

5 Eksenli apak Alma Robotunun Kontrolü

Hatice Aydın

**YÜKSEK LİSANS TEZİ**

Elektrik Elektronik Mühendisliđi Anabilim Dalı

Ađustos 2018

Control of 5 Axis Deburring Robot

Hatice Aydın

**MASTER OF SCIENCE THESIS**

Department of Electrical and Electronics Engineering

August 2018

# 5 Eksenli apak Alma Robotunun Kontrolü

Hatice Aydın

Eskişehir Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliğı Uyarınca  
Elektrik Elektronik Mühendisliğı Anabilim Dalı  
Kontrol ve Kumanda Sistemleri Bilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır

Danışman: Prof. Dr. Osman Parlaktuna

Bu tez Eskişehir Osmangazi Üniversitesi Bilimsel Araştırma Projeleri Komisyonu tarafından 201615A123 2016-1367 numaralı proje çerçevesinde desteklenmiştir.

Ağustos 2018

## ONAY

Elektrik Elektronik Mühendisliđi Anabilim Dalı Yüksek Lisans öđrencisi Hatice Aydın'ın YÜKSEK LİSANS tezi olarak hazırladıđı “5 Eksenli Çapak Alma Robotunun Kontrolü” başlıklı bu çalıřma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek oybirliđi ile kabul edilmiřtir.

**Danıřman** : Prof. Dr. Osman Parlaktuna

**İkinci Danıřman** : -

### **Yüksek Lisans Tez Savunma Jürisi:**

**Üye** : Prof. Dr. Osman Parlaktuna

**Üye** : Doç. Dr. Metin Özkan

**Üye** : Dr. Öğr. Üyesi Hakan Korul

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

Prof. Dr. Hürriyet ERŐAHAN  
Enstitü Müdürü

## ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Prof. Dr. Osman Parlaktuna danışmanlığında hazırlamış olduğum “5 Eksenli Çapak Alma Robotunun Kontrolü” 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. 13/08/2018

Hatice Aydın

İmza

## ÖZET

Sanayide, döküm parçaları üretiminde oluşan çapakların müşteriye teslim edilmeden önce temizlenmesi gerekmektedir. Çapakların temizlenebilmesi için insan kullanımı bazı parçaların üretiminde hem maliyeti arttırmaktadır hem de parça üretim işlemi daha uzun sürmektedir. Endüstriyel kullanımlar için geliştirilmiş bir robot kolunun ucuna, çapakları temizleyecek bir uç elemanı takılarak çapak alma robotuna dönüştürülebilmektedir. Ancak bu tür bir yaklaşımda çapakların yerlerinin bilindiği kabul edilmekte, robot kolunun uç elemanının belirli bir yörünge izlerken ızgara üzerine kuvvet uygulayarak çapakları alması sağlanmaktadır. Bu çalışmada, set üstü ocaklar için üretilen döküm ızgaraların üzerindeki çapakları almak üzere tasarlanan 5 eksenli SCARA (Selective Compliance Assembly Robot Arm) tipi robot kolu için, döküm parçasının CAD (Computer Aided Design) çizimlerinden aldığı verileri kullanarak parçanın etrafında dolaşacak şekilde otomatik yörünge yaratılması hedeflenmiştir. Robotun uç elemanının izleyeceği konum ve köşe bilgisi (X, Y ve Z) dünya koordinat sistemine göre hesaplanmış ve geliştirilen grafiksel kullanıcı ara yüzü ile robot kolunun uç elemanının izleyeceği yörünge görselleştirilmiştir. Yaratılan yörüngenin robot kolunun üzerinde uygulanabilir olabilmesi için, robot için gerekli malzeme tespiti yapılmıştır. Alınan malzemelere göre, robot pano bağlantıları yapılmış, sürücü parametreleri belirlenmiş ve her bir eklem için hız, ivme, pozisyon, PID değişkenleri belirlenmiştir. Son olarak, 5 eksenli SCARA tipi çapak alma robotunun kontrolü sağlanmıştır.

**Anahtar Kelimeler:** Robot, 5 eksenli robot kolu, CAD, yörünge, çapak alma.

## SUMMARY

In the industry, burrs formed in the production of casts must be cleaned before delivery to the customer. In order to clean the burrs, human use increases both the cost of production of some parts and the production of parts takes longer. A robot developed for industrial use can be transformed into a deburring robot by attaching an end effector to clean the burrs. However, in such an approach, it is assumed that the locations of the burrs are known, and end effector of the robot is able to take the burrs by applying a force on the grill while tracking a certain trajectory. In this study, for the 5-axis SCARA (Selective Compliance Assembly Robot Arm) type robot arm designed to take the burrs on the cast grill produced for the countertop stove is aimed to create an automatic trajectory around the part using the data obtained from CAD (Computer Aided Design) drawings of the casting part. The position and vertex information (X, Y, and Z) of end effector of the robot is calculated according to the world coordinate system and the trajectory of the end effector of the robot arm is visualized by the developed graphical user interface. In order for the created trajectory to be applicable on the robot arm, the material necessary for the robot is determined. According to purchased materials, robot panel connections are made, drive parameters are determined and velocity, acceleration, position, PID variables for each joint were determined. Finally, the 5 axis SCARA type deburring robot is controlled.

**Keywords:** Robot, 5 axis robot arm, CAD, trajectory, deburring.

## 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 Prof. Dr. Osman Parlaktuna'ya sonsuz teşekkürlerimi sunarım.

Tez çalışması için beni A1 nolu proje çerçevesinde destekleyen Eskişehir Osmangazi Üniversitesi Bilimsel Araştırma Projeleri Komisyonuna sonsuz teşekkür ederim.

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

Tez çalışması için geliştirdikleri çapak alma robot kolunun kullanılmasına izin veren Robotes Robotik Çözümler Ltd. firmasına sonsuz teşekkürlerimi sunarım.



## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> .....	vi
<b>SUMMARY</b> .....	vii
<b>TEŞEKKÜR</b> .....	viii
<b>İÇİNDEKİLER</b> .....	ix
<b>ŞEKİLLER DİZİNİ</b> .....	x
<b>ÇİZELGELER DİZİNİ</b> .....	xiv
<b>SİMGELER VE KISALTMALAR DİZİNİ</b> .....	xv
<b>1. GİRİŞ VE AMAÇ</b> .....	1
<b>2. LİTERATÜR ARAŞTIRMASI</b> .....	4
<b>3. TEORİK BİLGİ</b> .....	7
3.1. DWG Uzantılı Dosya Formatı .....	7
3.2. DXF Uzantılı Dosya Formatı .....	7
3.3. STL Uzantılı Dosya Formatı .....	13
<b>4. MATERYAL VE YÖNTEM</b> .....	15
4.1. DXF Dosya Formatının Analizi .....	15
4.1.1. Koordinat Sistemi .....	15
4.1.2. Çizgi değişkenlerinin belirlenmesi .....	18
4.1.3. Yay değişkenlerinin belirlenmesi .....	20
4.1.4. Daire değişkenlerinin belirlenmesi .....	23
4.1.5. Elips değişkenlerinin belirlenmesi .....	25
4.2. STL Dosya Formatının Analizi .....	33
<b>5. BULGULAR VE TARTIŞMA</b> .....	41
5.1. Donanım .....	41
5.1.1. Servo sürücü bağlantıları .....	43
5.1.2. Hareket kontrol kartı bağlantıları .....	47
5.2. Sürücü Değişkenleri .....	54
5.3. Robot Üzerinde Hız Kontrolünün Uygulanması .....	56
<b>6. SONUÇ VE ÖNERİLER</b> .....	70
<b>KAYNAKLAR DİZİNİ</b> .....	72

