}(k_{21}, k_1) \quad (4.29)$$

$$\theta_1^2 = \text{Atan2}(B, A) - \text{Atan2}(k_{22}, k_1) \quad (4.30)$$

### $\theta_4$ açısının bulunması

Bilinen değerleri ile (4.31) ve (4.32)'deki formüller ile  $\theta_4$ 'ün iki farklı çözümü bulunur.

$$\theta_4^1 = (\theta_{124} - \theta_{11} - \theta_{21}) \quad (4.31)$$

$$\theta_4^2 = (\theta_{124} - \theta_{12} - \theta_{22}) \quad (4.32)$$

## 4.2. Benzetim Ortamı Tasarımı

Benzetim ortamı, Robot İşletim Sistemi (**Robot Operating System** – ROS (Anonim3,2013)) ve **Gazebo** (Anonim5,2015) programları kullanılarak gerçekleştirilmiştir. Robotun benzetim modeli, “**Unified Robot Description Format (URDF)**” olarak tanımlanan bir formata göre hazırlanmıştır. URDF, XML formatında robot modellerinin tanımlanmasında kullanılan bir formattır. Gazebo programı ise, 3B benzetim ortamında URDF formatında tanımlanmış robotun benzetimini gerçekleştirmektedir. Model, , rviz (Anonim2,2014) adlı 3 boyutlu görselleştirme aracı ile çalıştırılabilmekte, robotun link ve eklemlerinin gerekli olan ölçü ve açı değerlerine uygunluğu kontrol edilebilmektedir.

Robot kolun hareket planlama ve kinematik denklemlerinin kodlamaları C++ programlama dili kullanılarak yapılmıştır. Bu hesaplamalarda bulunan matris işlemlerinde, Eigen (Anonim4,2015) olarak adlandırılan lineer cebir işlemlerini kapsayan C++ şablon kütüphanesi kullanılmıştır.

### 4.2.1. Robot işletim sistemi (ROS)

Stanford Yapay Zekâ laboratuvarlarında ilk olarak robot yazılım ara katman prototipi olarak geliştirilmiştir. 2007 yılında, Willow Garage olarak adlandırılan robotik uzman takımı tarafından resmi olarak sunulmuştur. Farklı robot ve donanım çeşitliliğinin yazılım geliştiricilere getirdiği zorlukları kolaylaştırmak için çeşitli yazılım kütüphanelerini içerecek şekilde geliştirilmiştir. Açık kaynak kodlu bir robot yazılım sistemi olan ROS, orijinal ismiyle Robot Operating System, yazılım geliştiricilere robot uygulamaları oluşturmak için kütüphaneler ve araçlar sağlar. Bu araçlar, donanım soyutlama, aygıt sürücüler, kütüphaneler, görüntüleyiciler, ileti aktarma, paket yönetimi şeklinde özetlenebilir. ROS, robot programları için standart formda mesajlaşmaya dayanan bir altyapı sunar. ROS ile entegre çalışacak ve mesajlaşma yapısını kullanacak robot uygulamaları geliştirildiğinde, bu uygulamalar tüm araştırmacılar tarafından kullanılabilir. Örneğin, gezgin robotların konumlandırılması için geliştirilen ROS

uyumlu bir uygulama, herhangi bir arařtırmacı tarafından kendi yazdıđı uygulamalarda kullanılabilir.

Robotik alanın birok kütüphane ve geliştirme aracı olmasına rağmen, ROS büyük ölçekli projeler geliştirilirken ortaya çıkan karmaşıklığı çözmek amacıyla tasarlanması ve modüler yapısıyla diğerlerinden ayrılarak řu an için benzersiz bir konumda bulunmaktadır.

ROS'un sağladığı faydalar 4 maddede sıralanabilir:

*Altyapı:* ROS hızlı ve kolay bir şekilde dağıtık bilgisayar sistemleri oluşturulabilmesini destekleyebilecek mesajlaşma altyapısını kullanmaktadır.

*Aralar:* ROS ayar, debug, görselleştirme, test etme gibi birçok hazır araç sunmaktadır.

*Yetenekleri:* ROS robot fonksiyonelliđi, hareket yönetimi gibi alanlarda kullanıcının işini kolaylařtıran kütüphaneler sağlamaktadır.

*Ekosistem:* ROS büyük bir grup tarafından desteklenip geliştirildiđi için güçlü bir dokümantasyonu ve yardım bölümleri mevcuttur. Modüler yapısıyla ihtiyaç duyulan program modülü daha önceden geliştiriciler tarafından paylaşılmış yüzlerce ROS paketi içinden aratılarak bulunabilir, bulunan paketin içinden gerekli modül alınarak çok basit bir şekilde başka programlara entegre edilebilir.

Açık kaynak kodlu olarak C++ dilinde geliştirilmiş olmasına rağmen Phyton dilinde de kod yazılabilmektedir. Resmi olarak ise Ubuntu desteklenmekle birlikte FEDORA ve MAC OS X gibi benzer sistemlerde de çalışmaktadır.

#### **4.2.2. Gazebo**

Gazebo, robot sistemleri için benzetim programıdır. ROS ile uyumludur. ROS altyapısını kullanan programlar ile mesajlaşarak, denetim programları tarafından yönetilebilir. Çeşitli robot tipleri, sensörler ve 3-boyutlu nesnelerin benzetimlerini yapılabilmektedir. Program, içerdđi dinamik hesaplama motoru sayesinde çarpışma ve eklem kuvveti uygulama gibi kütüphaneler ile geređe yakın benzetim gerçekleştirir.

Kullanıcı ister Gazebo'nun veri tabanından hazır modelleri seçebilir isterse de kendi modelini XML formatında tasarlayarak kullanabilmektedir.

Gazebo simülasyon programı içerdiği fizik, çarpışma, eklem çevirme gibi kütüphaneler ile dinamik simülasyonlara izin verir. Fizik soyutlama eylemi farklı fiziksel motorların Gazebo içerisinde kullanılmasını sağlar ve sonradan dahil edilebilecek bir eklenti ile herhangi bir fiziksel motor için birçok parametreye erişebilmeye imkân tanır.

Gazebo benzetim programının sağlam bir arayüzü vardır. Yeniden tasarlanmış grafik arayüzü kullanıcının, objeler üzerinden fiziksel değişiklik yapmasına ya da obje simülasyon özelliklerine, kısaca modellere erişimine geçit verir.

Gazebonun sahip olduğu özellikler şu şekilde listelenebilir:

*Dinamik Benzetim:* ODE, Bullet, Simbody ve DART gibi çok sayıda yüksek performanslı dinamik motorlara erişimi vardır.

*Gelişmiş 3-B Grafikler:* OGRE kütüphanesini kullanarak, yüksek kaliteli ışıklandırma, gölgeler ve dokuları kapsayan ortamların gerçekçi sahnelenmesini sağlar.

*Algılayıcılar ve Gürültü:* Lazer mesafe algılayıcı, 2B/3B kameralar, kuvvet-tork gibi algılayıcılardan, istenildiğinde gerçekçilik sağlamak üzere gürültü ekleyerek, algılayıcı verisi üretir.

*Eklentiler (Plugins):* Robot, algılayıcı ve ortam kontrolü için kişisel eklentiler geliştirilebilir. Eklentiler, Gazebo API'lerine direk erişim sağlar.

*Robot modelleri:* PR2, Pioneer 2DX gibi bir çok robot modelini kullanıma hazır olarak sağlar. Ya da XML formatında model tanımlaması yapılarak, kendi robot modelinizi de oluşturabilirler.

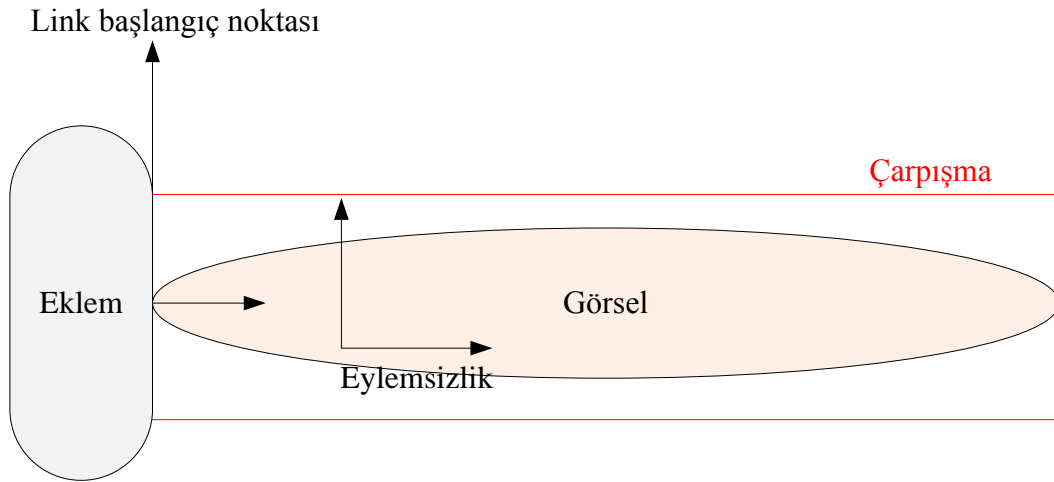
*TCP/IP ile İletme* Uzaktaki bir sunucuda çalıştırılan Gazebo programına, soket tabanlı mesaj iletimi ile ulaşılabilmektedir.

*Komut satırı araçları:* Geniş komut satırı araçları, benzetimin iç gözlem ve kontrolünü kolaylaştırır.

### 4.2.3. Gazebo ile kullanmak üzere robot modeli oluşturma

#### *Linklerin Modellemesi*

Link model, Şekil 4.2’de görüldüğü gibi 3 bölümden oluşmaktadır: Eylemsizlik (İnertia), görsel (visual) ve çarpışma (collision). Eylemsizlik, her bir link için tektir. Diğer bölümler birden fazla olabilir. Yani bir linkin görseli ya da çarpışmaya maruz kalacak yüzeyleri birden çok geometrik şekil ile tanımlanabilir. Eylemsizlikte linkin ağırlık merkezi ve eylemsizlik katsayıları belirtilir. Görselde linkin şekli (kare, çember vs.), rengi ve başlangıç konumu belirtilir. Çarpışmada ise linkin etrafında güvenli alan oluşturmayı sağlar. Robotun hareketi esnasında linkin bir linke veya cisme çarpmasını engeller. Burada da şekil tanımlaması yapılır ve o şeklin başlangıç konumu belirtilir. Orijinlerdeki XYZ’ler koordinat eksenindeki konumu, RPY’ler ise dönme eksenini belirtir. Ayrıntılı bilgi <http://wiki.ros.org/urdf/XML/link> adresinde bulunmaktadır. Şekil 4.3’de örnek link modeli görülmektedir. Şekil 4.4’de ise aynı linkin benzetim görüntüsü görülmektedir.



Şekil 4.2. Örnek Link Görüntüsü.



```

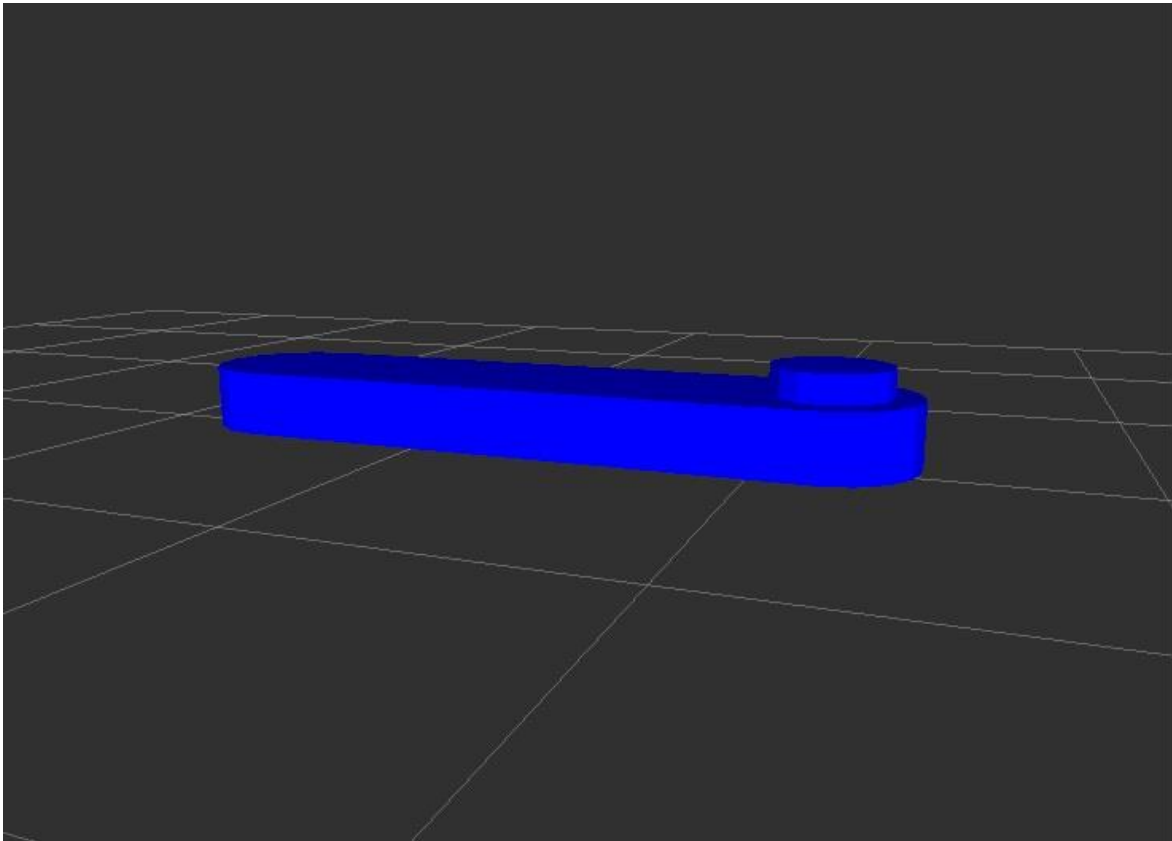
<link name="shoulder_pan"> <!--Link 1 -->
  <inertial>
    <origin xyz="0.2 0 0" rpy="0 -0 0"/> <!-- metre -->
    <mass value="20.81695"/> <!-- kilogram -->
    <inertia ixx="4986.47941134" ixy="2041.22277415" ixz="-0.80548565"
      iyy="1478.10939608" iyz="-2.10914985"
      izz="6403.32584988" /> <!-- metre-kare -->
  </inertial>

  <collision name="shoulder_pan_geom1">
    <origin xyz="0 0 0.025" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="0.065" length="0.050"/>
    </geometry>
  </collision>

  <visual name="shoulder_pan_geom1_visual">
    <origin xyz="0 0 0.025" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="0.065" length="0.050"/>
    </geometry>
    <material name="Cyan">
      <color rgba="0 1 1 1.0"/>
    </material>
  </visual>
</link>

```

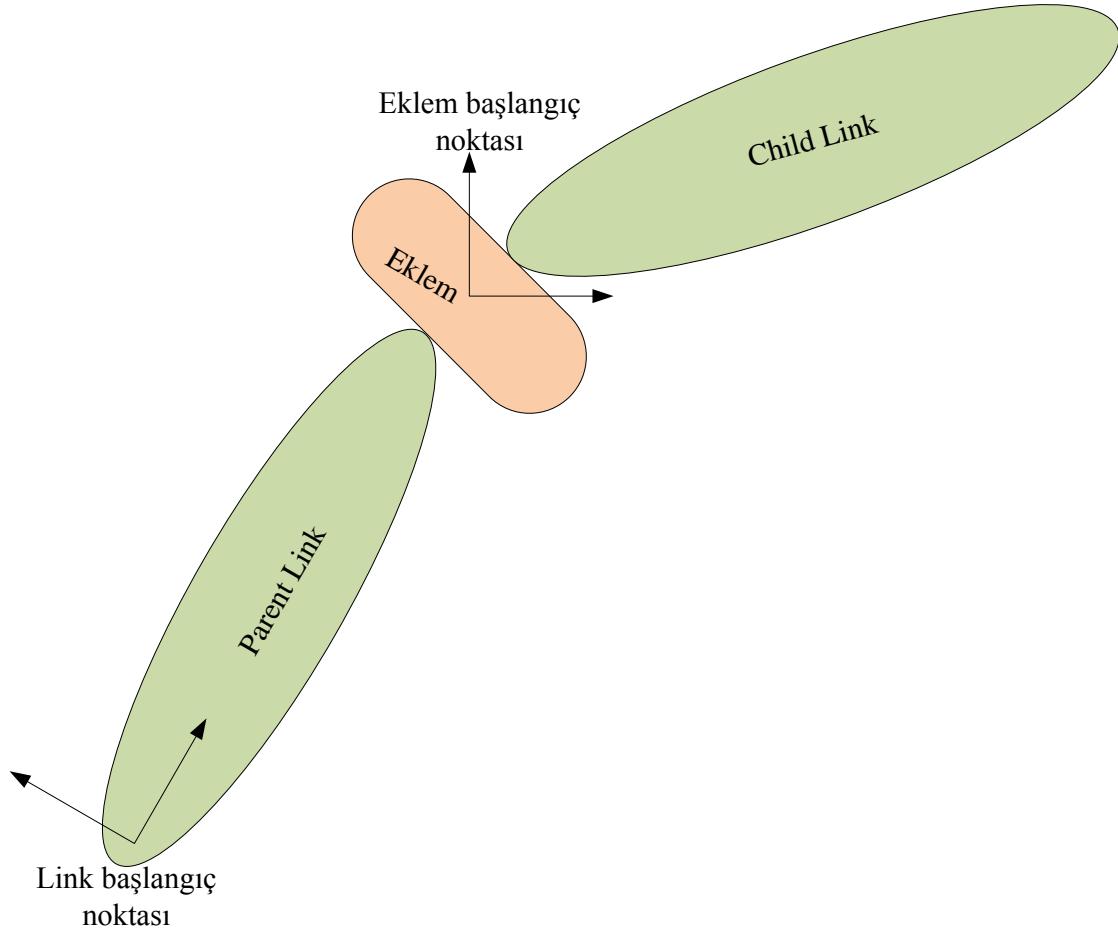
Şekil 4.3. Örnek Link Modeli.