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
1.1. Autocad programında örnek ızgara çizimi .....	1
1.2. Çapaklı ızgara örnekleri .....	1
1.3. Robotes firmasına ait 5 eksenli SCARA çapak alma robotu Solidworks çizimi .....	2
1.4. Çalışmanın akış şeması .....	3
3.1. Bir çizim dosyasında DXF dosyasına ait etiketli verinin bir kısmı .....	8
3.2. DXF dosya formatının bölüm haritası .....	12
3.3. Cisme ait yüzey ve köşe ilişkisi .....	14
4.1. Kartezyen Koordinat Sistemi .....	16
4.2. Sağ-el kuralı .....	17
4.3. Çizgi değişkenlerinin DXF dosyasından okunmasını gösteren Matlab kodu .....	18
4.4. Matlab ortamında çizgi denklemlerinin kodlanması .....	19
4.5. Autocad'te 3 boyutlu çizgilerden oluşturulmuş cisim .....	19
4.6. Şekil 4.5.'in Matlab'ta çizimi .....	20
4.7. Yay değişkenlerinin DXF dosyasından okunmasını gösteren Matlab kodu .....	21
4.8. Matlab ortamında yay denklemlerinin kodlanması .....	22
4.9. Autocad'te 3 boyutlu yaylardan oluşturulmuş cisim .....	22
4.10. Şekil 4.9.'un Matlab'ta çizimi .....	23
4.11. Daire değişkenlerinin DXF dosyasından okunmasını gösteren Matlab kodu .....	23
4.12. Matlab ortamında daire denklemlerinin kodlanması .....	24
4.13. Autocad'te 3 boyutlu dairelerden oluşturulmuş cisim .....	24
4.14. Şekil 4.13.'ün Matlab'ta çizimi .....	25

## ŞEKİLLER DİZİNİ (devam)

<u>Sekil</u>	<u>Sayfa</u>
4.15. Elips değişkenlerinin DXF dosyasından okunmasını gösteren Matlab kodu .....	26
4.16. Matlab ortamında elips denklemlerinin kodlanması .....	27
4.17. X eksen yönündeki dönüşüm matrisi .....	27
4.18. Y eksen yönündeki dönüşüm matrisi .....	27
4.19. Z eksen yönündeki dönüşüm matrisi .....	28
4.20. Autocad'te 3 boyutlu elipslerden oluşturulmuş cisim .....	28
4.21. Şekil 4.20.'nin Matlab'ta çizimi .....	28
4.22. DXF dosya formatı analizinde kullanılan fonksiyonların birbiriyle ilişkisi .....	29
4.23. Autocad'de çizilen ocak üstü ızgara çizimi .....	30
4.24. Matlab'da çizilen ocak üstü ızgara çizimi .....	30
4.25. Autocad'de DXF dosya formatında bir makina çizimi .....	31
4.26. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge 1 .....	32
4.27. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge 2 .....	32
4.28. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge X, Y ve Z noktalarının başlangıç kısmı .....	33
4.29. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge doğruluk testi 1 .....	34
4.30. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge doğruluk testi 2 .....	34
4.31. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge doğruluk testi 3 .....	35
4.32. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge doğruluk testi 4 .....	35

## ŞEKİLLER DİZİNİ (devam)

<u>Sekil</u>	<u>Sayfa</u>
4.33. Sol tarafta Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge ve sağ tarafta uç elemanının izleyeceği yörünge köşe noktaları .....	36
4.34. Sol tarafta Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge X, Y ve Z noktalarının başlangıç kısmı, sağ tarafta bu yörünge köşe noktalarının başlangıç kısmı .....	37
4.35. Matlab'ta yerleşim düzenleyicisi aracılığıyla tasarlanan grafiksel ara yüz 1 .....	38
4.36. Matlab'ta yerleşim düzenleyicisi aracılığıyla tasarlanan grafiksel ara yüz 2 .....	39
4.37. Tasarlanan ara yüz aracılığıyla yörünge yaratılması .....	40
5.1. 5 eksenli çapak alma robot kolu .....	42
5.2. Pano içerik tanımı .....	43
5.3. Sürücü bağlantı şeması .....	43
5.4. 36 pin MDR konektörün pin dizilişi .....	44
5.5. Frenleme pin bağlantısı .....	45
5.6. 20 pin MDR konektörün pin dizilişi .....	46
5.7. Network Port .....	46
5.8. CN3 portu ve bilgisayar arasındaki bağlantı şeması .....	47
5.9. Sürücü ve motor arasındaki güç bağlantısı .....	47
5.10. Hareket kontrol kartı MC508 .....	48
5.11. 5 kanallı konektör ve özellikleri .....	48
5.12. I/O konektör 1 özellikler .....	49
5.13. I/O konektör 4 özellikleri .....	49
5.14. MDR20 konektörü pin ve özellikleri .....	50
5.15. 5 eksenli çapak alma robotunun tek eklem için bağlantı şeması .....	51

**ŞEKİLLER DİZİNİ (devam)**

<b><u>Sekil</u></b>	<b><u>Sayfa</u></b>
5.16. 5 eksenli çapak alma robotunun kontrol pano elemanları .....	52
5.17. 1., 2., 3. ve 4. eklem için klemens bağlantıları .....	53
5.18. 5. eklem için klemens bağlantıları .....	53
5.19. Sürücü parametreleri .....	55
5.20. IP adres ve IP port .....	56
5.21. Birinci eklem ev konumu .....	58
5.22. Birinci eklem saat yönünde 20 derece dönüşü .....	58
5.23. Birinci eklem saat yönünde 30 derece dönüşü .....	59
5.24. İkinci eklem ev konumu .....	60
5.25. İkinci eklem saat yönünde 20 derece dönüşü .....	60
5.26. İkinci eklem saat yönünde 30 derece dönüşü .....	61
5.27. Üçüncü eklem ev konumu .....	62
5.28. Üçüncü eklem -120 mm yol alışı .....	62
5.29. Üçüncü eklem -150 mm yol alışı .....	63
5.30. Dördüncü eklem ev konumu .....	64
5.31. Dördüncü eklem saat yönünde 2 derece dönüşü .....	64
5.32. Dördüncü eklem saat yönünde 3 derece dönüşü .....	65
5.33. Beşinci eklem kontrol programı .....	66
5.34. Beşinci eklem hareketi sırasındaki eksen değişkenleri .....	67
5.35. Eklemlerin ev konumu .....	68
5.36. Eklemlerin bir arada hareketi ve eksen parametreleri .....	69

**ÇİZELGELER DİZİNİ**

<b><u>Çizelge</u></b>	<b><u>Sayfa</u></b>
3.1. Yüzey Listesi .....	14
3.2. Köşe Listesi .....	14
5.1. Sürücü üzerinde yer alan portlar ve özellikleri .....	44
5.2. CN1 portundaki pinlerin numaraları ve görevleri .....	45
5.3. CN2 portundaki pinlerin numaraları ve görevleri .....	46

## SİMGELER VE KISALTMALAR DİZİNİ

<b><u>Simgeler</u></b>	<b><u>Açıklama</u></b>
X	Kartezyen koordinat sisteminde X eksen
Y	Kartezyen koordinat sisteminde Y eksen
Z	Kartezyen koordinat sisteminde Z eksen
N	Normal vektörü
N <sub>x</sub>	Normal vektörün X eksen
N <sub>y</sub>	Normal vektörün Y eksen
W <sub>y</sub>	Dünya koordinat sisteminde Y eksen
W <sub>z</sub>	Dünya koordinat sisteminde Z eksen
A <sub>x</sub>	Rastgele X eksen
A <sub>y</sub>	Rastgele Y eksen
A <sub>z</sub>	Rastgele Z eksen
x <sub>1</sub>	Başlangıç x noktası
y <sub>1</sub>	Başlangıç y noktası
z <sub>1</sub>	Başlangıç z noktası
x <sub>2</sub>	Bitiş x noktası
y <sub>2</sub>	Bitiş y noktası
z <sub>2</sub>	Bitiş z noktası

<b><u>Kısaltmalar</u></b>	<b><u>Açıklama</u></b>
SCARA	Seçilebilir uyumlu montaj robot kolu
CAD	Bilgisayar destekli tasarım
DWG	Çizim dosya uzantısı
DXF	Çizim ara yüz dosya uzantısı
STL	Stereolitografi dosya uzantısı
TXT	Metin dosya uzantısı
PNG	Taşınabilir ağ grafiği dosya uzantısı
XML	Genişletilebilir işaretleme dili

**SİMGELER VE KISALTMALAR DİZİNİ (devam)****Kısaltmalar****Açıklama**

ADACS

Gelişmiş çapak alma ve pah kırma sistemi

ASCII

Bilgi değişimi için Amerikan standart kodlama sistemi

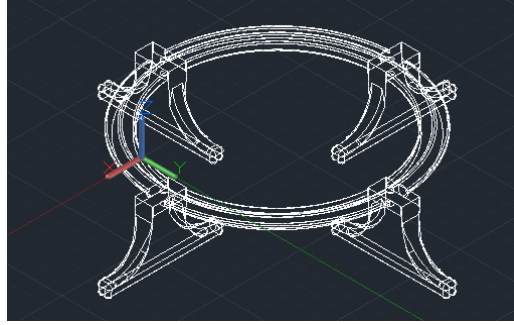
GUIDE

Grafiksel kullanıcı ara yüz tasarım ortamı

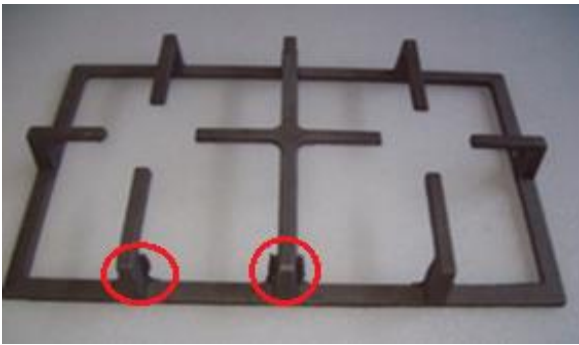


## 1. GİRİŞ VE AMAÇ

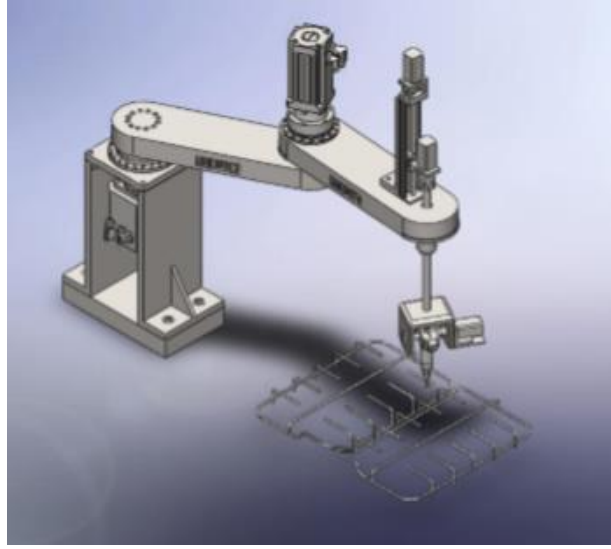
Çapak alma problemi döküm imalatında önemlidir. Birçok firma endüstride çapak alma işlemini işçiler vasıtasıyla yapmaktadır. Ancak günümüzde bu işlemler endüstriyel robot kolları tarafından otomatik olarak yapılmaktadır. Bu çalışmada, Şekil 1.1.'de Autocad programında örnek çizimi görünen ocak üstü ızgara üretimi yapan firmaların çapak alma ihtiyaçları (Şekil 1.2.) doğrultusunda ortaya çıkan gereksinimlerine uygun şekilde tasarlanmış olan Şekil 1.3.'de Solidworks programında çizimi görünen Robotex firmasına ait 5 eksenli SCARA (Selective Compliance Assembly Robot Arm) çapak alma robot kolunun Autocad programında oluşturulmuş CAD çizimlerinden robot kolunun uç elemanının izleyeceği otomatik bir yörünge yaratılması hedeflenmiştir.



Şekil 1.1. Autocad programında örnek ızgara çizimi



Şekil 1.2. Çapaklı ızgara örnekleri



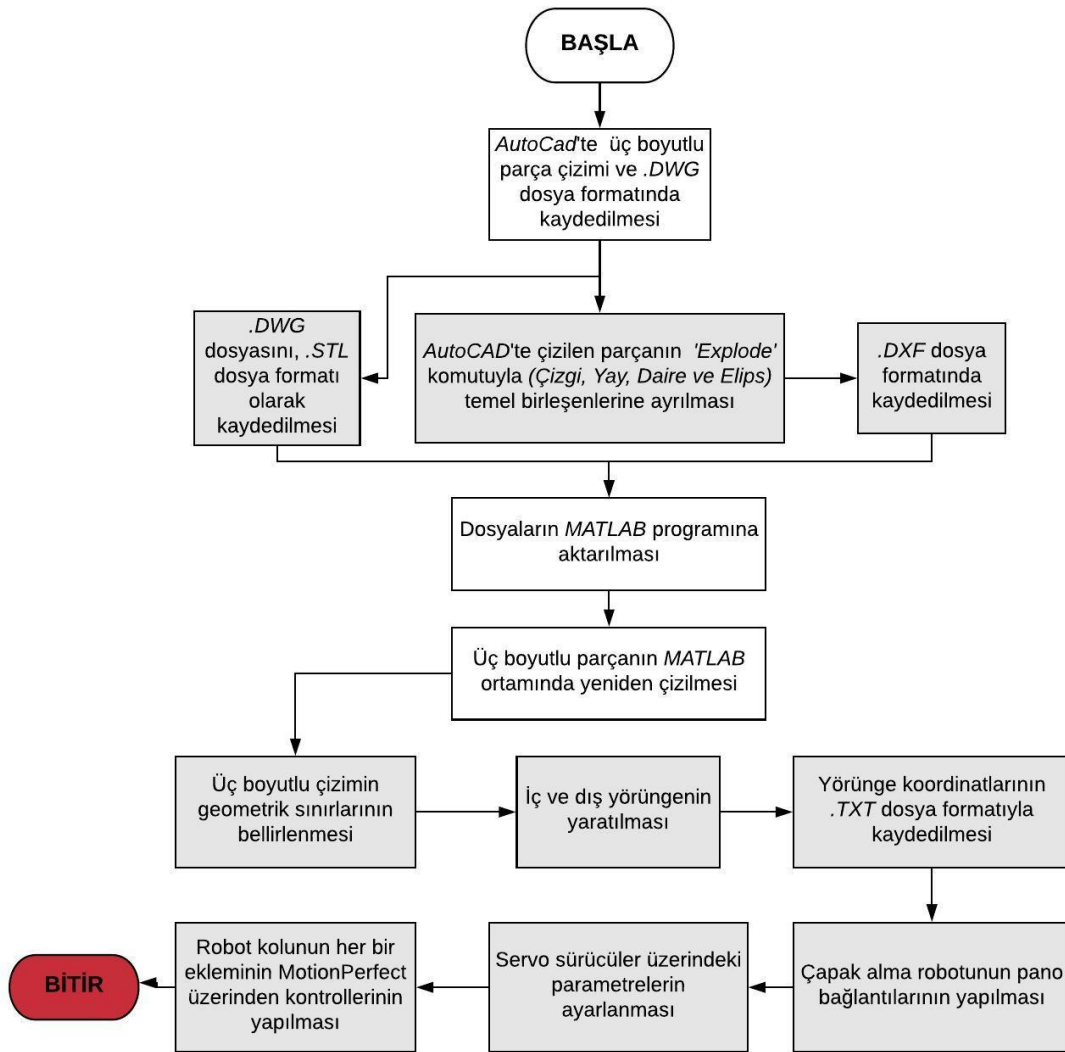
Şekil 1.3. Robotes firmasına ait 5 eksenli SCARA çapak alma robotu Solidworks çizimi

Öncelikle, Autocad programında çizilen üç boyutlu parçaların DWG (**DraWinG**) uzantılı dosya formatından, çizim verilerini elde etmek için kullanılacak olan DXF (**Draw Exchange File**) uzantılı dosya formatına ve STL (**STereoLithography**) uzantılı dosya formatına dönüştürülmelidir. Matlab programı, TXT (**TEXT**), PNG (**Portable Newtwork Graphics**) ve XML (**Extensible Markup Language**) uzantılı dosya formatlarını desteklemesine rağmen DXF ve STL uzantılı dosya formatlarını desteklemez. Bu çalışma sırasında geliştirilmiş algoritmalar ile birlikte bu dosya formatları Matlab programında kullanılabilir. Matlab programında, Autocad programında çizilen CAD parçaları bu algoritmalar ile birlikte yeniden çizdirilebilir.

Üç boyutlu parçaların geometrik sınırı, parçanın bir kısmını oluşturan çizim parametreleri (çizgi, daire, yay, elips vb.) ile tanımlanır. Tespit edilen sınır, yörüngenin temelini oluşturur. 5 eksenli SCARA çapak alma robot kolunun uç noktasının çap bilgisiyle beraber robot kolunun uç noktasının izleyeceği yörünge geometrik olarak yaratılarak bir TXT uzantılı dosyaya robot kolunun uç noktasının gitmesi gereken koordinatlar ve başka bir TXT uzantılı dosyaya robot kolunun uç noktasının gitmesi gereken köşe noktalarına ait koordinatlar yazdırılır. Bu koordinatların robot kolunun üzerinde uygulanabilir olabilmesi için robot pano bağlantılarının, robot üzerinde uygulanacak kontrol şekline göre yapılmış olması gerekmektedir. Son olarak, servo sürücü parametreleri kontrol şekline uygun bir şekilde ayarlanarak ve her bir robot eklemi için hız, ivme, pozisyon, PID değişkenleri

belirlenir. Böylelikle, 5 eksenli SCARA tipi çapak alma robotunun kontrolü sağlanmış olur. Bu çalışmanın bütünlüğünü oluşturan, ana adımları gösteren çalışmanın akış şeması Şekil 1.4.'de gösterilmiştir.