**Şekil 4.4.** Örnek Link Modelinin Gazebo Görüntüsü.

***Eklemlerin Modellenmesi***

Şekil 4.5’de örnek eklem görüntüsü verilmektedir. Tip (Type), eklem tipini belirtmekte kullanılır. Döner eklemler için ‘revolute’, sabit eklemler için ‘fixed’, doğrusal hareketli eklemler için ise ‘prismatic’ tanımlamaları kullanılmaktadır. Orijin, eklemin merkezini belirtir. Parent link eklemden önceki link, child link ise eklemden sonraki linki tanımlar. Ayrıntılı bilgi <http://wiki.ros.org/urdf/XML/joint> adresinde bulunmaktadır. Şekil 4.6’de örnek eklem modeli görülmektedir.



**Şekil 4.5.** Örnek Eklem Görüntüsü.

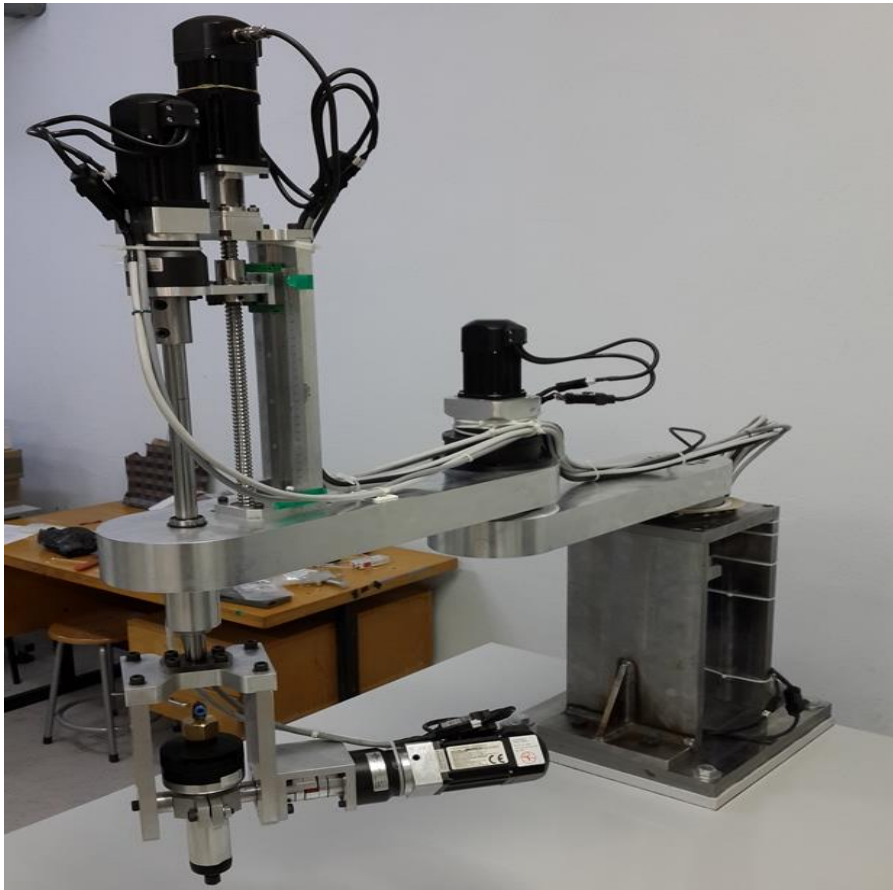
```

<joint name="shoulder_pan_joint" type="revolute"> <!-- Joint 1 -->
  <axis xyz="0 0 1"/>
  <parent link="base_link"/>
  <child link="shoulder_pan"/>
  <origin xyz="0 0 0.50459" rpy="0 -0 0"/>
  <limit lower="-2.35619449073033" upper="2.35619449073033" effort="2000" velocity="200"/>
</joint> <!-- Joint 1 -->

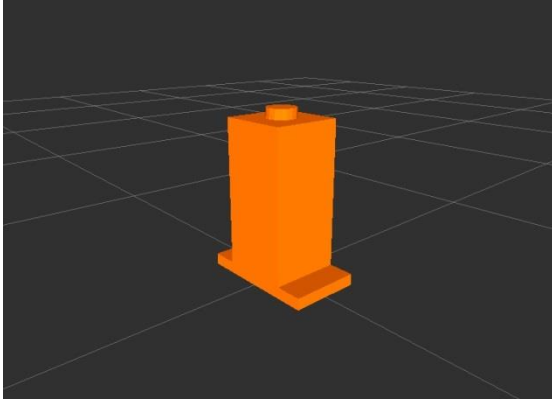
```

Şekil 4.6. Örnek Eklem Modeli.

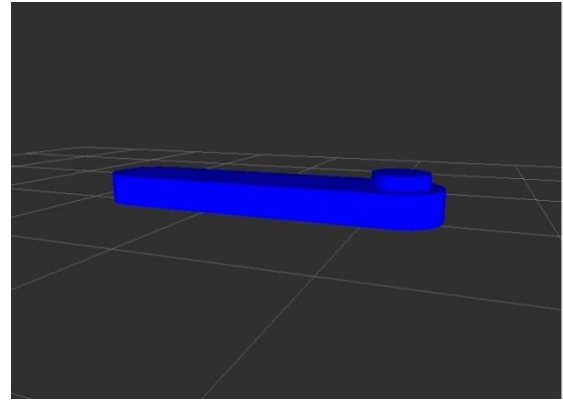
Şekil 4.7’ de resmi görülen robotun tüm eklem ve linkleri yukarıda tanımlandığı şekilde, gerçek parametreleri kullanılarak oluşturulmuştur. Şekil 4.8’de tanımlaması yapılan linkler ayrı ayrı gösterilmiştir. Şekil 4.9’de ise robotun link ve eklemleri bir arada gösterilmiştir.



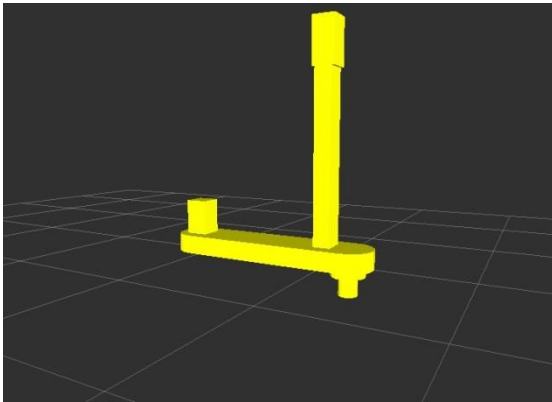
Şekil 4.7. Robotun Gerçek Görüntüsü (Robotes Robotic Solutions Ltd.).



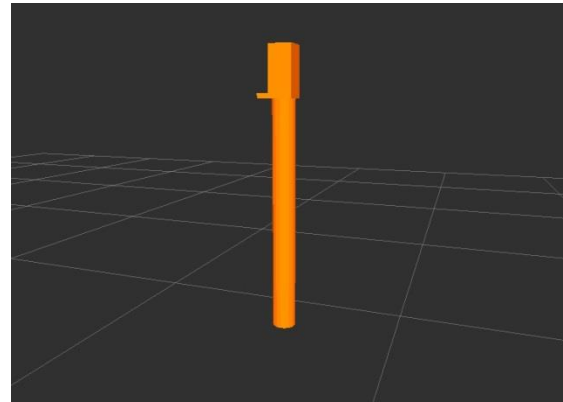
(a)



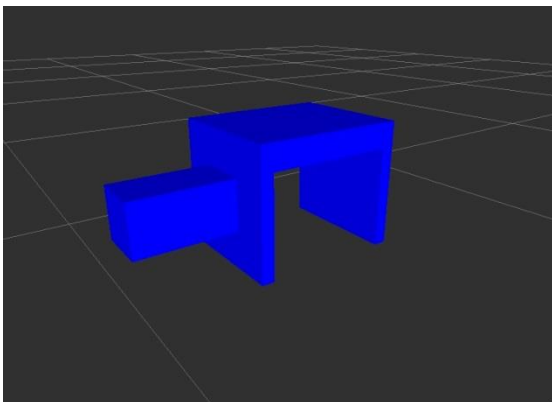
(b)



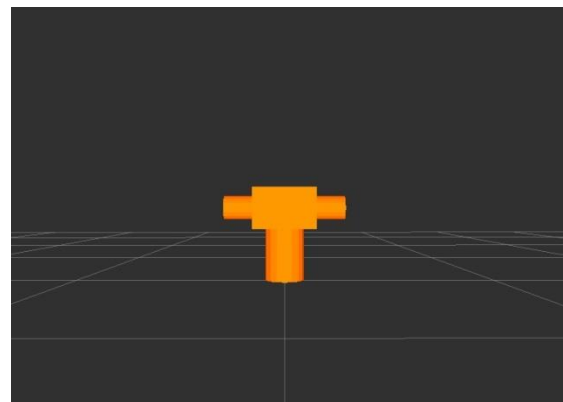
(c)



(d)

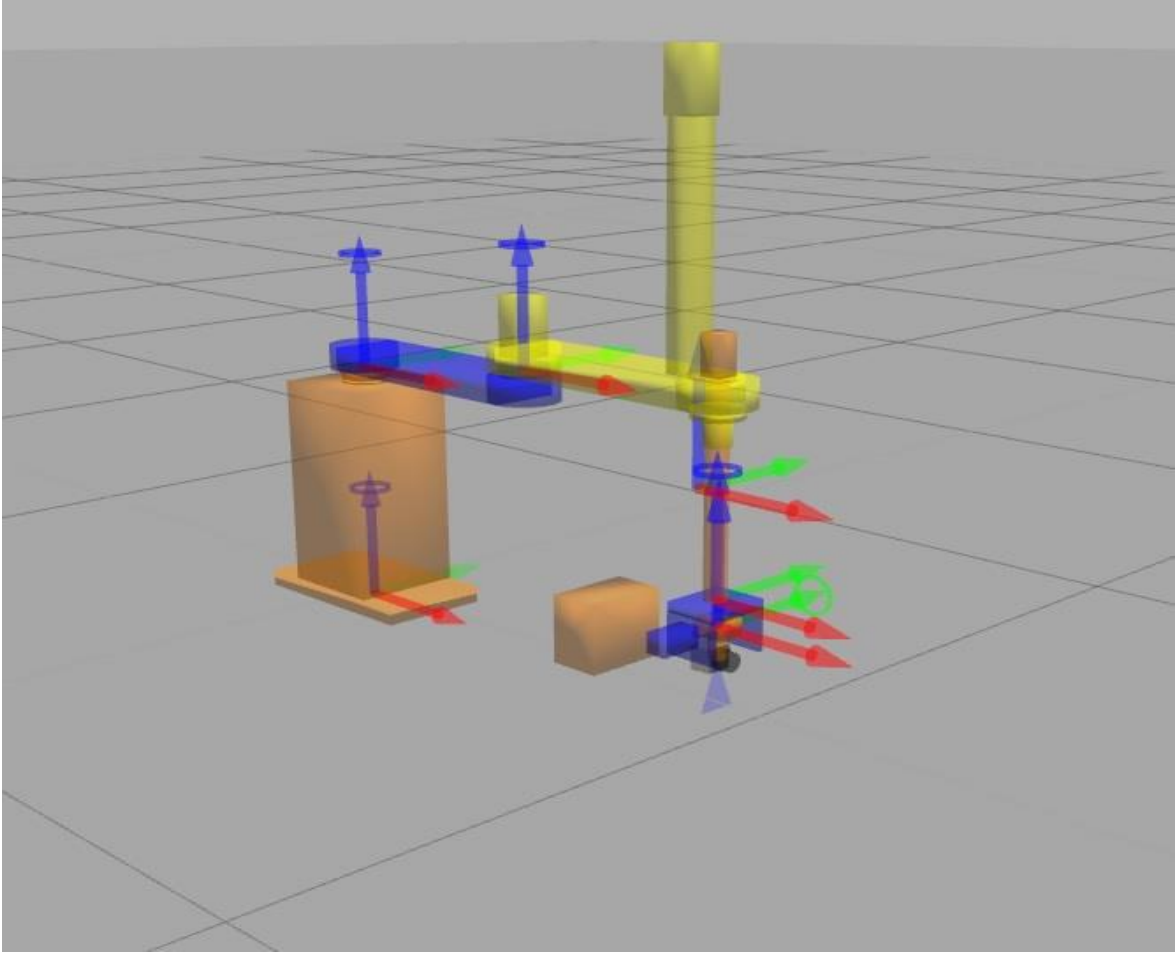


(e)



(f)

**Şekil 4.8.** Linklerinin Ayrı Ayrı Gazebo Görüntüleri a) link 0, b) link 1, c) link 2, d) link 3, e) link 4, f) link 5.



**Şekil 4.9.** Robot Modelin Link ve Eklemlerinin Gazebo Görüntüsü.

## 5. ÖNERİLEN YAKLAŞIM

Bu kısımda önerilen yaklaşım olarak, otonom robotik 3 boyutlu cisim modelleme yöntemleri geliştirilirken kullanılacak, benzetim ortamı tasarımı sunulmaktadır. Tasarımı yapılan benzetim ortamında, 5 ekseli SCARA tipi endüstriyel robot kol ve kol ucunda monteli lazer profil algılayıcı ile cisimlerin 3B modellemesinin benzetimi gerçekleştirilebilmektedir.

Şekil 5.1’de sistemin genel yapısı gösterilmektedir. Burada, “3B Benzetim Sistemi” modülü Bölüm 4’de anlatıldığı üzere robotun örnek modelinin oluşturulduğu ve Gazebo programı aracılığıyla benzetiminin yapıldığı kısımdır. Bu modülün yerini gerçek robot kolu da alabilir. ROS plugin olarak belirtilen, eklenti kodlanmakta ve bu kodlar vasıtasıyla benzetim ortamındaki robot ile denetim ve durum verilerin aktarımı mümkün olmaktadır. Projede kullanılan eklenti “gazebo\_ros\_joint\_trajectory” aşağıda verilmiştir.

```
<gazebo>
  <pluginname="gazebo_ros_joint_trajectory"filename="libgazebo_ros_joint_trajectory.so">
    <topicName>joint_trajectory</topicName>
    <robotNamespace>/myrobo</robotNamespace>
    <updateRate>0.0</updateRate>
  </plugin>
</gazebo>
```

Bu eklenti, yörünge planlamasına göre hesaplanan açı, hız ve ivme değerlerinin gazebo programına iletilmesinde kullanılmaktadır. Eklenti bir mesaj formatı tanımlamaktadır ve gazebo tarafından bu mesaj formatı tanınmaktadır. Bu mesaj formatına uygun şekilde hazırlanan yörünge planlaması verileri **/joint\_trajectory** başlığından yayınlanmaktadır. Gazebo tarafında ise mesaj formatı tanındığından veriler okunup robot kolunun hareketi sağlanmaktadır. Mesaj formatı aşağıda gösterilmiştir.

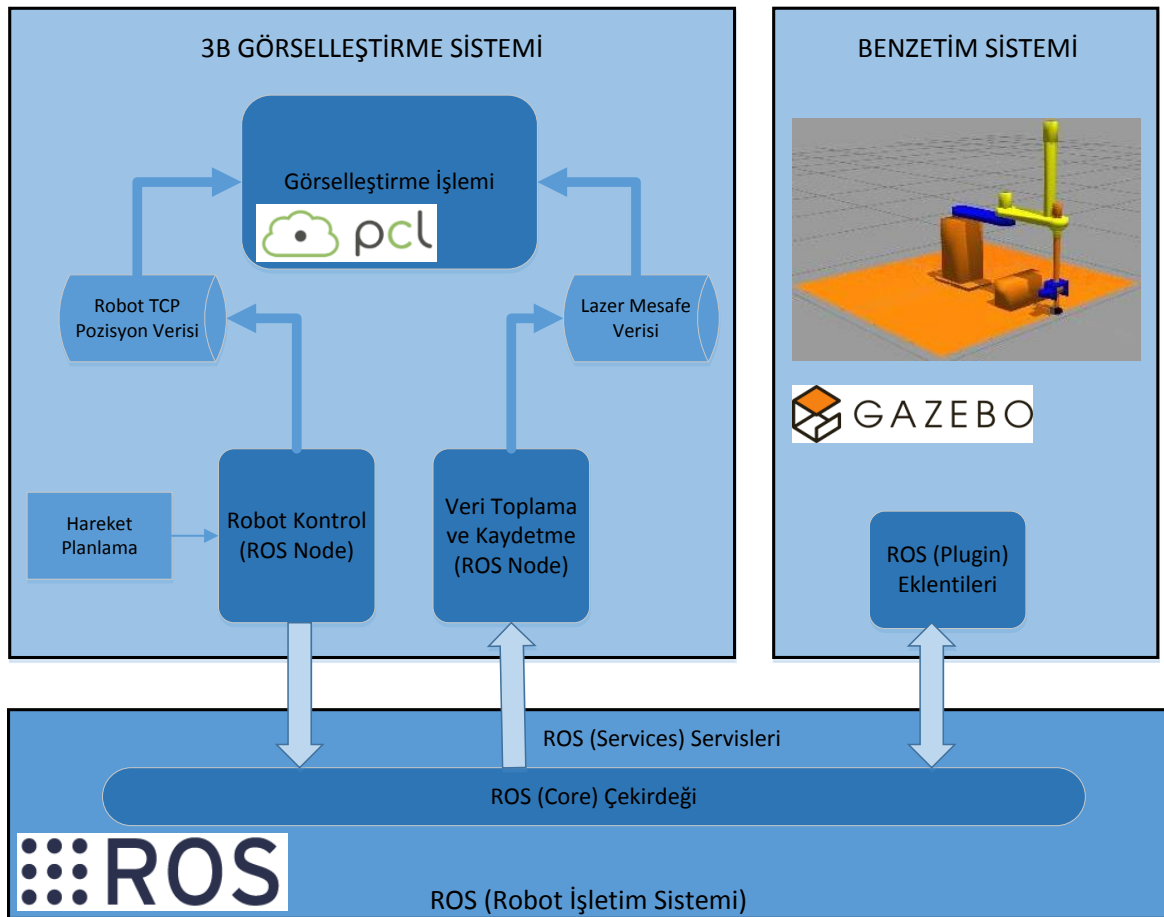
Std\_msgs / Header header

String [] joint\_names

Trajectory\_msgs / JointTrajectoryPoint [] points

“Görselleştirme Sistemi” olarak adlandırılan modül ise robotun hareketinin planlanması, hareket komutlarının benzetim ortamına iletilmesi ve lazer sensörden verilerin okunup, işlenip ve görüntü olarak ekranda gösterilmesini sağlayan kontrol kısmıdır. Bu modül 3 başlık altında açıklanacaktır;

- Robot Denetimi (ROS Düğümü)
- Veri Toplama ve Kayıt (ROS Düğümü)
- 3B Görselleştirme



Şekil 5.1. Sistemin Genel Yapısı (Ergür ve Özkan'dan, 2014).