Bu proje sırasında geliştirilen algoritmalar ile birlikte, ince detaylara sahip döküm ızgaralarının üzerinde bulunan çapakları temizlenmesi sırasında gerçekleştirilen yüzey takip analizi işleminden farklı olarak çapak alma işlemi en yüksek seviyede gerçekleştirilir. Bu sayede ürünlerin üretim aşaması kısa sürede ve daha verimli şekilde tamamlanarak zaman ve işgücü tasarrufu sağlanır.



Şekil 1.4. Çalışmanın akış şeması

## 2. LİTERATÜR ARAŞTIRMASI

Literatür incelendiği zaman çapak alma, yörünge planlaması ve yaratılması ile ilgili birçok çalışma yapıldığı görülmektedir. Bu çalışmalara göre;

Mohammad (2008), “Modeling a Deburring Process, Using DELMIA V5®” ismiyle yayınlanan makalede, otomatik çapak alma işleminin başarısının, çapak alma aletinin esnekliğine ve tutarlılığına bağlı olduğuna değinilmiştir. Sistem tasarımcılarının genellikle robotu ve çapak alma aracını ayrı noktalarla tanımlanan bir yol boyunca hareket ettirmek için programladıklarından bahsedilmiştir. Bu sebepten ötürü bu yolun, parçanın kendi içindeki değişimler veya parça kenarı ile robotun enterpolasyonu arasındaki tam yol arasındaki farklar nedeniyle, çapağı alınacak yüzeyin şekli veya çevre çizgisi ile tam olarak çakışmayabileceğinden söz edilmiştir. Aksi takdirde sistem tasarımcısının, otomatik olarak hatayı azaltmak için çok fazla fiziksel yakın etiket noktası oluşturmak zorunda kalacağı aktarılmıştır.

Stouffer ve Robert (1995), çapak alma işlemi ile ilgi yaptıkları çalışmalarında robot hatalarını azaltmak için geliştirilmiş CAD tabanlı bir programlama otomasyonu olan ADACS’dan (Gelişmiş Çapak Alma ve Pah Kırma Sistemi) bahsederler. Bu çalışmada, robot programları CAD bölüm verilerinden otomatik olarak geliştirilir ve bir endüstriyel robot koluna bu programlar robot çalışırken aktarılır. Robot kolunun uç elemanı, (TriKinetics Uyarlanabilir Çapak Alma Aleti), iki serbestlik derecesi sağlamak üzere hareket edebilen bir gövdeye monte edilmiştir. Robot kolunun uç elemanının üzerinde bulunan sensörler çapak alma kuvvetlerini ve robot kolunun uç elemanının konumunu, robot programlarında hesaplanan konumuna göre takip eder.

Murphy vd. (1998), “Cad Directed Robotic Deburring” isimli makalede robotun çapak alma yollarını otomatik olarak oluşturmak için CAD geometri verilerini kullanan bir teknikten söz eder. Bu çalışmaya göre, bir grafik arabirimi kullanan bir operatör, kenarları çizilecek olan parçaların kenarlarını, kullanılacak çapak alma aletlerini ve istenen hızları, ilerleme hızlarını ve temas kuvvetlerini belirtir. Çapak alma yolları üretilir ve bir PUMA 760

robotuna gönderilir. Robot çapak alma için iki geçişli bir teknik kullanır. İlk geçişte robot, robot kinematik hataları, takım aşınması ve küçük parçaların yanlış yerleştirilmesini hesaba katmak için çapak alma yol noktalarını düzeltmek için kuvvet geribildirimini kullanır. İkinci geçişte robot, parçanın çapaklarını temizleyerek düzeltilmiş bir yolu izler.

Song vd. (2002), çapak alma çok fazla gürültü, titreşim ve toz içeren bir işleme süreci olduğundan o ortamda çalışan işçilere zararı olabileceğine “Tool Path Generation based on Matching between Teaching Points and CAD Model for Robotic Deburring” isimli makalede değinmişlerdir. Bu çalışmada, insan işçiler yerine çapak alma işlemini gerçekleştirmek için bir robot kolu kullanmak için çok fazla araştırma yapıldığından bahsetmişlerdir. Makalede, CAD modeline dayanan bir takım yolu üretim yöntemi ve doğrudan öğretim yöntemi, iş parçalarının konum ve yönelim hatalarını en aza indirmek için önerilmiştir. Elde edilmesi zor olan iş parçasının konumu ve yönelimi hakkında bilgi alınmadan, çıkartılan takım yolunun CAD modelinden öğretim noktalarına eşleştirilmesiyle en uygun çapak alma yolu oluşturulur ve aşırı temas kuvvetinin uygulanmasını önlemek için empedans kontrolü kullanılır.

Asakawa vd. (2002), tarafından yayınlanan “Automation of Chamfering by an Industrial Robot; for the Case of Hole on Free-Curved Surface” isimli makalede endüstriyel bir robot kullanılarak CAD verileri temelinde serbest eğimli yüzeydeki delik için otomatik pah kırma işleminden söz edilir. Bir pah kırma aleti olarak, bir elektrikli motor tarafından tahrik edilen bir döner çubuk, aletin rastgele seçilmiş bir pozisyonu vermek için altı serbestlik derecesine sahip olan robotun koluna monte edilir. Pah kırma yolundan dönüştürülen robot kontrol komutu doğrudan robota iletilir. Deneysel sonuçlardan, sistemin serbest eğimli yüzeyli bir iş parçasının üzerindeki bir deliğin kenarı boyunca çapak çıkarmak için etkili olduğu bulunmuştur.

Jinno vd. (1999), geleneksel endüstriyel robot sistemlerinde operatörlerin her bir iş parçası türü için ayrı ayrı robot dili programı yazmaları gerektiğinden özellikle eğer her bir iş parçasının farklı bir şekle sahip olması durumunda operatörlere büyük bir iş yükü getirdiğinden bahseder. Yayınladıkları makalede, kuvvet kontrolü ve bilgisayar görüşü kullanarak çeşitli şekil ve kalınlıklardaki iki boyutlu iş parçalarının bitirilmesi için bir robot

sistemi tanımlanır. Bu robot sistemi, iş parçasının CAD verilerine dâhil edilmesi veya operatör tarafından girilmesi için şekil bilgisi gerektirmez. Her iş parçası şekli, parçaların kenarlarına düz çizgi çizer ve iş parçasının görüntü verisinden dairesel yaylar şeklinde elde edilir. Her bir iş parçası için robot dili programı, iş parçası şekli verisinden ve sonlandırma koşulu verilerinden otomatik olarak üretilir. Önerilen yöntemin etkinliği, yeni geliştirilen bir robot sistemi kullanılarak yapılan deneylerle doğrulanmıştır. Sistemde robot, iş parçası standında şekli önceden bilinmeyen bir iş parçasını alır ve otomatik olarak oluşturulmuş robot dili programını kullanarak son işlemlerini gerçekleştirir. Bu yöntem, operatörler üzerindeki programlama ve iş parçası ayarı yükünü azaltan kompakt ve ucuz bir bitirme robot sistemi sunar.

Rososhansky (2010), yüksek lisans tezinde otomatik parlatma takım yolu planlaması için bir yöntem geliştirmiştir. Bu çalışmada, işleme için takım yolu planlaması bir geometri problemi olarak ele alınırken, parlatma için takım yolu planlamasının, parlatma aleti ile parça arasındaki temas etkisinden dolayı bir temas gerilme problemi olarak ele alınması gerektiği gösterilmektedir. Bu yöntemi geliştirmek için temas mekaniği modelleme ve analizi için temas mekaniği uygulanır. Temas alanı, verilen parlatma yolu boyunca birçok nokta için belirlendikten sonra, parlatma sırasında kapsama alanını göstermek için temas alanının bir haritası oluşturulur ve kullanılır. Bu harita daha sonra parlatma için tam bir kaplama sağlayan parlatma yollarını planlamak için kullanılır.