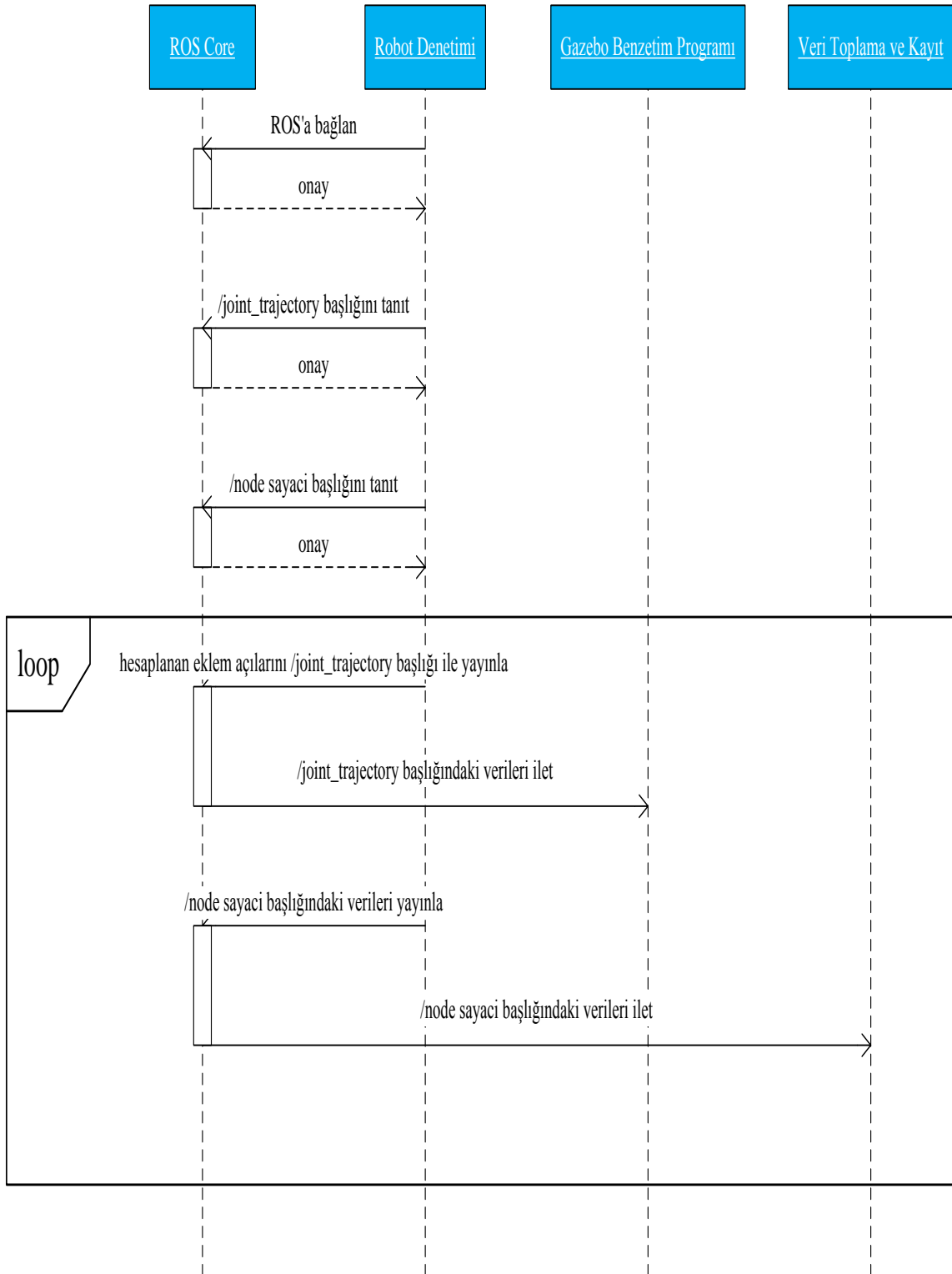
### 5.1. Robot Denetimi (ROS Dügümü)

Robot denetim programı çalıştırılmadan önce Gazebo ile bağlantısını sağlayacak Ros servislerinin çalıştırılması gereklidir. Bunun için terminal açılır ve “roslun” komutu ile Ros servisleri çalıştırılır. Daha sonra yeni terminal penceresinde “roslun proje-adi Publisher” komutu ile robot denetim programı çalıştırılır. Bu program, robotun istenilen yörüngede hareket etmesini ve lazer sensörünü kontrol eder. Robotun hareket ettirilmesinden önce lazer sensör aktif hale getirilir. Bunun sebebi robotun hareketine başladığında lazer sensörden bilgilerin gecikmesiz okunabilmesi içindir. Lazer sensör verileri Gazebo programında /base\_scan başlığı altında yayınlar. Robotun hareket ettirilmesi için gerekli bilgiler olan açı, hız ve ivme değerlerinin Gazebo programına gönderilebilmesi amacıyla “/joint\_trajectory” başlığını açar. Yörünge planlamasının yapılıp robotun hareket ettirilmesi gereken eklem değerleri gönderilir.

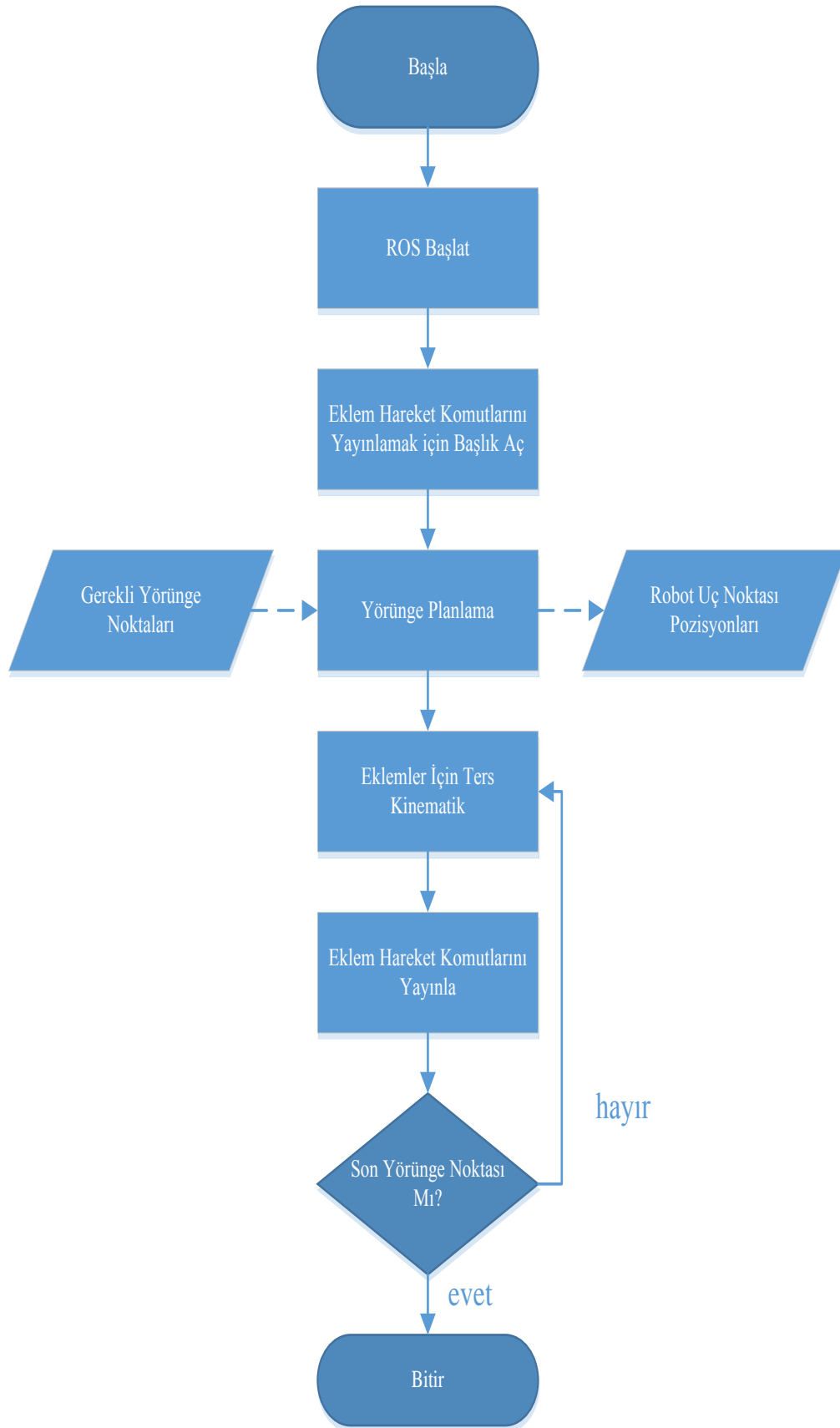
Robot denetim programının sistemdeki diğer programlarla arasındaki mesajlaşma Şekil 5.2’te gösterilmektedir. Görüldüğü gibi, robot denetim programı ROS core programı ile bağlantı oluşturduktan sonra mesaj başlıklarını açmaktadır. Sonra, eklem açılarını ve senkronizasyonu sağlayan değeri açılan başlıklardan yayınlamaktadır.

Şekil 5.3’te robot denetim programının akış şeması verilmektedir. Robot denetim programı, hareket planlaması öncesi gerekli işlemleri yaptıktan sonra yörünge üzerindeki noktaların listesinin bulunduğu dosyadan yörünge noktalarını okur. Bu noktalar arasında doğrusal hareket gerçekleştirilmesi için hareket planlaması yapılır. Yörünge planlaması ile elde edilen uç noktalardan ters kinematik ile eklem açıları hesaplanarak benzetim programına komut olarak iletilir.





Şekil 5.2. Robot Denetimindeki Mesajlaşmalar

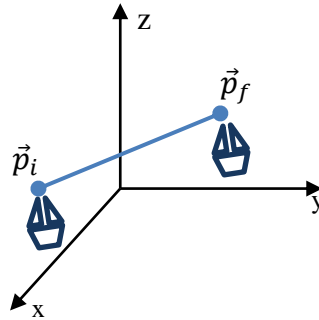


**Şekil 5.3.** Robot Denetimi akış diyagramı (Ergür ve Özkan'dan, 2014).

### 5.1.1. Yörünge planlama

#### 5.1.1.1. Kartezyen uzayında doğrusal hareket

İki nokta arasında doğrusal yörünge planlamasıdır. Şekil 5.4’de örnek gösterimde, Kartezyen uzayında iki nokta arasında doğrusal hareket gösterilmiştir.



**Şekil 5.4.** Kartezyen uzayında iki nokta arasında örnek doğrusal hareket (Ergür ve Özkan’dan, 2014).

Burada  $\vec{p}_i$  başlangıç noktası,  $\vec{p}_f$  bitiş noktasıdır. Başlangıç ve bitiş’de hızlar sıfırdır ve yol boyunca maksimum hız ( $V_{max}$ ) ve maksimum ivme ( $a_{max}$ ) bilindiği takdirde yol

$$\vec{p}(k) = \vec{p}_i + k(\vec{p}_f - \vec{p}_i) \quad (5.1)$$

formülü ile yol üzerindeki her nokta hesaplanabilir. Burada  $k = \sigma/L$ ,  $\sigma \in [0, L]$  olarak tanımlıdır. Başlangıç ve bitiş noktaları arasındaki Öklid uzaklık

$$L = \|\vec{p}_f - \vec{p}_i\| \quad (5.2)$$

olarak tanımlanır. Hareketin sağlanabilmesi için gereken hız ve ivmeler, aşağıda verilen denklemler ile hesaplanabilmektedir.

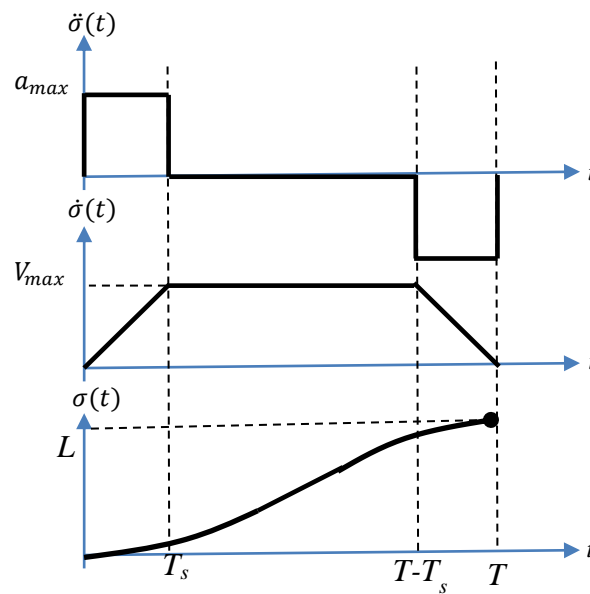
$$\dot{\vec{p}}(k) = \frac{d\vec{p}}{ds} \dot{k} = (\vec{p}_f - \vec{p}_i) \dot{k} = \frac{\vec{p}_f - \vec{p}_i}{L} \dot{\sigma} \quad (5.3)$$

$$\ddot{\vec{p}}(k) = (\vec{p}_f - \vec{p}_i) \ddot{k} = \frac{\vec{p}_f - \vec{p}_i}{L} \ddot{\sigma} \quad (5.4)$$

Şekil 5.5'teki grafikler elde edildiğinde hız, ivme ve zaman arasındaki ilişki (5.5) ve (5.6)'da gösterilmiştir.

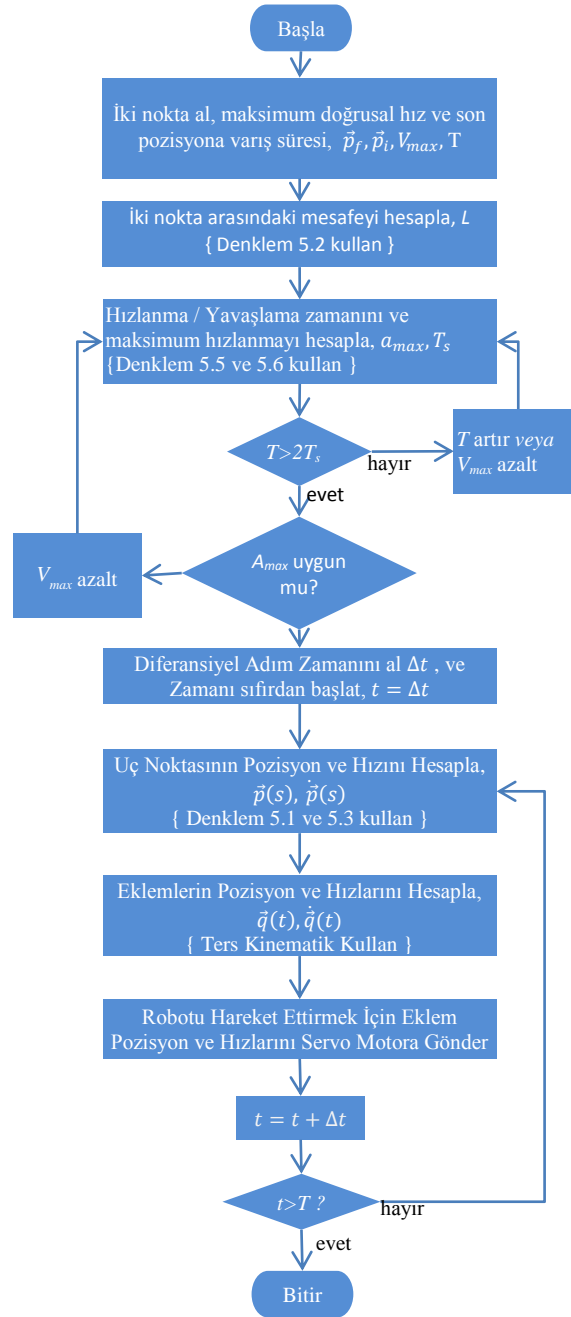
$$a_{max} = T_s V_{max} \quad (5.5)$$

$$a_{max} = \frac{V_{max}^2}{TV_{max} - L} \quad (5.6)$$



**Şekil 5.5.** Doğrusal yörünge için yol, hız ve ivme grafiği (Ergür ve Özkan'dan, 2014)

Şekil 5.6'daki akış diyagramında görüldüğü üzere ilk önce başlangıç ve bitiş noktası seçilir. Bu noktalar programda dosyadan takip edilecek sıraya göre okunur. Zaman değeri olan  $T$ 'de bilindiğinde (5.2), (5.5) ve (5.6)'daki formüller ile  $L$ ,  $a_{max}$  ve  $V_{max}$  değerleri hesaplanır. Hesaplanan  $a_{max}$  ve  $V_{max}$  değerleri uygun aralıkta ise adım sayısı ( $n$ )



**Şekil 5.6.** Doğrusal harekette yörünge planlamasının akış diyagramı (Ergür ve Özkan'dan, 2014).

değerine bağlı olarak başlangıç ve bitiş noktası arası n noktaya bölünür. (5.1), (5.3) ve (5.4)'teki formüller ile bu noktaların pozisyonları ile robot uç noktasının bu noktalardaki hız ve ivmesi hesaplanır. Formüllerdeki  $\sigma(t)$ ,  $\dot{\sigma}(t)$  ve  $\ddot{\sigma}(t)$  değerlerini  $V_{max}$ ,  $a_{max}$  ve  $T$ 'ye bağlı olarak (5.7), (5.8) ve (5.9)'daki formüller ile elde edilmektedir.