Proctor ve Murphy (1989), endüstrideki robotların genellikle, sıkıcı ve zaman alıcı bir süreç olan önceden öğretim ile programlanmasına alternatif bir yaklaşım olarak, istenen robot koordinatlarını doğrudan, çevrimdışı programlama olarak bilinen bir yöntem kullanarak hesaplamışlardır. Yaptıkları çalışmada, robot kolunun uç elemanının genel konumsal hatasının, esas olarak robot hatalarından kaynaklandığına, ancak parça işleme sırasındaki hataların, büyük parça toleranslarının, takım aşınmasının ve fikstürde parçaların yanlış hizalanmasının da probleme katkıda bulunduğuna değinmişlerdir. Bununla birlikte, çapak alma sürecinin, insan müdahalesi olmadan, kullanılabilir robot yollarının otomatik olarak verilerden üretilme ilkesine dayandığını söylemişlerdir.

### 3. TEORİK BİLGİ

Bu çalışmada, bilgisayar destekli tasarım için Autocad programı kullanılmıştır. Autocad, CAD çizimleri için kullanılan bir yazılım uygulamasıdır. Yazılım hem iki boyutlu hem de üç boyutlu çizimleri desteklemektedir. Autodesk tarafından geliştirilmiş, satılmış ve Aralık 1982'de piyasaya sürülmüştür. Autocad yazılımı, diğer meslek dallarının yanı sıra mimarlar, proje yöneticileri ve mühendisler tarafından çeşitli sektörlerde kullanılmaktadır. (Anonim1, 2016)

Autocad, dosya formatı olarak çizim anlamına gelen DWG uzantısını kullanır. Ayrıca, DXF uzantılı başka bir çizim formatı kullanarak çizim ara değişim biçimi oluşturur ve DWG dosya formatının diğer CAD programları tarafından okunmasını sağlar. (Anonim1, 2016) Autocad, bir çizim editörü olarak kullanılabilir. Ancak, bazı uygulamalarda, diğer programlar Autocad çizimlerini incelemeli veya Autocad ile görüntülenecek, değiştirilecek veya çizilecek çizimler üretmelidir. Bu çalışmada, yörünge yaratımı ve çizim işlemleri süreçlerinde Autocad dosya formatlarını desteklemeyen Matlab programı kullanılacağından, Matlab ortamında, bu dosya formatları incelenmelidir. Bu yüzden, çalışmanın bu bölümünde Matlab ortamında yörünge yaratımı süreçlerinde kullanılacak olan Autocad programına ait DWG, DXF ve STL uzantılı dosya formatlarından bahsedilecektir.

#### 3.1. DWG Uzantılı Dosya Formatı

DWG uzantılı dosya formatı, Autocad programı tarafından kullanılır. İki ya da üç boyutlu çizimleri destekler. Bu çalışmada, üç boyutlu çizilen parçalar DWG uzantılı dosya formatı olarak kaydedilmiş daha sonra DXF ve STL uzantılı dosya formatlarına dönüştürülmüştür.

#### 3.2. DXF Uzantılı Dosya Formatı

DXF uzantılı dosya formatları, Autocad ile diğer programlar arasındaki çizimlerin değiştirilmesini sağlar. DXF dosyaları ASCII (American Standard Code for Information

Interchange) metin tabanında veya ikili sistem formatı olarak iki şekilde de oluşturulabilir. ASCII DXF dosyaları ikili sistem formatından daha yaygındır ve diğer programlarla daha fazla uyumluluğa sahiptir. (Autodesk Inc, 2012)

DXF uzantılı dosya formatı, bir Autocad çizim dosyasında (DWG) bulunan tüm bilgilerin etiketli veri sunumudur. (Şekil 3.1.) Etiketli veri, grup kodu olarak tanımlanan bir tamsayı ile dosyadaki her bir veri ögesinin önüne geçtiği anlamına gelir. Bir grup kodunun değeri, hangi tipte elemanların izlendiğini gösterir. Bu değer ayrıca belirli bir nesne tipi için bir veri elemanının anlamını gösterir. Bir çizim dosyasındaki kullanıcı tarafından belirtilen hemen hemen bütün bilgi DXF formatında gösterilebilir. DXF ile CAD yazılımında oluşturulan veriler renk, tip ve katman özelliklerini de tanımlamak için kullanılır. (Autodesk Inc, 2012) Ayrıca, DXF metin editöründeki bütün bu verileri içeren binlerce komutu ayırmak zordur. Bu çalışmada, DXF'e ait metin editöründeki bilgiler okunur ve parça çizimlerinin şekli, boyutu ve dünya koordinat sistemindeki konumu elde edilir.

```

2358 AcDbBlockEnd
2359 0
2360 ENDSEC
2361 0
2362 SECTION
2363 2
2364 ENTITIES
2365 0
2366 LINE
2367 5
2368 2A1
2369 330
2370 1F
2371 100
2372 AcDbEntity
2373 8
2374 0
2375 100
2376 AcDbLine
2377 10
2378 25.0
2379 20
2380 35.0
2381 30
2382 0.0
2383 11
2384 25.0
2385 21
2386 25.0
2387 31
2388 0.0
2389 210
2390 0.0
2391 220
2392 0.0
2393 230
2394 -1.0
2395 0

```

Şekil 3.1. Bir çizim dosyasında DXF dosyasına ait etiketli verinin bir kısmı



Etiketli verinin her bir bölümü Şekil 3.1.'de gösterildiği gibi grup kodu olan “0” ile başlar ve 0’ı takip eden bölüm (section) metni yazar. Bunu bir grup kodu 2 ve bölümün adını belirten bir metin takip eder. Bu bölümler üstbilgi (header), sınıflar (classes), tablolar (tables), blok (block), varlık (entity), nesnelere (objects) ve pul imge (thumbnailimage) 'dir. (Autodesk Inc, 2012)

- Üstbilgi bölümü: Bu bölüm çizim ile ilgili genel bilgileri ve çizim değişkenlerini ayarlamayı içerir. Her değişken, üstbilgi bölümünde değişkenin adını veren “9” rakamı ile etiketlenir, ardından değişkenin değerini sağlayan gruplar gelir. (Autodesk Inc, 2012)
- Sınıflar bölümü: Bu bölüm veri tabanının blok, varlık ve nesne bölümlerinde tanımlanan bilgileri kaydeder. Sınıf hiyerarşisinde bir sınıf tanımının kalıcı olarak sabitlendiği varsayılmaktadır. (Autodesk Inc, 2012)
- Tablolar bölümü: Bu bölüm grup kodlarını açıklar ve uygulamalar tarafından kullanılır. Tablolar bölümü, her biri bir değişken sayısı kadar varlık içerebilen çeşitli tablolar içerir. Bu kodlar, varlık tanımlama listelerinde AutoLISP ve ObjectARX uygulamaları tarafından da kullanılır. Tablolar bölümünde yer alan tablolar; Autocad uygulama tanımlama tablosu, blok kayıt tablosu, boyut stili tablosu, katman tablosu, çizgi tipi tablosu, yazma türleri tablosu, kullanıcı koordinat sistemi tablosu, görüntü tablosu ve görünüm penceresi tablosudur. (Autodesk Inc, 2012)
- Blok bölümü: Bu bölümde tanımlı bloklar incelenir. “Hatch” komutunun oluşturduğu anonim bloklar ve ilişkisel boyutlandırma da dâhil olmak üzere, çizimde kullanılan blokları oluşturan varlıkları içerir. Bu bölümdeki varlıkların formatı, varlıklar bölümündekilerle aynıdır. Blok bölümündeki tüm varlıklar blok (block) ve bitiş blok (endblk) varlıkları arasında görünür. Blok ve bitiş blok varlıkları yalnızca blok bölümünde görünür. Blok tanımları hiçbir zaman iç içe geçmiş değildir. Yani, başka bir blok-bitiş (blok-endblk)