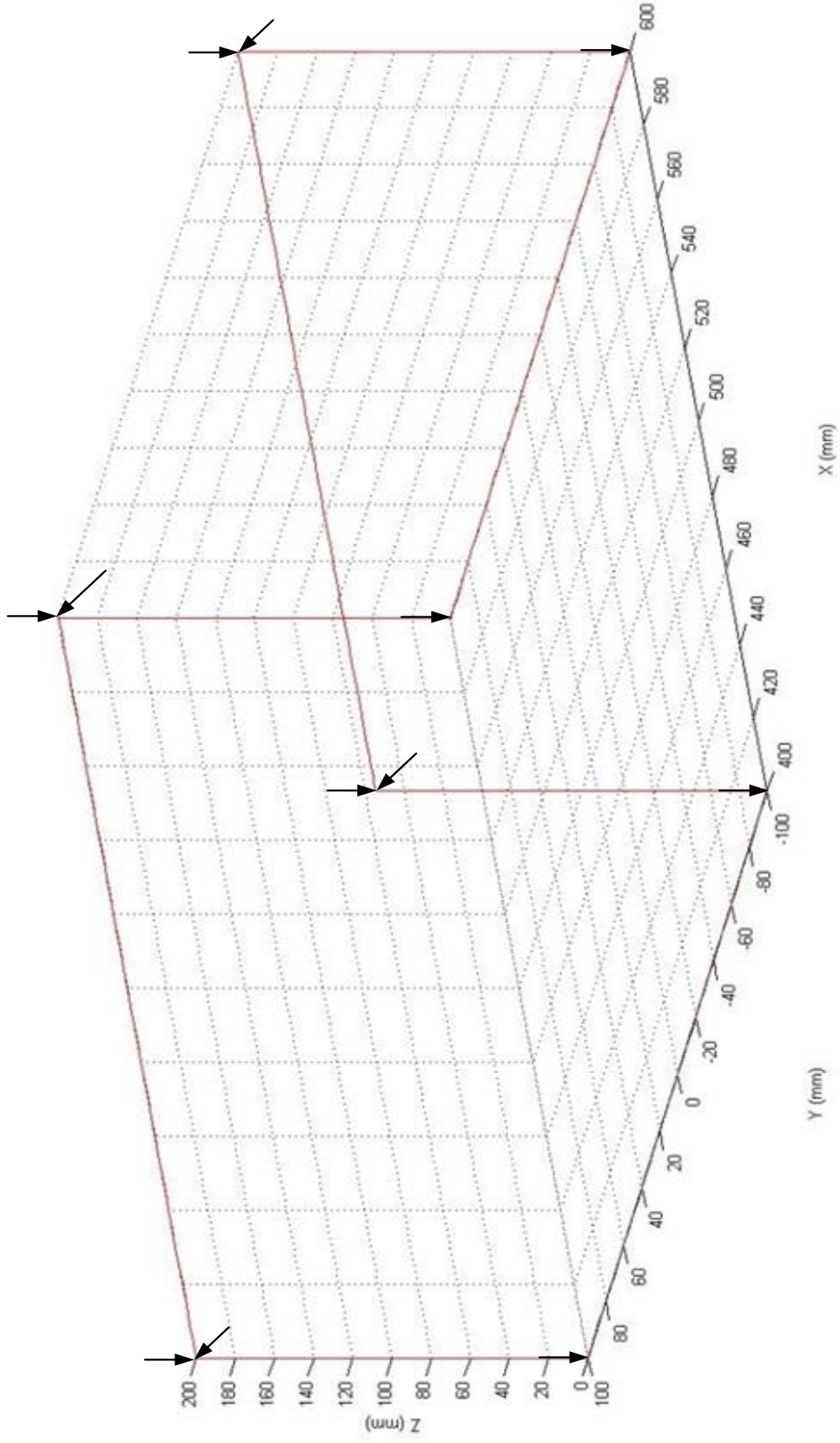
$$\sigma(t) = \begin{cases} a_{max}t^2/2 & , t \in [0, T_s] \\ V_{max}t - \frac{V_{max}^2}{2a_{max}} & , t \in [T_s, T - T_s] \\ -\frac{a_{max}(t-T)^2}{2} + V_{max}T - \frac{V_{max}^2}{a_{max}} & , t \in [T - T_s, T] \end{cases} \quad (5.7)$$

$$\dot{\sigma}(t) = \begin{cases} a_{max} * t & , t \in [0, T_s] \\ V_{max} & , t \in [T_s, T - T_s] \\ -a_{max} * (t - T) & , t \in [T - T_s, T] \end{cases} \quad (5.8)$$

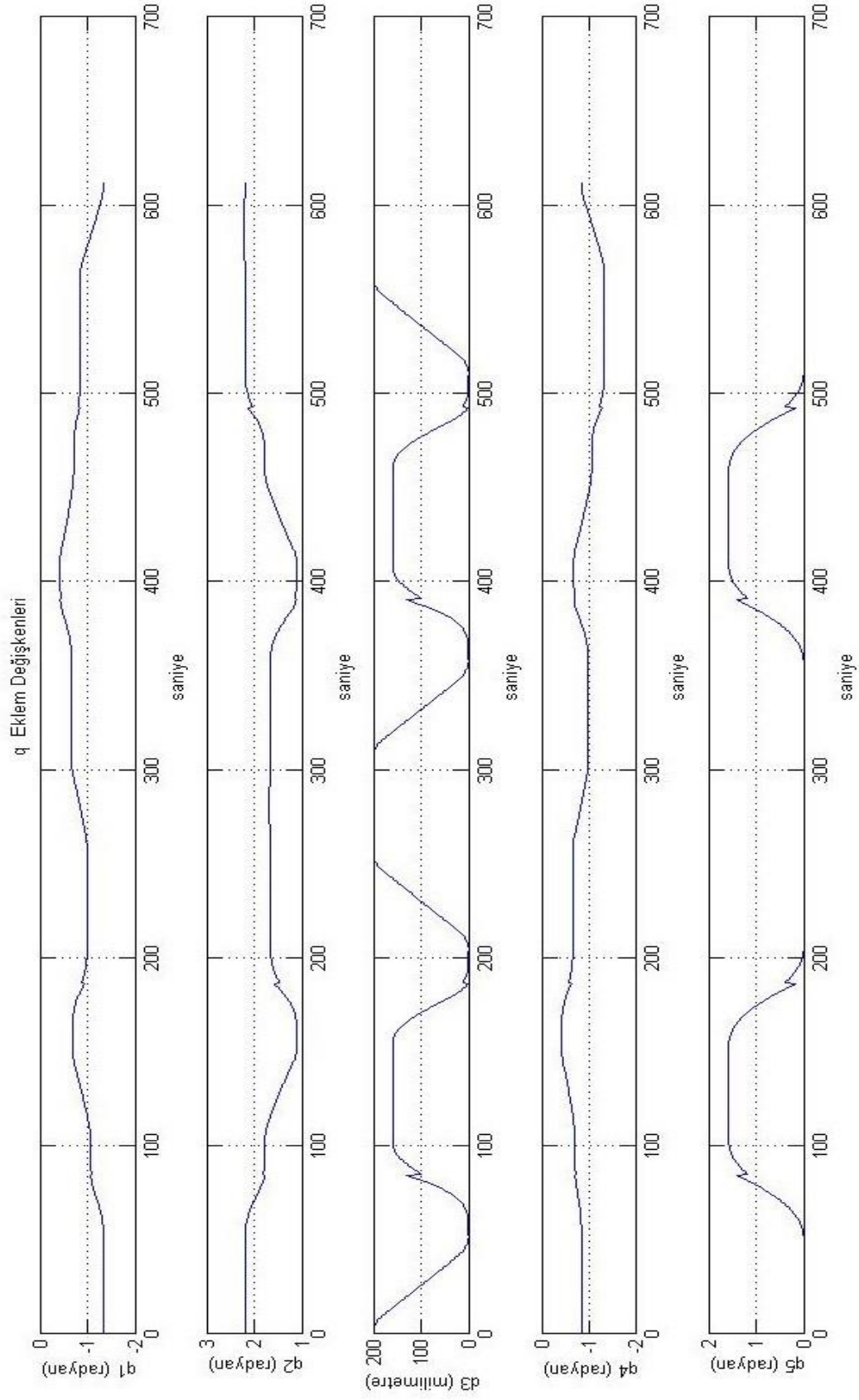
$$\ddot{\sigma}(t) = \begin{cases} a_{max} & , t \in [0, T_s] \\ 0 & , t \in [T_s, T - T_s] \\ -a_{max} & , t \in [T - T_s, T] \end{cases} \quad (5.9)$$

### 5.1.1.2. Kartezyen uzayında doğrusal hareket testleri

Yörünge planlaması algoritması, öncelikle Matlab ortamında kodlanarak test edilmiştir. Şekil 5.7'de iki nokta arası doğrusal hareket için planlanan yörünge, robotun oluşturulan bu yörüngede hareketi sonucu eklemlerinin açı değerleri Şekil 5.8'de, hız değerleri Şekil 5.9'da ve ivme değerleri Şekil 5.10'de gösterilmiştir. Şekil 5.11'de ise Matlab altında yapılan robotun hareket benzetimi görülmektedir.

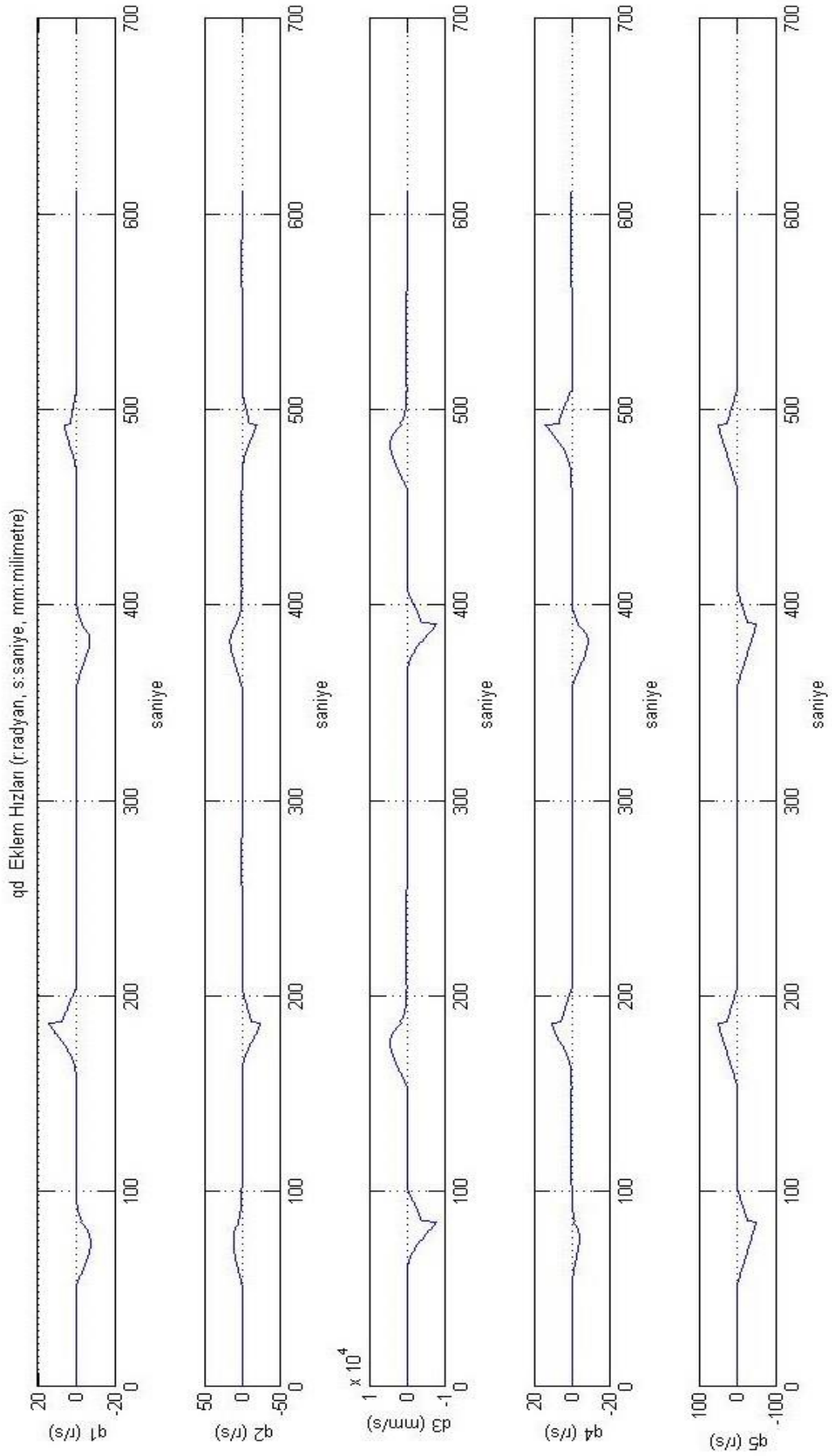


**Şekil 5.7.** İki nokta arası doğrusal hareket için planlanan yörünge.

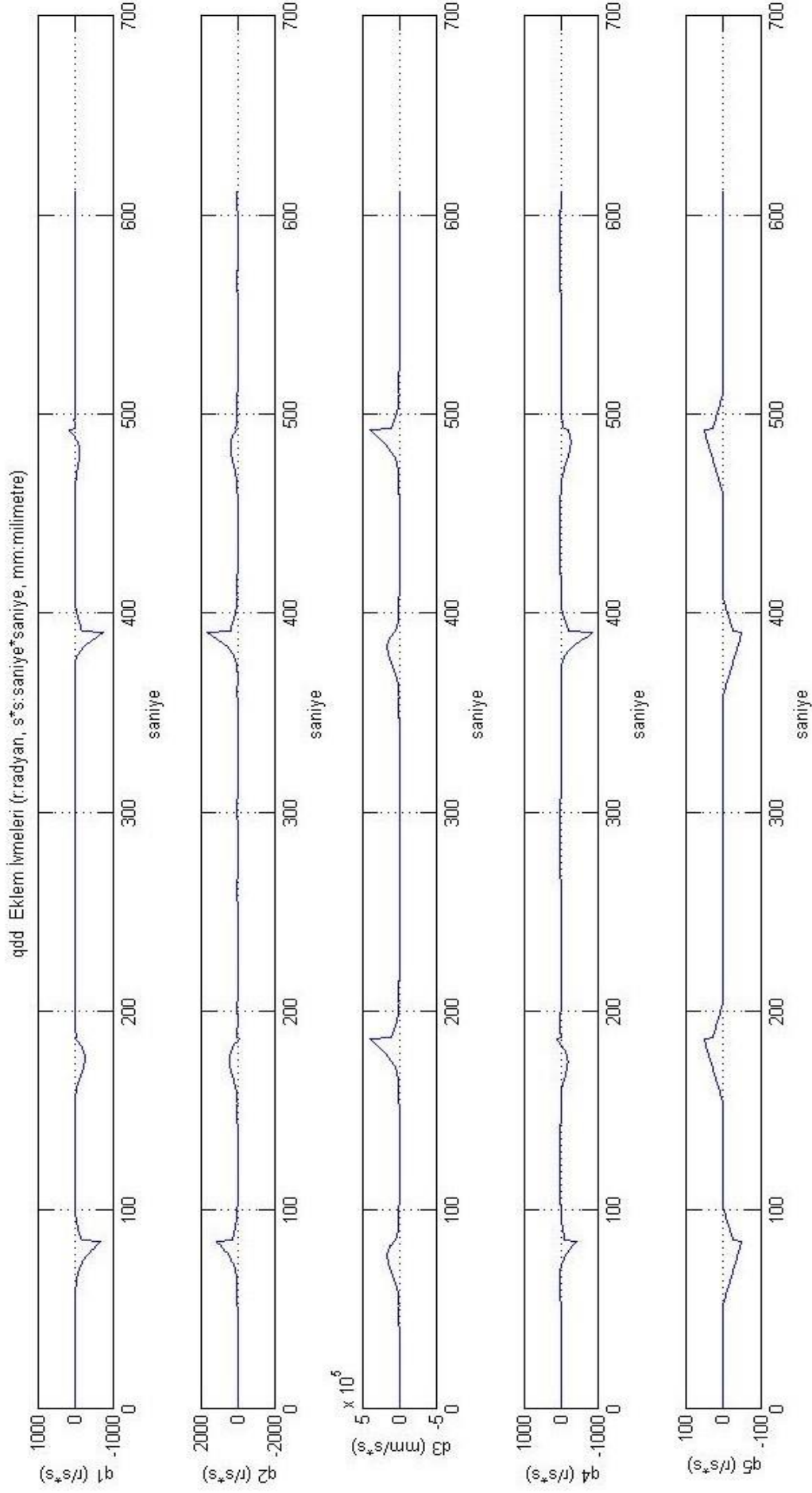


**Şekil 5.8.** İki nokta arası doğrusal hareket için eklem değişkenleri.

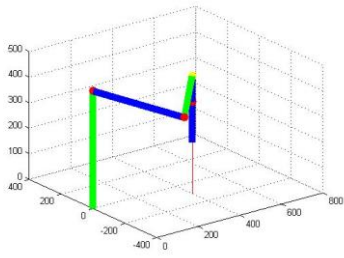




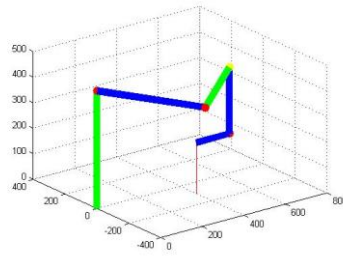
Şekil 5.9. İki nokta arası doğrusal hareket için eklem hızları.



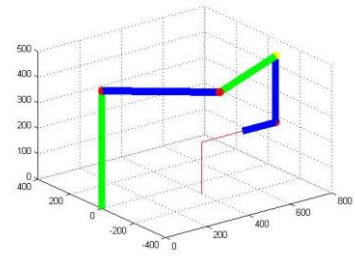
**Şekil 5.10.** İki nokta arası doğrusal hareket için eklemler ivmeleri.



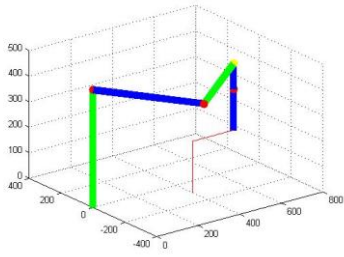
(a)



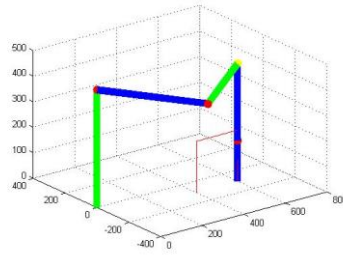
(b)



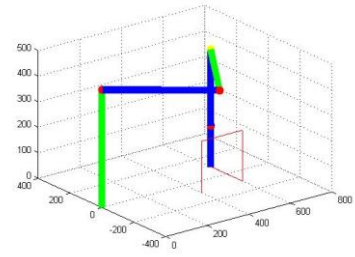
(c)



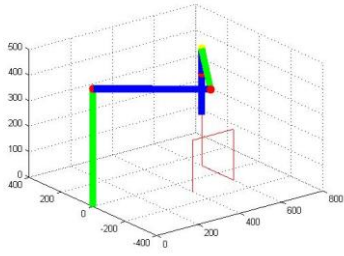
(d)



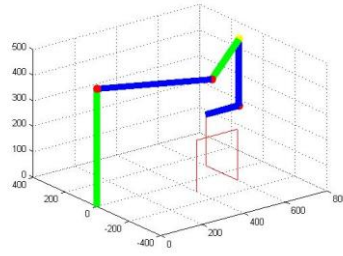
(e)



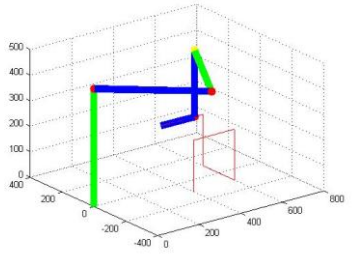
(f)



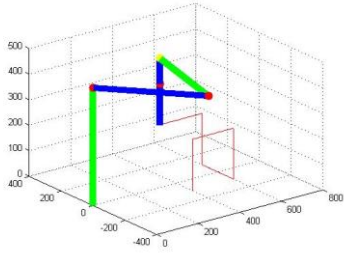
(g)



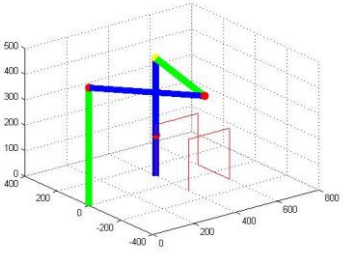
(h)



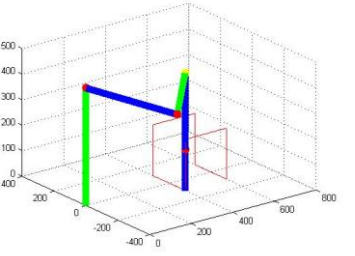
(i)



(j)



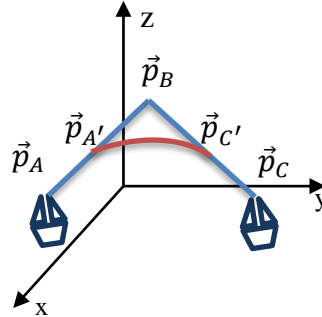
(k)



(l)

**Şekil 5.11.** a-k Matlab ortamında iki nokta arası yörünge planlaması esnasında elde edilen benzetim görüntüleri.

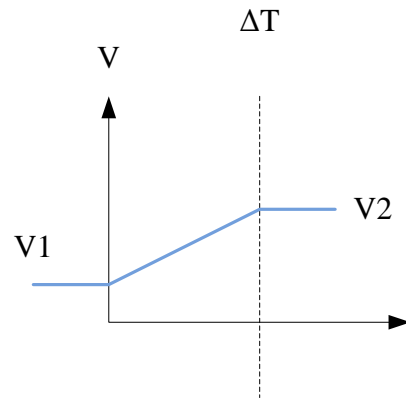
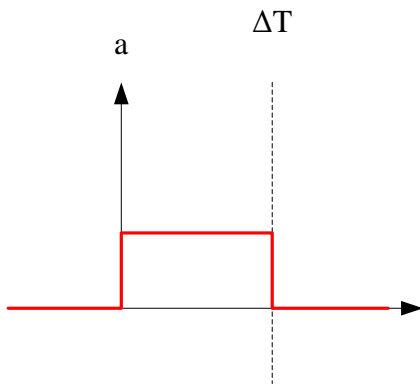
### 5.1.1.3. Kartezyen uzayında ara nokta ile doğrusal hareket



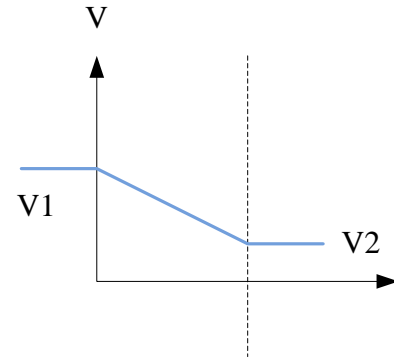
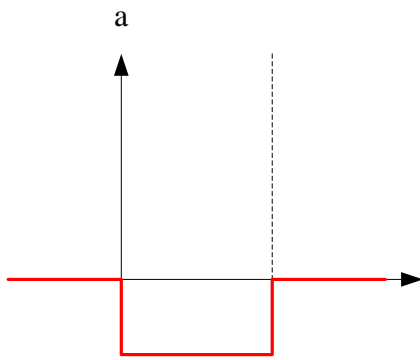
Şekil 5.12. Ara nokta ile doğrusal hareket.

İkiden fazla nokta kullanılarak yapılmak istenen yörünge planlamalarında iki nokta arası yörünge planlaması metodu yetersiz kalmaktadır çünkü bu metotta başlangıç ve bitiş hızları sıfırdır. Bunun sonucunda robot her iki nokta arasında durmaktadır bu da zaman açısından dezavantajdır. Robotun sadece ilk ve son noktada hızının sıfır olup diğer noktalarda sabit hızla geçmesi için ara noktalar ile yörünge planlaması metodu tercih edilmelidir. Şekil 5.12’de gösterilen yörüngeyi örnek alırsak  $\vec{P}_B$  noktasını açısının 90 dereceden büyük olduğu durumlarda robot uç noktası  $\vec{P}_A$ ,  $\vec{P}_B$  ve  $\vec{P}_C$  noktalarını sabit bir hızla takip edebilmektedir.  $\vec{P}_B$  açısının 90 dereceden küçük olduğu durumlarda ise  $\vec{P}_{A'}$  ve  $\vec{P}_{C'}$  noktaları arasında eğik hareket yaparak  $\vec{P}_B$  noktasını kullanmadan sabit hızla hareket yapılabilmektedir. Eğik hareket yapılmasının sebebi açının 90 dereceden küçük olduğu durumlarda doğrultu değiştiğinden  $\vec{P}_B$  noktasında yine robotun durması gerekmektedir. Robotun durmadan hareket etmesi için eğik hareket tercih edilerek hareket sağlanmaktadır.

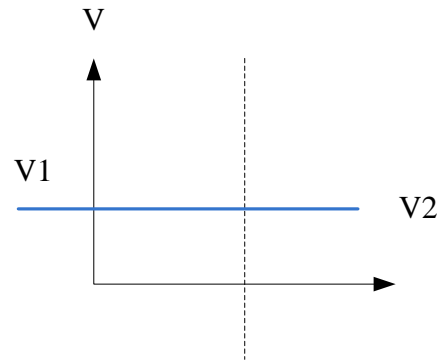
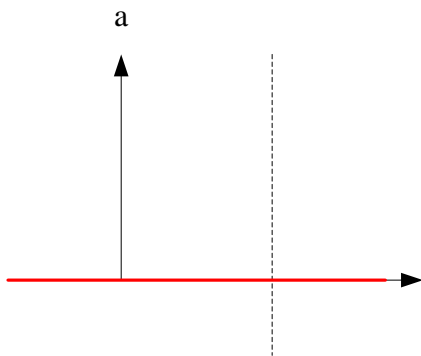
Ara nokta ile doğrusal harekette  $\vec{P}_A$  ile  $\vec{P}_B$  arası  $V_1$ ,  $\vec{P}_B$  ile  $\vec{P}_C$  arası  $V_2$  hızı ile hareket etmektedir.  $V_1$  ve  $V_2$  için 3 durum mevcuttur  $V_1 > V_2$ ,  $V_1 < V_2$ ,  $V_1 = V_2$  ve Şekil 5.13’da gösterilmiştir.



(a)



(b)



(c)

**Şekil 5.13.** Ara nokta ile harekette a)  $V_1 > V_2$ , b)  $V_1 < V_2$ , c)  $V_1 = V_2$  olduğu durumlar.

Uç noktası için yörünge noktası, hız ve ivme formülleri (5.10), (5.11) ve (5.12)'de verilmiştir.

$$p(t) = P_{A'} + V_1 K_{AB} t + t^2 / 2 \Delta T (V_2 K_{BC} - V_1 K_{AB}) \quad (5.10)$$

$$\dot{p}(t) = V_1 K_{AB} + t / \Delta T (V_2 K_{BC} - V_1 K_{AB}) \quad (5.11)$$

$$\ddot{p}(t) = 1 / \Delta T (V_2 K_{BC} - V_1 K_{AB}) \quad (5.12)$$

Denklemlerdeki  $K_{AB}$  ve  $K_{BC}$  birim vektörlerdir ve (5.13) ve (5.14) gösterilmiştir.

$$K_{AB} = \frac{P_B - P_A}{\|P_B - P_A\|} \quad (5.13)$$

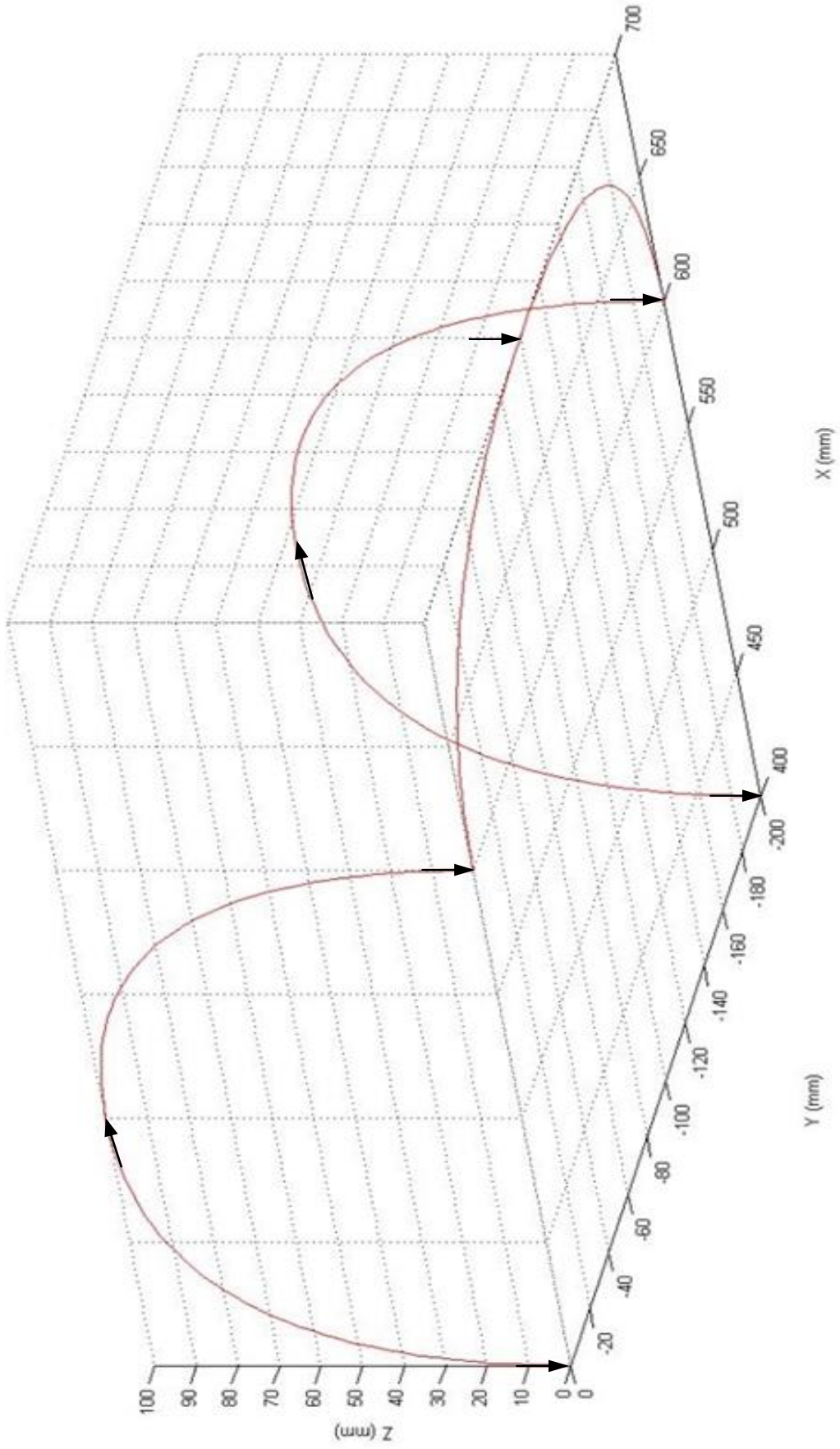
$$K_{BC} = \frac{P_C - P_B}{\|P_C - P_B\|} \quad (5.14)$$

$\Delta T = 2d_1/V_1$  veya  $\Delta T = 2d_2/V_2$  ile hesaplanabilir. Burada  $d_1$  ve  $d_2$  (5.15)'de gösterilmiştir.