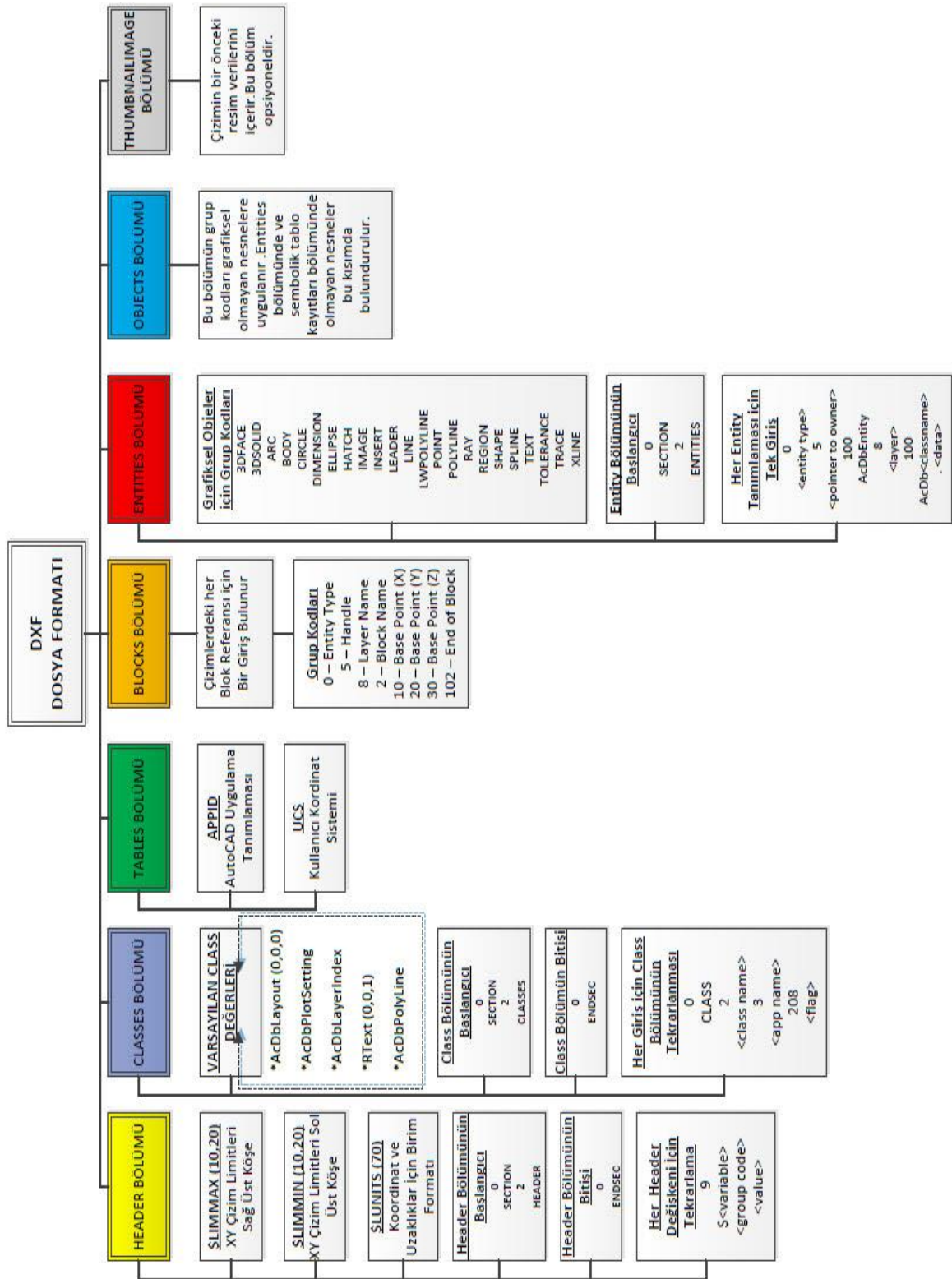
çiftinde hiç blok veya bitiş işareti görünmez. Ancak bir blok tanımı bir ekleme ögesi içerebilir. (Autodesk Inc, 2012)

- Varlık bölümü: Çizim içeriğini blok referansı ile gösterir. Bu bölümde gösterilen temsil edilen grup kodları DXF dosyaları, uygulamaları (AutoLISP veya ARX) veya her ikisi için geçerli olabilir. Bir kodun açıklaması uygulamalar ve DXF dosyaları için farklı olduğunda (veya sadece biri veya diğeri için geçerlidir.), uygulamaya ait özel açıklama ve DXF dosyasına ait özel açıklama varlık bölümünde ön tanımlıdır. (Autodesk Inc, 2012)
- Nesneler bölümü: Grafikselsel formda olmayan verileri içerir. Bu kodlar bir DXF dosyasının nesneler bölümünde bulunur ve varlık tanımlama listelerinde AutoLISP ve ARX uygulamaları tarafından kullanılır. Nesneler, grafikselsel veya geometrik bir anlamı olmayanlar haricinde, varlıklar ile benzerdir. Varlıklar veya sembol tablosu kayıtları veya sembol tabloları olmayan tüm nesneler bu bölümde saklanır. Bu bölüm, sahip olunan nesnelerin topolojik olarak sıralanmasıyla nesnelerin homojen bir yığını temsil eder, böylece sahipler her zaman sahip oldukları nesnelere önce görünürler. (Autodesk Inc, 2012)
- Pul imge bölümü: Bu bölüm, bir önceki görüntü verilerini içerir. Bu ise isteğe bağlıdır. (Autodesk Inc, 2012)

Yukarıda bahsedilen bu ayrıştırma (Şekil 3.2.), küçük bir çizimden bir DXF dosyası üretmek, yazdırmak ve daha sonra sunulan bilgiyi başka bir uygulama ile okurken bu ayrıştırmaya başvurma adına önemlidir. Bir çizim dosyasında belirtilen neredeyse tüm veriler DXF dosya formatında temsil edilebilir.

DXF uzantılı dosya formatı Matlab programına aktarılırken varlık bölümünde sınıflandırılan grafikselsel çeşitliliklere göre ayrıştırılır. Bu uzantıdaki çizim komutlarını sembolize eden grup kodları (Şekil 3.2.) saptanıp çizilen verinin şekline göre gerekli olan özellikleri ayıklanıp kaydedilir. Çizim programlarından alınan üç boyutlu verilerin bilgileri

Matlab ortamına aktarılıp ayrıştırılması yapıldıktan sonra komutlarını sembolize eden geometrik sınırlarının belirlenmesi sağlanır. Matlab programında kullanılan doğru kodlarla fonksiyonlar oluşturularak üç boyutlu şekillerin çizimleri gerçekleştirilir. Çizim programının tüm sürümlerinde DXF uzantılı kodların komutları değişiklik göstermemesine rağmen çizimler sırasında oluşan ölçek farklılıkları belirlenerek sabit bir ölçeğe çevrilir ve bu ölçeğe göre Matlab programında çizimler oluşturulur. (Autodesk Inc, 2012)



Şekil 3.2. DXF dosya formatının bölüm haritası

### 3.3. STL Uzantılı Dosya Formatı

STL uzantılı dosya formatı, CAD yazılımı ile ilgili bir dosya formatıdır ve üç boyutlu parçalar hakkında bilgi depolar. STL diğer birçok yazılım paketi tarafından desteklenmekle beraber hızlı prototipleme ve bilgisayar destekli üretim için yaygın olarak kullanılmaktadır. STL dosyaları, renk, doku veya diğer yaygın CAD modeli özniteliklerinin hiçbir temsili olmadan sadece üç boyutlu bir parçanın yüzey geometrisini tanımlar. STL dosya formatının temel amacı, üç boyutlu bir parçanın iki boyutlu dış yüzeyini küçük üçgenler kullanarak (mozaikleyerek) bir dosyadaki yüzeylerin yüzey geometrisini kodlamaktır. Mozaikleme, bir yüzeyde üst üste binme veya boşluk bulunmayacak şekilde bir veya daha fazla geometrik şekil ile döşeme işlemidir. STL dosya formatı ile üç boyutlu bir parçaya ait köşe koordinatları, yüzey bilgileri ve normal vektör bilgileri saklanabilir. (Anonim2, 2017)

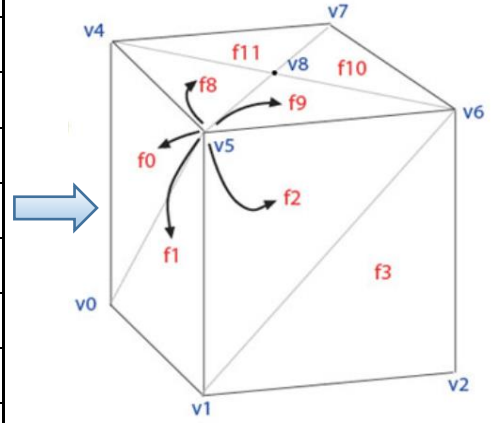
Bu çalışmada, çizim dosyalarından alınan veriler DXF uzantısıyla alınırken üç boyutlu parçanın sadece alt yapısını oluşturan bilgiler aktarılır. Fakat aktarımın düzgün bir şekilde tamamlanması ve veri kayıplarını en aza indirmesi ve yüzey sınırlarının elde edilebilmesi için Autocad programına ait ASCII veya ikili sistem (binary) yapısında oluşturulabilen STL dosya formatından da faydalanılmaktadır. STL dosya formatından elde edilen üç boyutlu bir parçaya ait yüzeyler, belirlenen üç tepe noktası (köşe) ile ilişkili olarak tanımlanır. (Çizelge 3.1., Çizelge 3.2., Şekil 3.3.) Bunlar normallerine göre birleştirilir. Bu durum parça yüzeylerini tanımlamaya yardımcı olur. Bu sayede parçanın yapı taşlarını oluşturan noktalar STL dosya yapısının yüzeyindeki verilerle kesiştirilerek sınırlar belirlenir.