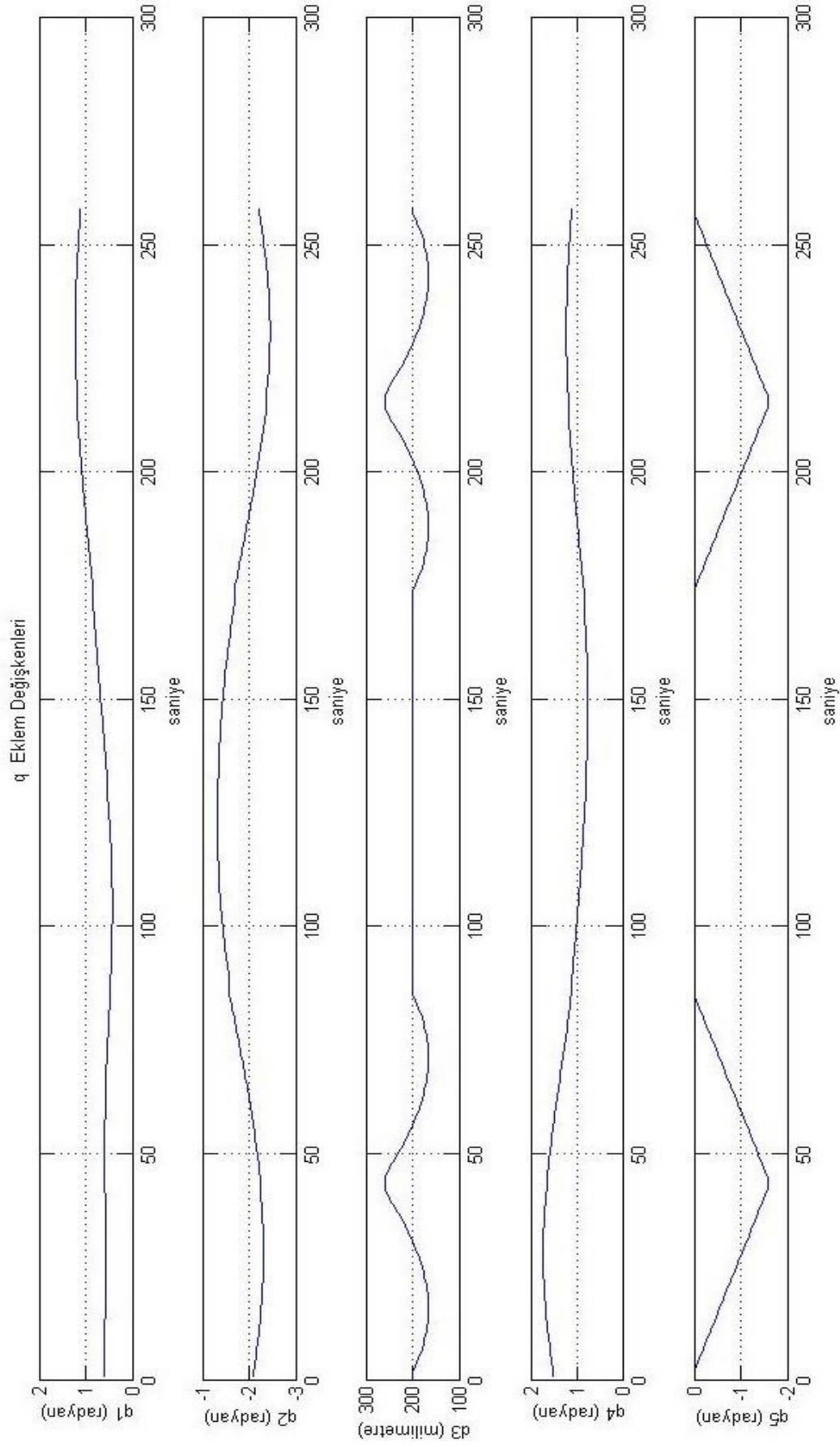
$$\begin{aligned} P_B - P_{A'} &= d_1 K_{AB} \\ P_{C'} - P_B &= d_2 K_{BC} \end{aligned} \quad (5.15)$$

#### **5.1.1.4. Kartezyen uzayında ara nokta ile doğrusal hareket testleri**

Matlab ortamında yapılan testlere göre Şekil 5.14-17'de ara nokta ile doğrusal hareket için planlanan yörünge ve planlanan yörüngede hareket sonucu eklemlerinin açısı, hız ve ivme değerleri gösterilmiştir. Şekil 5.18'de ise Matlab ortamında robotun hareket benzetimi esnasında elde edilen görüntüler verilmiştir.

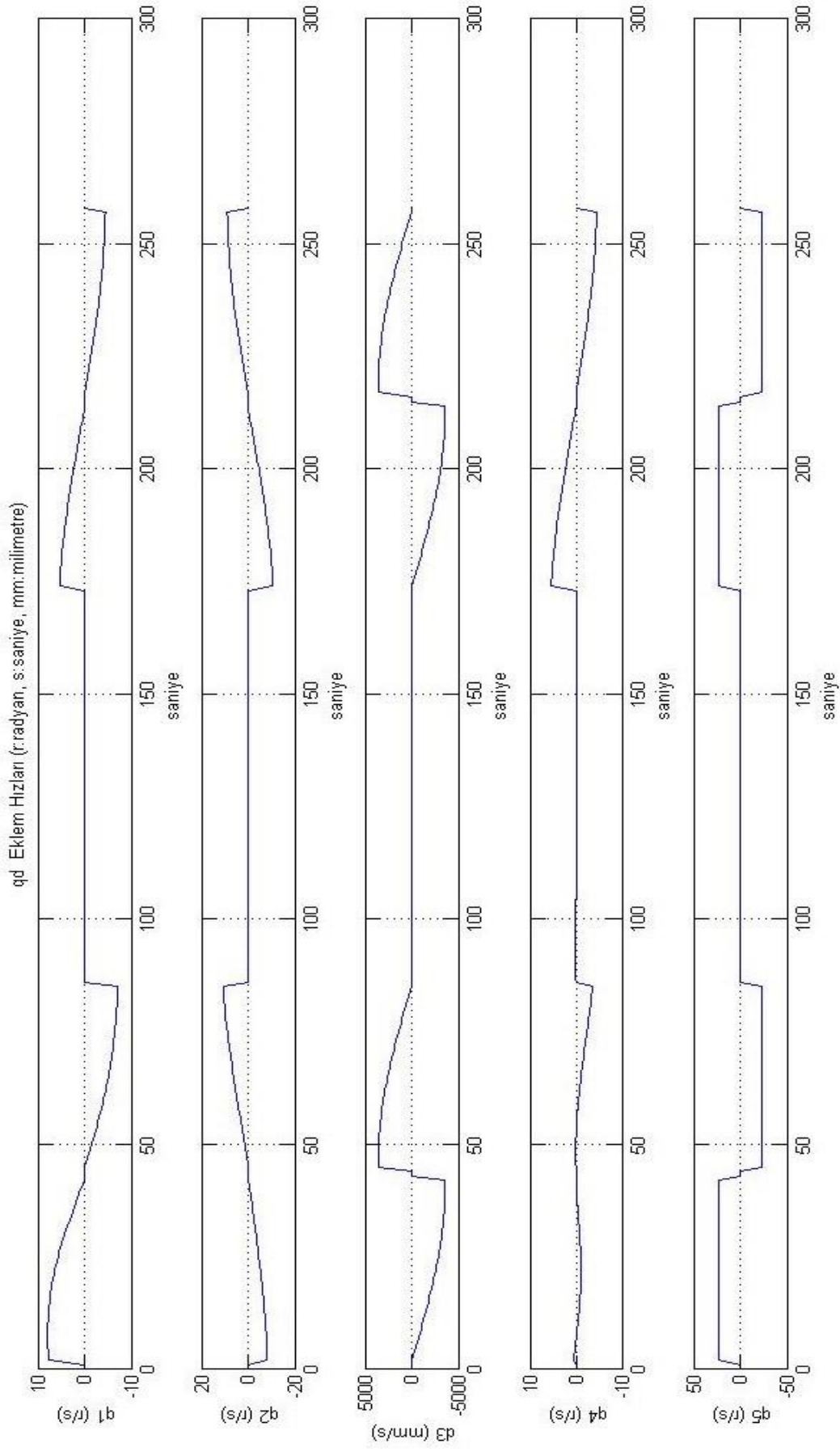


**Şekil 5.14.** Ara nokta ile doğrusal hareket için planlanan yörünge.

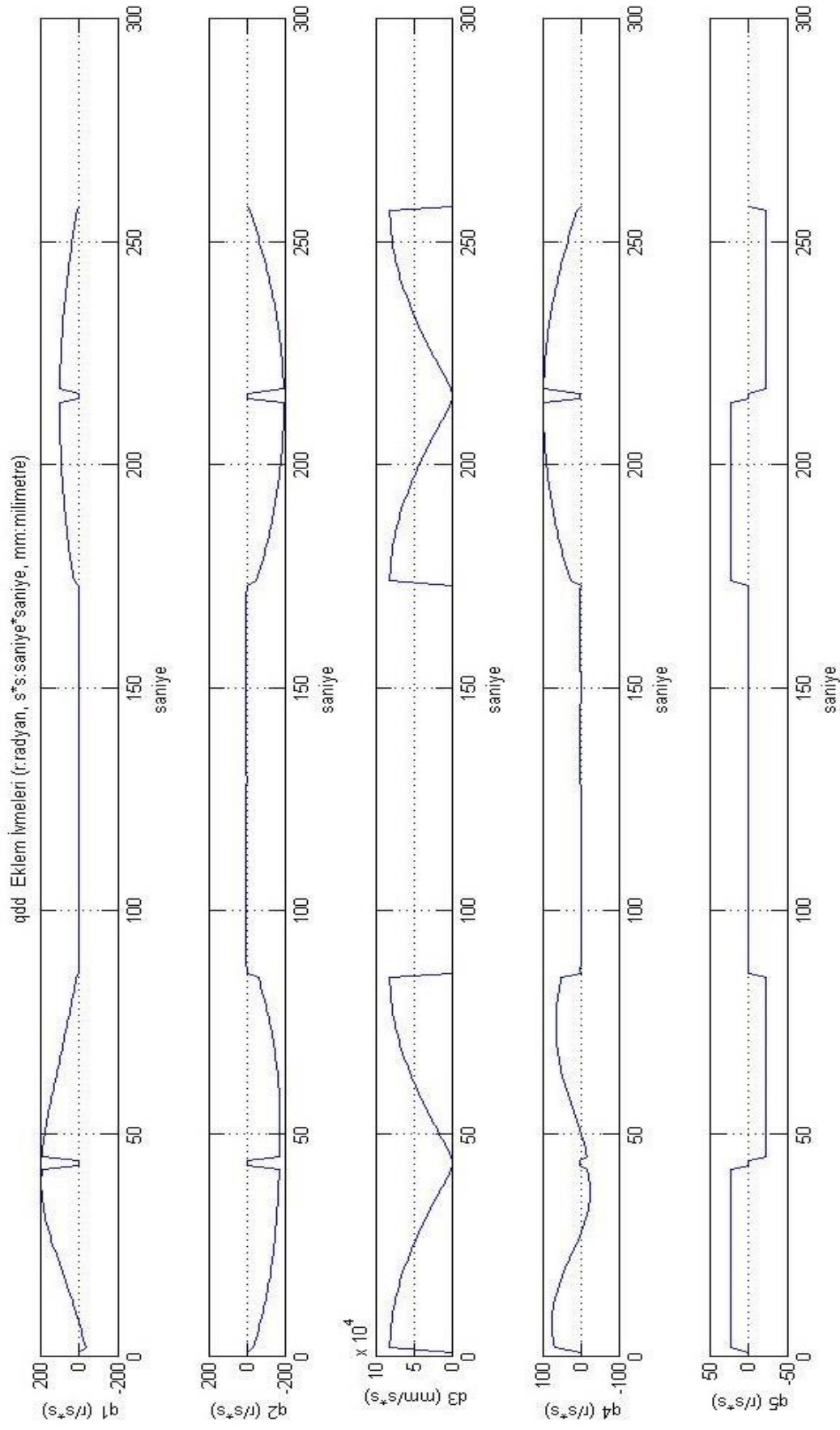


**Şekil 5.15.** Ara nokta ile doğrusal hareket için eklem değişkenleri.

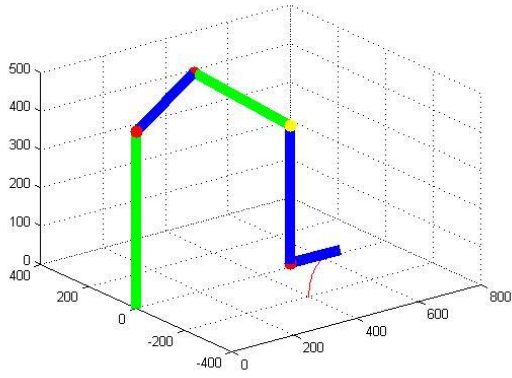




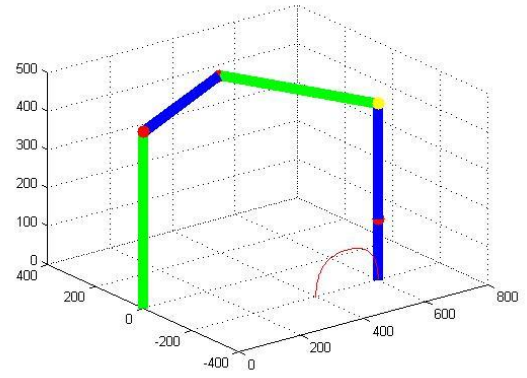
**Şekil 5.16.** Ara nokta ile doğrusal hareket için eklem hızları.



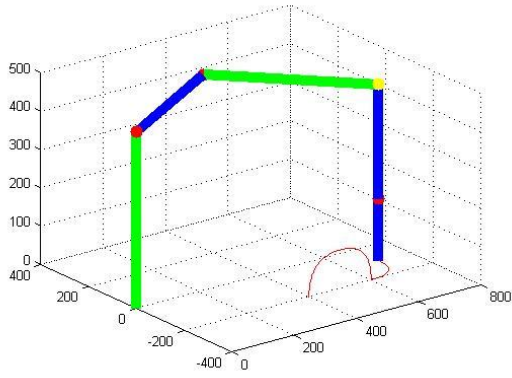
**Şekil 5.17.** Ara nokta ile doğrusal hareket için eklem ivmeleri.



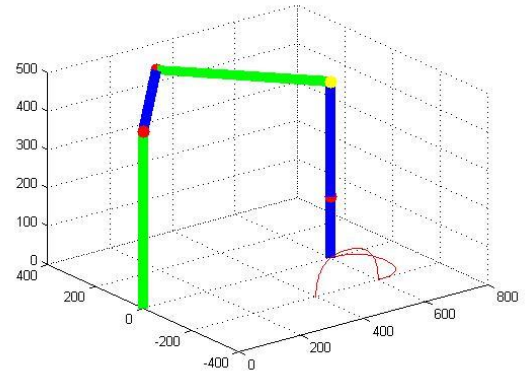
(a)



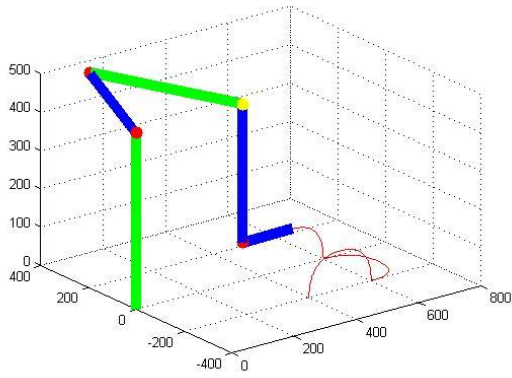
(b)



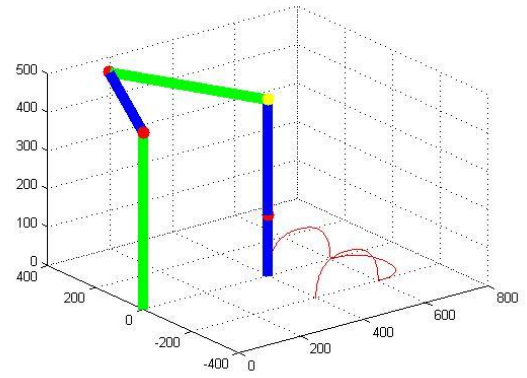
(c)



(d)



(e)



(f)

Şekil 5.18. a-f Matlab ortamında ara nokta ile yörünge planlaması esnasında elde edilen benzetim görüntüleri.

### 5.1.2. Yörünge planlama algoritması

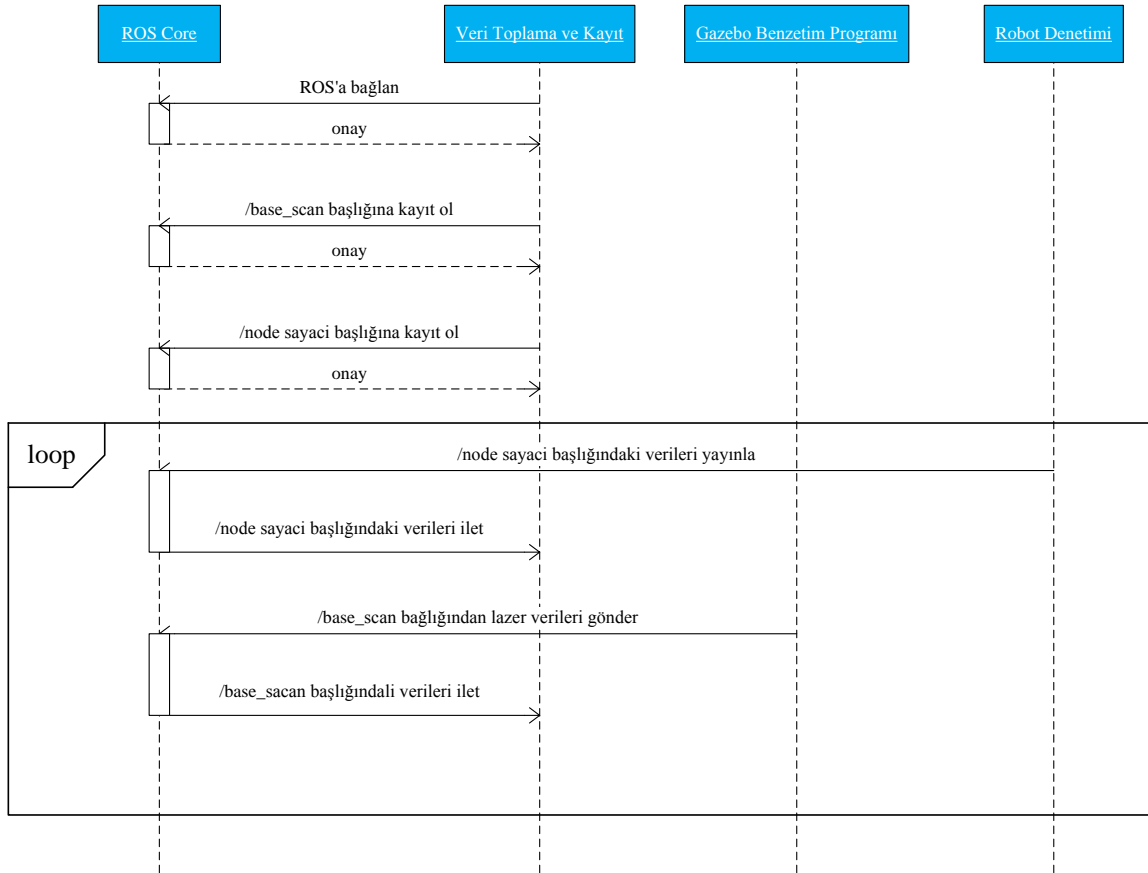
Yörünge planlama algoritmasının, önerilen sistemde kullanılması için C/C++ programlama dilinde kodlanmıştır. Bu algoritma için akış şeması Şekil 5.6’da verilmektedir. Bu algoritmada, iki noktanın koordinatları, maksimum hız ve hareket süresi girdi olarak istenmektedir. Maksimum hız belirlenirken, uç noktanın bu hızla hareket etmesi esnasında eklemelerdeki hızların sağlanabilir olması gözetilmelidir. Hareket süresinin değeri de, ivmelenme değerini etkilemektedir. Bu nedenle belirlenen hareket süresi için eklemelerin ulaşması gereken ivmeler, eklem eyleyicileri tarafından sağlanabilecek seviyede olmalıdır.

### 5.2. Veri Toplama ve Kayıt (ROS Düğümü)

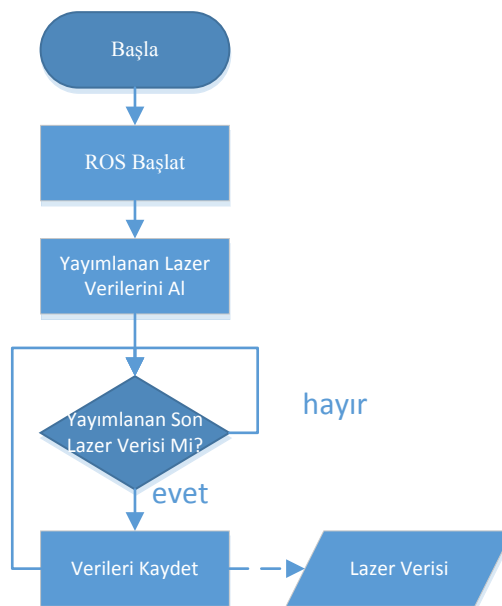
Robot Denetim programı ile birlikte eşzamanlı çalıştırılan diğer program “**Veri Toplama ve Kayıt**” programıdır. Bu program “/base\_scan” başlığı altında yayınlanan verileri myrobo\_laser\_data.txt isimli dosyaya kaydeder. Veriler;

Sıra no \$ zaman \* veri veri veri veri

şeklinde sırayla kaydedilir. Şekil 5.19’de veri toplama ve kayıt esnasındaki mesajlaşmaların sıra diyagramı gösterilmiştir. Şekil 5.20’de ise programın akış şeması gösterilmiştir.



Şekil 5.19. Veri Toplama ve Kayıt'daki Mesajlaşmalar.



Şekil 5.20. Veri Toplama ve Kayıt akış diyagramı (Ergür ve Özkan'dan, 2014).

### 5.3. 3B Görselleştirme

Robot ile nesne tarama işlemi bittiğinde Robot Denetim ve Veri Toplama ve Kayıt programları kapatılır. Terminal penceresinde “rosvun proje-adi pcd” komutu ile görüntüyü oluşturacak program çalıştırılır. Bu programın yazılmasında Point Cloud Library (PCD) kütüphanesinden faydalanılmıştır. Elde edilen lazer verileri lazer algılayıcı üzerindeki koordinat sistemine göre tanımlıdır. Görüntüyü oluşturabilmek için, bu mesafelerin robotun referans koordinat sistemine göre dönüştürülmesi gereklidir. Bu sebeple (5.16)’daki dönüşüm formülü kullanılmıştır.

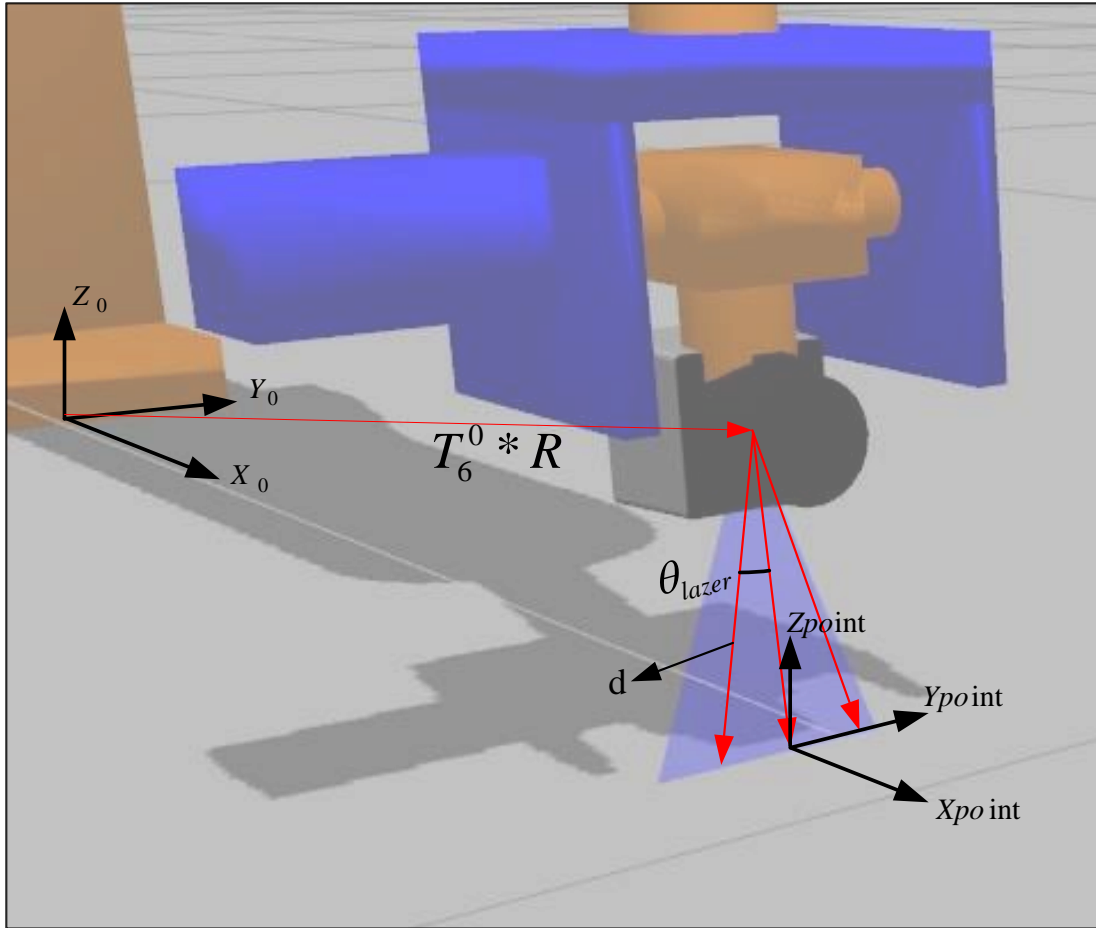
$$\begin{bmatrix} X_{point} \\ Y_{point} \\ Z_{point} \end{bmatrix} = T_6^0 * R * \begin{bmatrix} d * \cos \theta \\ d * \sin \theta \\ 0 \end{bmatrix} \quad (5.16)$$

Burada  $T_6^0$  denklem (4.8)’de gösterilmiştir. R, Y ekseninde  $90^\circ$  dönüşüm matrisidir ve denklem (5.17)’de gösterilmiştir. Lazer dönüşüm matrisi denklem (5.16)’verilmiştir ve görsel olarak ise Şekil 5.21 gösterilmiştir.

$$R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (5.17)$$

$$\begin{bmatrix} X_{point} \\ Y_{point} \\ Z_{point} \end{bmatrix} = \begin{bmatrix} X_{lazer} - \sin(\theta_5) * \cos(\theta_1 + \theta_2 + \theta_4) * d * \cos(\theta_{lazer}) - \sin(\theta_1 + \theta_2 + \theta_4) * d * \sin(\theta_{lazer}) \\ Y_{lazer} - \sin(\theta_5) * \sin(\theta_1 + \theta_2 + \theta_4) * d * \cos(\theta_{lazer}) + \cos(\theta_1 + \theta_2 + \theta_4) * d * \sin(\theta_{lazer}) \\ Z_{lazer} - \cos(\theta_{lazer}) * d * \cos(\theta_{lazer}) \end{bmatrix} \quad (5.18)$$

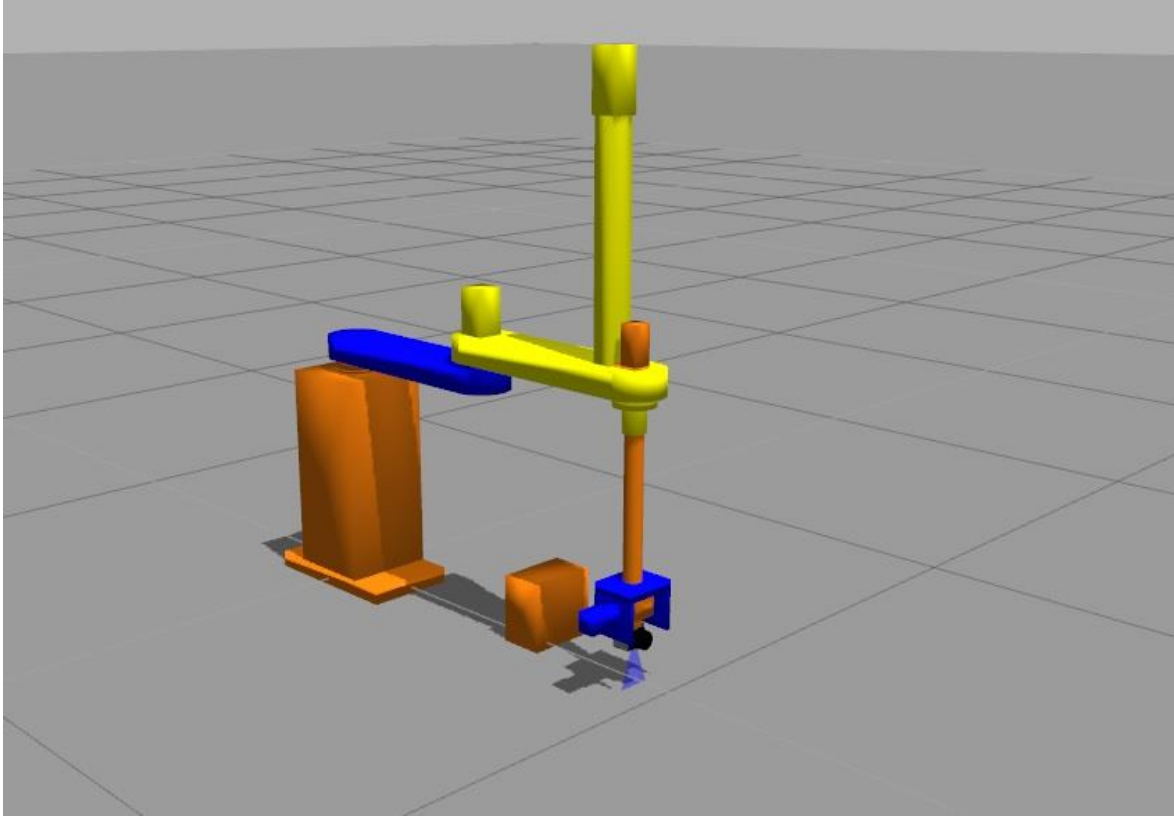
Çarpım sonucu elde edilen matris denklem (5.18)’de gösterilmiştir. Daha sonra elde edilen  $X_{ref}$ ,  $Y_{ref}$ ,  $Z_{ref}$  noktaları “myrobo\_pcd.pcd” dosyasına sonradan da kullanılmak üzere kaydedilir. Ayrıca bu noktalar ekrana basılarak taranmış cismin görüntüsü elde edilir.



**Şekil 5.21.** Lazer verilerinin referans koordinat sistemine göre XYZ noktalarının belirlenmesi.

## 6. UYGULAMA

Oluřturulan yörünge planlaması ve benzetim sistemi boyutları 100x160x170 mm olan dikdörtgenler prizması bir cismin 3 boyutlu görüntüsünü oluşturmak için uygulanmıştır. Kübik cismin ve robotun resmi Şekil 6.1’de gösterilmiştir. Yörünge planlaması sonucu robotun uç noktasının izlemesi istenen yörünge Şekil 6.2’deki mavi çizgilerle gösterilmiştir. Bu sebeple yörünge planlamasında kullanılmak üzere Şekil 6.2’deki kırmızı çerçeveli noktalar başlangıç noktası olan 1 numaralı çerçeveden başlayarak sırası ile parametreler dosyasına (Bkz. Bölüm 5.1) kaydedilmiştir. Tablo 6.1’de bu noktalar verilmektedir.

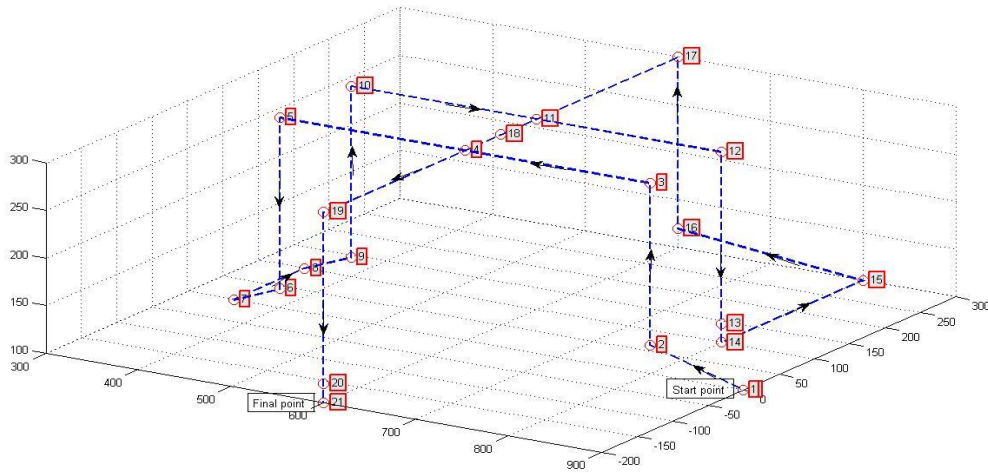


**Şekil 6.1.** Kübik cismin ve robotun gazebo benzetim ortamı görüntüsü.

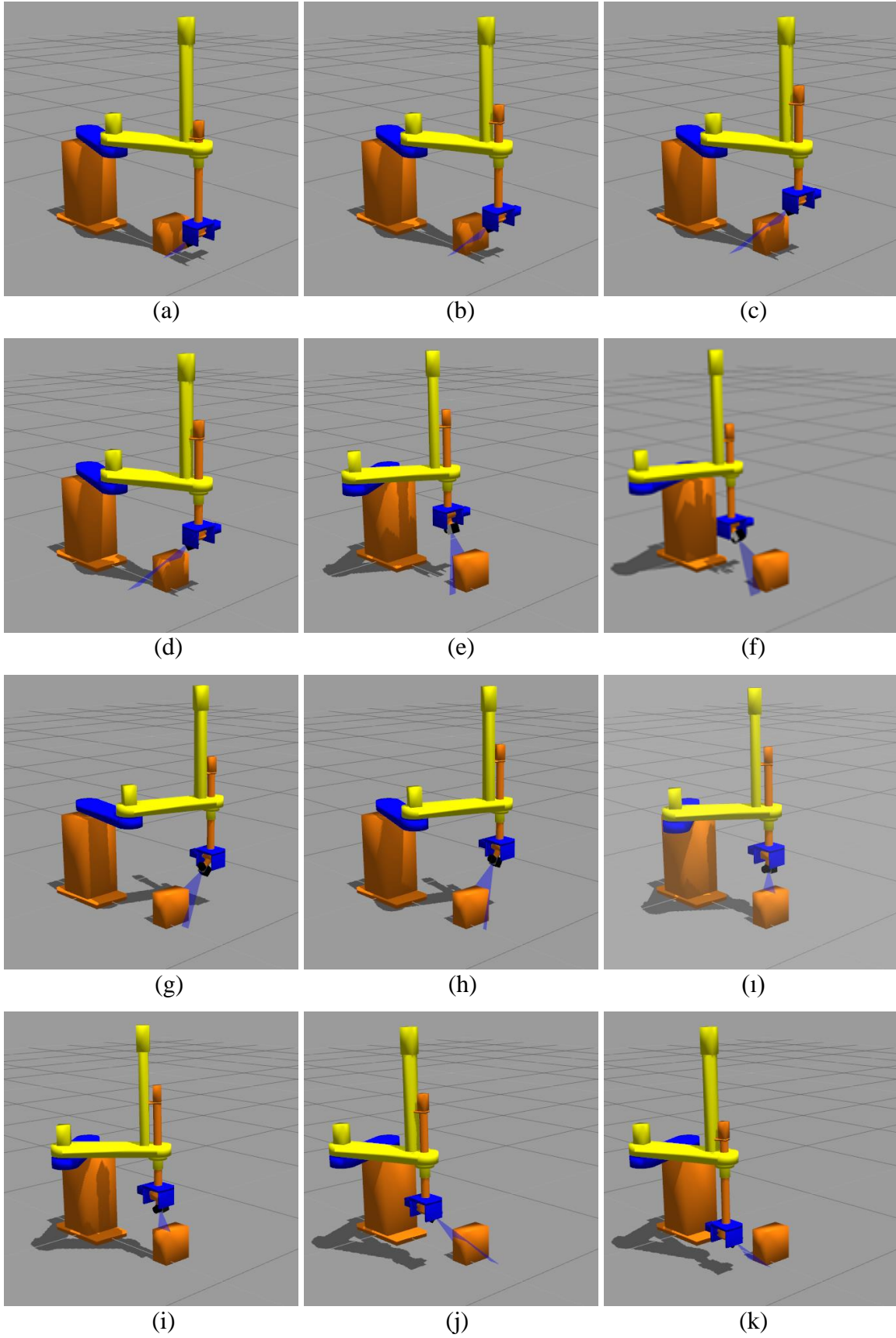


Çizelge 6.1. Yörünge Planlamasında Kullanılacak Parametreler.