<b>f0</b>	v0 v4 v5
<b>f1</b>	v0 v1 v1
<b>f2</b>	v1 v5 v6
<b>f3</b>	v1 v5 v2
<b>f4</b>	v2 v6 v7
<b>f5</b>	v2 v6 v3
<b>f6</b>	v3 v7 v4
<b>f7</b>	v3 v7 v0
<b>f8</b>	v8 v4 v4
<b>f9</b>	v8 v5 v5
<b>f10</b>	v8 v6 v6
<b>f11</b>	v8 v7 v7
<b>f12</b>	v9 v4 v4
<b>f13</b>	v9 v5 v5
<b>f14</b>	v9 v6 v6
<b>f15</b>	v9 v7 v7

Çizelge 3.2. Yüzey Listesi

<b>v0</b>	0,0,0	<b>f0 f1 f12 f15 f7</b>
<b>v1</b>	1,0,0	<b>f2 f3 f13 f12 f1</b>
<b>v2</b>	1,1,0	<b>f4 f5 f14 f13 f3</b>
<b>v3</b>	0,1,0	<b>f6 f7 f15 f14 f5</b>
<b>v4</b>	0,0,1	<b>f6 f7 f0 f8 f11</b>
<b>v5</b>	1,0,1	<b>f0 f1 f2 f9 f8</b>
<b>v6</b>	1,1,1	<b>f2 f3 f4 f10 f9</b>
<b>v7</b>	0,1,1	<b>f4 f5 f6 f11 f10</b>
<b>v8</b>	.5,.5,0	<b>f8 f9 f10 f11</b>
<b>v9</b>	.5,.5,1	<b>f12 f13 f14 f15</b>

Çizelge 3.1. Köşe Listesi



Şekil 3.3. Cisme ait yüzey ve köşe ilişkisi

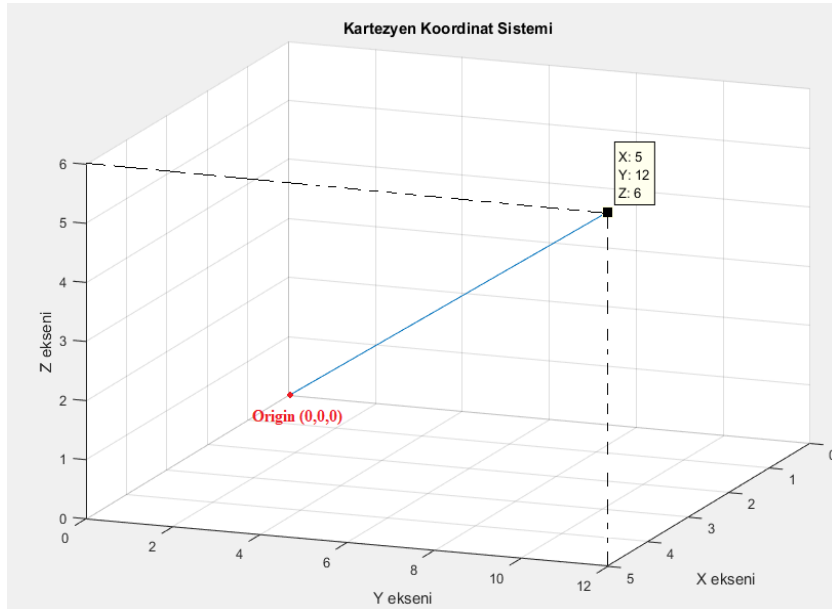
## 4. MATERYAL VE YÖNTEM

Çalışmanın bu bölümünde, Autocad programında çizilen üç boyutlu parçaların DXF ve STL dosya formatlarında kaydedilmesinden sonra bu dosyalardan çekilen veriler kullanılarak Matlab programında analizleri gerçekleştirilir. Yapılan analizler sonucunda 5 eksenli çapak alma robot kolunun uç elemanının üç boyutlu bir parçanın etrafında takip edeceği yörünge ve parçanın köşe noktalarına ait koordinatlar hesaplanır ve bu koordinatlar yörünge ve köşe koordinatları olmak üzere ayrı ayrı TXT dosya formatlarında kaydedilir.

### 4.1. DXF Dosya Formatının Analizi

#### 4.1.1. Koordinat Sistemi

CAD çizim programları, bir cismin çizilebilmesi için koordinat sistemine ihtiyaç duyarlar. Uzayda yer alan bir noktanın veya başka bir geometrik elemanın konumu koordinat sistemi üzerinde birleşenlerine ayrılarak ifade edilir. Bu çalışmada, koordinat sistemi olarak Kartezyen koordinat sistemi (dünya koordinat sistemi) kullanılmıştır. Kartezyen koordinat sistemi birbirine dik üç adet koordinat ekseninden oluşur ve bunlar X, Y ve Z eksenleridir. Genellikle X ve Y eksenleri yatayda yer alan iki boyutlu bir noktayı veya geometrik elemanı ifade etmek için kullanılır. Z eksenine ise dikey eksen olarak kabul edilerek üçüncü boyutta yer alan bir noktanın veya geometrik elemanın yükseklik bilgisini tutar. X, Y ve Z eksenlerinin kesiştiği nokta ise başlangıç noktası (orijin) olarak tanımlanır ve bu noktanın koordinatları (0,0,0)'dır. ( $X=0$ ;  $Y=0$ ;  $Z=0$ ) (Şekil 4.1.) (Anonim3, 2010)



Şekil 4.1. Kartezyen Koordinat Sistemi

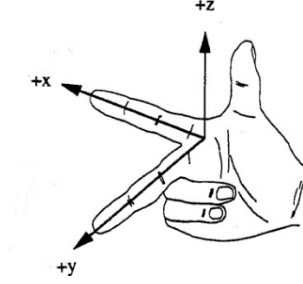
Autocad programında, çizim veri tabanından (ve DXF dosyasından) yer kazanmak için, her bir geometrik elemanla ilişkili noktalar, programın kendi nesne koordinat sistemi cinsinden ifade edilir. Nesne koordinat sistemi ile geometrik elemanın üç boyutlu uzaydaki konumunu tanımlamak için gereken bilgi nesne koordinat sisteminin Z eksenini ve yükseklik değerini açıklayan 3 boyutlu vektördür. Belirli bir Z eksenini (veya ekstrüzyon) yönü için, başlangıç noktasını üç boyutlu uzayda tanımlayarak ve X, Y eksenlerini Z eksenini etrafında döndürerek tanımlanan sonsuz sayıda koordinat sistemi vardır. Bununla birlikte, aynı Z eksenini yönü için, sadece bir nesne koordinat sistemi vardır ve aşağıdaki özelliklere sahiptir:

- Başlangıç noktası dünya koordinat sistemi ile çakışır.
- X, XY ve Y eksenlerinin XY düzlemine yönelimi, rastgele fakat tutarlı bir şekilde hesaplanır. Autocad programı, bu hesaplamayı rastgele eksen algoritması kullanarak gerçekleştirir. (Autodesk Inc, 2012 )

Rastgele eksen algoritması, bir koordinat sisteminin Z eksenini olarak kullanılacak birim uzunluktaki bir vektör verildiğinde koordinat sistemi için karşılık gelen bir X eksenini üretir. Y eksenini de sağ el kuralının uygulanmasıyla bulunur. (Şekil 4.2.) Yönteme göre, verilen Z eksenini (Normal vektör olarak da adlandırılır.) incelenir. Pozitif veya negatif dünya Z eksenine yakınsa, rastgele X eksenini, Y eksenini ve verilen Z ekseninin vektörel çarpımı ile



bulunur. Yakın değilse, rastgele X eksenini, dünya Z ekseninin ve verilen Z ekseninin vektörel çarpımı ile bulunur. Kararın verildiği sınır 1/64 olarak Autocad programı tarafından belirlenmiştir. (Autodesk Inc, 2012)



Şekil 4.2. Sağ-el kuralı

Rastgele eksen algoritmasına göre, (Tüm vektörlerin üç boyut boşluğunda olduğu varsayılır ve dünya koordinat sisteminde belirtilir.) verilen vektör N'nin normal vektör olarak adlandırıldığı varsayıldığında dünya Y ve dünya Z eksenlerine sırasıyla  $W_y$  ve  $W_z$  denir, bunlar sırasıyla (0,1,0) ve (0,0,1) olur. Amaç, normal vektörünü (N) kullanarak, rastgele X ( $A_x$ ) ve Y ( $A_y$ ) eksenlerini bulmaktır. Normal vektörü aynı zamanda rastgele Z eksenini ( $A_z$ ) olarak da adlandırılabilir. Bu durumda algoritmanın akışı aşağıdaki gibi özetlenebilir. (Autodesk Inc, 2012)

Eğer ( $N_x$ 'in mutlak değeri)  $< 1/64$  ve ( $N_y$ 'nin mutlak değeri)  $< 1/64$ ) ise,  
 $A_x = W_y \times N$  ("X" vektörel çarpım işaretidir.)  
 Değilse,  
 $A_x = W_z \times N$  ( $A_x$  birim uzunluğa ölçeklenir.)  
 $A_y = N \times A_x$  ( $A_y$  birim uzunluğa ölçeklenir.)