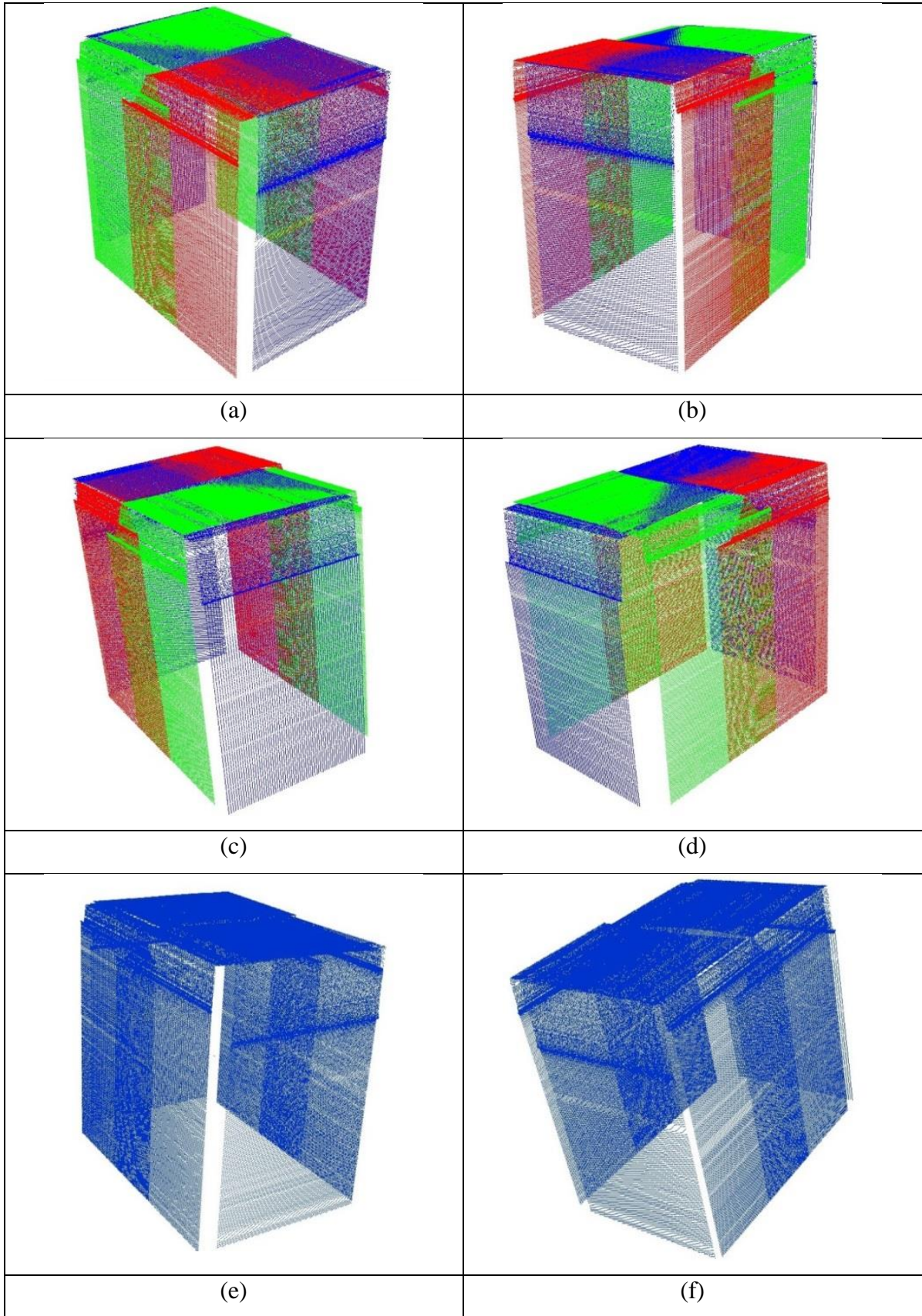
SIRA	X Noktası	Y Noktası	Z Noktası	X Açısı	Y Açısı	Z Açısı
1	900	0	100	0	0	0
2	900	0	100	0	0	90
3	900	0	140	0	0	180
4	800	0	140	0	45	180
5	800	0	300	0	45	180
6	600	0	300	0	0	180
7	400	0	300	0	-45	180
8	400	0	140	0	-45	180
9	350	0	100	0	0	180
10	350	100	100	0	0	0
11	400	100	140	0	45	0
12	400	100	300	0	45	0
13	600	100	300	0	0	0
14	800	100	300	0	-45	0
15	800	100	140	0	-45	0
16	800	100	100	0	0	0
17	800	270	100	0	0	-90
18	600	270	140	0	45	-90
19	600	270	300	0	45	-90
20	600	50	300	0	0	-90
21	600	-170	300	0	-45	-90
22	600	-170	140	0	-45	-90
23	600	-170	100	0	0	-90



Şekil 6.2. Cismin taranması için izlenmesi beklenen yörünge.



Şekil 6.3. Tarama esnasından görüntüler (a-k).



Şekil 6.4. Tarama sonrası elde edilen görüntü (Farklı Açılardan) (a-f).

Tarama işlemi esnasında benzetim programından alınan çerçeve resimler Şekil 6.3'te gösterilmiştir. Tarama işlemi esnasında cismin yan yüzeyleri  $45^0$  açıyla, üst yüzeyi ise kenarlardan orta noktaya doğru  $45^0$ 'den  $90^0$ 'ye doğru olacak şekilde taranmıştır. Şekil 6.4'de ise tarama sonrası 3B Görüntüleme programından elde edilen görüntüler gösterilmiştir. Görüntülerin net anlaşılması için farklı yüzeylerden yapılan tarama verileri farklı renklerde gösterilmiştir.

Görüntüde verilen şeklin üst kısımlarında hafif atlamalar görülmektedir. Bunun nedeni, tarama sırasında lazer algılayıcılardan alınan veriler ile robotun uç nokta koordinatlarının eş zamanlı kaydedilmesindeki güçlükten kaynaklanmaktadır. Bu problemin giderilmesi için çalışma yapma ihtiyacı bulunmaktadır.

## 7. SONUÇ VE ÖNERİLER

Bu çalışmada, genellikle 4 eksenli olarak bulunabilen SCARA tipi bir robotun, özel tasarım 5 eksenli formu için tasarım ve uygulamalar yapılmıştır. Söz konusu robot platformu, sanayide döküm parçaları üzerindeki çapakların temizlenmesinde kullanılmak üzere Robotes adlı bir firma tarafından tasarlanmış ve prototip üretimi gerçekleştirilmiştir. Bu robot platformu için, yörünge planlama gerçekleştirmek üzere algoritma geliştirilmiştir ve programlanmıştır. Ayrıca, cisimlerin yüzeylerinden veri toplamak ve 3B modelini elde etmek üzere bir benzetim ortamı tasarlanmıştır.

Çalışmada geliştirilen yörünge planlaması, doğrusal ve ara nokta üzerinden olmak üzere iki şekilde yapılmıştır. Planlama algoritması, hem MATLAB komutları kodları hem de C/C++ dilinde programlanmış ve testler gerçekleştirilmiştir. Ayrıca, C/C++ dilinde yazılan yörünge planlama algoritma kodları ve kinematik denklemlerin kodlamaları gerçek robot prototipi üzerinde denenmiş ve başarılı bir şekilde çalıştığı görülmüştür.

Ayrıca, geliştirilen hareket planlama algoritmalarının kullanıldığı bir 3B benzetim ortamı geliştirilmiştir. Bu ortamın tasarımında, robotik çalışmalarında yaygın olarak kullanılan ROS platformu kullanılmıştır. Verilen noktalar üzerinden hareket ederek, robot uç noktasına bağlanmış lazer profil algılayıcı cisimler etrafında dolaştırılmakta ve cisimlerin yüzeyinden 3B veriler toplanmaktadır. Bu veriler, 3B modelinin elde edilmesi ve gösterilmesinde kullanılmaktadır. Bu amaçla, bir sistem tasarımı ve gerçekleştirilmesi yapılmıştır. Bu sistem, son zamanlarda önem kazanan ve yoğun çalışmalar yapılan robotik 3B modelleme yöntemlerinin geliştirme safhasında yöntemlerin test edilmesi amacıyla kullanılabilir.

Gelecekte tasarımı yapılan sistemin fonksiyonları arttırılabilir. Sistemde bulunan 3B Benzetim modülü yerini alacak gerçek robot denetleyicisi programı geliştirilerek, sistemin gerçek robot üzerinde daha etkin çalışması sağlanabilir. Ayrıca, kullanıcıların daha kolay bir şekilde sistemi kullanabilmesi için grafik kullanıcı arayüz yazılabilir.

## KAYNAKLAR DİZİNİ

Anonim1, 11.05.2011, urdf / XML / link, <http://wiki.ros.org/urdf/XML/link>, 21.10.2013.

Anonim2, 02.06.2014, Rviz, <http://wiki.ros.org/rviz>, 21.10.2013.

Anonim3, 09.05.2013, Documentation, <http://wiki.ros.org/>, 21.10.2013.

Anonim4, 16.06.2015, Eigen, [http://eigen.tuxfamily.org/index.php?title=Main\\_Page](http://eigen.tuxfamily.org/index.php?title=Main_Page), 21.10.2013.

Anonim5, 26.01.2015, Gazebo, <http://gazebosim.org/>, 21.10.2013.

Asakawa, N., Toda, K., Takeuchi, Y., 2002, Automation of chamfering by an industrial robot; for the case of hole on free-curved surface, *Robotics and Computer Integrated Manufacturing* 18, 379–385.

Aurich, J.C., Dornfeld, D., Arrazola, P.J., Franke, V., Leitz, L., Min,S., 2009, Burrs - Analysis, control and removal, *CIRP Annals - Manufacturing Technology* 58, 519–542.

Bonilla, I., Mendoza, M., González-Galván, E.J., Chávez-Olivares, C., Flores, A.L., Reyes, F., 2012, Path-Tracking Maneuvers With Industrial Robot Manipulators Using Uncalibrated Vision and Impedance Control, *IEEE Transactions On Systems Man and Cybernetics —Part C: Applications and Reviews* Vol. 42 No. 6, 1716-1729.

Chan, C., Bradley, C., Vickers, G.W., 2001, A multi-sensor approach to automating co-ordinate measuring machine-based reverse engineering, *Computers in Industry* 44, 105-115.

Buckmaster, D.J., Newman, W.S., Some, S.D., 2008, Compliant Motion Control for Robust Robotic Surface Finishing, *Proceedings of the 7th World Congress on Intelligent Control and Automation* June 25 - 27 Chongqing-China; 559-564.

Coleman, D., 05.06.2013, urdf / XML / joint, <http://wiki.ros.org/urdf/XML/joint>, 21.10.2013.

Coleman, D., 17.10.2013, urdf / XML, <http://wiki.ros.org/urdf/XML>, 21.10.2013.

**KAYNAKLAR DİZİNİ (devam ediyor)**

- Dumitras, C.G, 2007, An Approach on Micro-Cutting (Deburring) Process, Proceedings of the World Congress on Engineering 2007 Vol II WCE July 2 - 4 London-U.K, 6 p.
- Gasparetto, A., Boscariol, P., Lanzutti, A., Vidoni, R., 2012, Trajectory Planning in Robotics, *Math.Comput.Sci.* 6, 269–279.
- Graaf, M., Aarts, R., Jonker, B., Meijer, J., 2010, Real-time seam tracking for robotic laser welding using trajectory-based control, *Control Engineering Practice* 18, 944–953.
- Guigue, A., Ahmadi, M., Langlois, R., John, M., Hayes, D., 2010, Pareto Optimality and Multiobjective Trajectory Planning for a 7-DOF Redundant Manipulator, *IEEE Transaction On Robotics* Vol. 26 No. 6, 1094-1099.
- Hsu, F.Y., Fu, L.C., 2000, Intelligent Robot Deburring Using Adaptive Fuzzy Hybrid Position/Force Control, *IEEE Transactions On Robotics and Automation* Vol. 16, No. 4, 325-335.
- Jayaweera, N., Webb, P., 2011, Robotic edge profiling of complex components, *Industrial Robot: An International Journal* Vol. 38 Iss: 1, 38 - 47.
- Jeon, J.W., Park, S., Kim, D., Kim, S., 1993, An Efficient Trajectory Generation for Industrial Robots, ISBN 0-7803-1462-x/93\$03.00 IEEE, 137-2143.
- Jetchev, N., Toussaint, M., 2013, Fast motion planning from experience: trajectory prediction for speeding up movement generation, *Auton Robot* 34, 111–127.
- Jinno, M., Uenohara, M., Oaki, J., Tatsuno, K., 1999, Teaching-less Robot System for Finishing Workpieces of Various Shapes Using Force Control and Computer Vision, Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 573-578.
- Kazerooni, H., 1998, Automated Robotic Deburring Using Impedance Control, Presented at the 1987 Conference on Robotics and Automation Raleigh North Carolina, March 31-April 3, 21-25.
- Kröger, T., Wahl, F.M., 2010, Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events, *IEEE Transactions On Robotics* Vol. 26, No. 1, 94-111.



**KAYNAKLAR DİZİNİ (devam ediyor)**

- Lee, S.H., Li, C.J., Kim, D.H., Kyung, J.H., Han, C.S., 2009, The Direct Teaching and Playback Method for Robotic Deburring System Using the Adaptive-Force-Control, IEEE International Symposium on Assembly and Manufacturing 17-20 November Suwon-Korea, 235-241.
- Li, C.J., Park, C.H., Kyung, J.H., Chung, G.J., Han, C.S., 2011, Study on Teaching Path Reconstruction Algorithm based Direct teaching and Playback Method, The 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2011) Nov. 23-26 Incheon-Korea, 778-779.
- Liao, L., Xi, F., Liu, K., 2008, Modeling and control of automated polishing/deburring process using a dual-purpose compliant toolhead, International Journal of Machine Tools & Manufacture 48, 1454–1463.
- Mohammad, M., Babriya, V., Sobh, T., 2008, Modeling a Deburring Process, Using DELMIA V5®, Journal of Engineering and Applied Sciences 3 (11), 835-847.
- Murphy, K.M., Norcross, R.J., Proctor, F.M., 1998, Cad Directed Robotic Deburring, Robot Systems Division National Bureau of Standards, 8 p.
- Ramabalan, S., Saravanan, R., Balamurugan, C., 2009, Multi-objective dynamic optimal trajectory planning of robot manipulators in the presence of obstacles, Int J Adv Manuf Technol, 580–594.
- Rejc, J., Cinkelj, J., Munih, M., 2009, Dimensional measurements of a gray - iron object using a robot and a laser displacement sensor, Robotics and Computer-Integrated Manufacturing 25, 155–167.
- Rososhansky, M., Xi, F., 2011, Coverage based tool-path planning for automated polishing using contact mechanics theory, Journal of Manufacturing Systems 30, 144–153.
- Santos, R.R., Steffen, V., Saramago, S.F.P., 2008, Robot path planning in a constrained workspace by using optimal control techniques, Multibody Syst Dyn 19, 159–177.
- Song, H.C., Kim, B.S., Song, J.B., 2012, Tool Path Generation based on Matching between Teaching Points and CAD Model for Robotic Deburring, The 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics July 11-14 Kaohsiung-Taiwan, 890-895.



**KAYNAKLAR DİZİNİ (devam ediyor)**

- Vakanski, A., Mantegh, I., Irish, A., Janabi-Sharifi, F., 2012, Trajectory Learning for Robot Programming by Demonstration Using Hidden Markov Model and Dynamic Time Warping, *IEEE Transactions On Systems Man and Cybernetics —Part C: Cybernetics* Vol. 42 No. 4, 1039-1052.
- Wurll, C., Henrich, D., 2001, Point-to-Point and Multi-Goal Path Planning for Industrial Robots, Special Issue on "Motion Planning" of the *Journal of Robotic Systems*, 17 p.
- Zhang, H., Chen. H., Xi, N., Zhang, G., He, J., 2006, On-Line Path Generation for Robotic Deburring of Cast Aluminum Wheels, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* October 9 - 15 Beijing - China, 2400-2405.
- Zheng, C., Su, Y., Müller, P.C., 2009, Simple online smooth trajectory generations for industrial systems, *Mechatronics* 19, 571–576.
- Ziliani, G., Visoli, A., Legnani, G., 2009, A mechatronic approach for robotic deburring, *Mechatronics* 17, 431–441.