Rastgele eksen algoritması işlemi tamamlandıktan sonra, sınıflandırma ve sıralama işlem basamakları gerçekleştirilir. Çizim şekilleri (çizgi, daire, yay, elips) çizim yönünü tanımlayan ekstrüzyon yön vektör bilgisine sahiptir. Bu bilgiler, değişkenleri benzer çizimlerden ayırmak için kullanılır. Bu nedenle çizim değişkenleri çizimlerin normal vektörüne doğru kayması için gruplandırılmıştır. Buradan yola çıkarak, diğer teknik çizim verileriyle kaydedildikten ve ilişkilendirildikten sonra, üç boyutlu uzayda yeniden çizim mümkün hale gelir. Gruplama adımı tamamlandıktan sonra ise sıralama işlemi yapılır.

Sıralama işlemi, diğer iki eksen sabitken, bir eksen üzerindeki değeri en düşük değerden en büyük değere değiştirilmesiyle yapılır. (Autodesk Inc, 2012)

#### 4.1.2. Çizgi değişkenlerinin belirlenmesi

Üç boyutlu bir çizginin çizilebilmesi için, Kartezyen koordinat sisteminde, çizgiye ait başlangıç noktalarının  $(x_1, y_1, z_1)$ , bitiş noktalarının  $(x_2, y_2, z_2)$  ve ekstrüzyon yön vektörünün bilinmesi gerekir. Bu değişkenlerin Autocad programından elde edilen DXF dosya formatının, varlık bölümünden çizgi değişkenlerine ait grup kodları taranarak çizimde yer alan çizgiler Matlab programına aktarılır. DXF dosya formatında çizgiye ait grup kodları (10, 20, 30) çizginin dünya koordinat sisteminde başlangıç noktalarını, (11, 21, 31) çizginin dünya koordinat sisteminde bitiş noktalarını, (210, 220, 230) ise çizginin ekstrüzyon yön vektörünü temsil eder. (Şekil 4.3.) (Autodesk Inc, 2012)

```
switch EntityName
case 'LINE'
    % ((X,Y,Z)Start Point (in WCS), (X,Y,Z) End point (in WCS), (X,Y,Z) Extrusion Direction)
    c_Line{iLine,1} = [str2double(ValGroupCode(10,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(20,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(30,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(11,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(21,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(31,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(210,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(220,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(230,GroupCodeEntity,EntityValues))];
    iLine = iLine+1;
```

Şekil 4.3. Çizgi değişkenlerinin DXF dosyasından okunmasını gösteren Matlab kodu

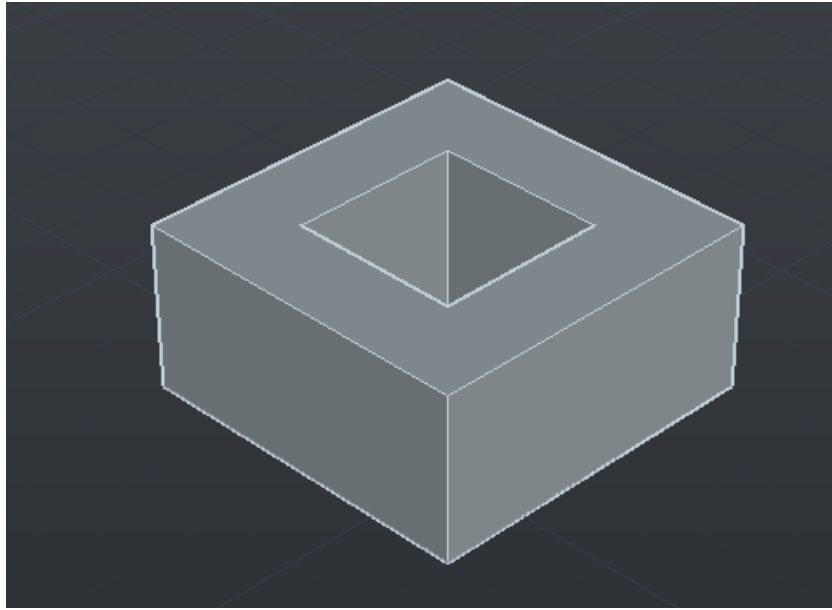
Çizgiye ait değişkenlerin Matlab programında okunmasından sonra, çizgiye ait yörünge Matlab ortamında iki noktası bilinen çizgi denklemlerini kullanan “*plot3*” komutuyla (Şekil 4.4.) çizdirilir. (Şekil 4.5., Şekil 4.6.)

```

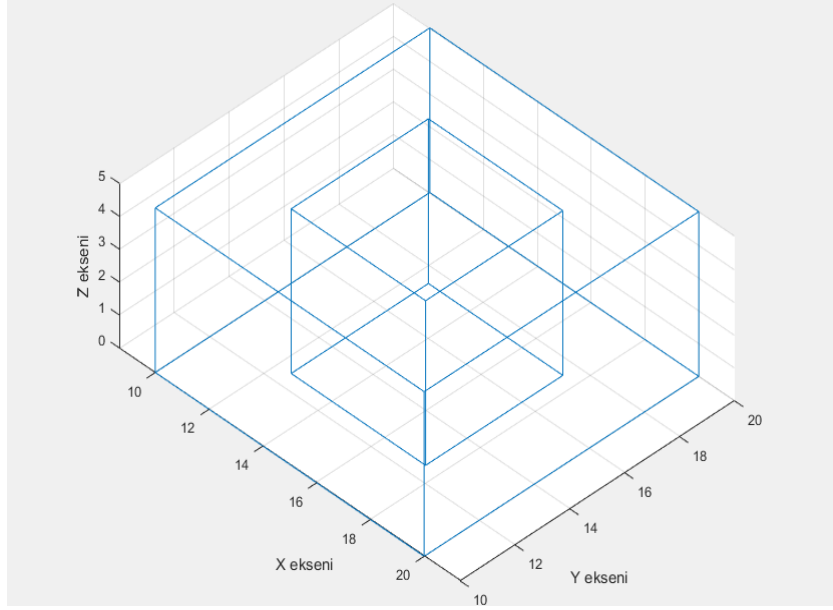
%% Line Plotting
hl=findobj('Type','axes','Tag','trajectory');
axes(hl);
rotate3d on
if(~isempty(c_Line{1}))
[   for i=1:length(c_Line)
        plot3([c_Line{i}(1,1) c_Line{i}(1,4)],[c_Line{i}(1,2) c_Line{i}(1,5)],...
              [c_Line{i}(1,3) c_Line{i}(1,6)],'Color','b');
        hold on
    end
end
xlabel('X eksen')
ylabel('Y eksen')
zlabel('Z eksen')
title('DXF dosyasının MATLAB da yörüngesinin oluşturulması')
grid on
axis equal
hold on

```

Şekil 4.4. Matlab ortamında çizgi denklemlerinin kodlanması



Şekil 4.5. Autocad'te 3 boyutlu çizgilerden oluşturulmuş cisim



Şekil 4.6. Şekil 4.5.'in Matlab'ta çizimi

#### 4.1.3. Yay değişkenlerinin belirlenmesi

Üç boyutlu bir yayın çizilebilmesi için, Kartezyen koordinat sisteminde, yaya ait merkez noktalarının, yarıçapının, başlangıç açısının, bitiş açısının ve ekstrüzyon yön vektörünün bilinmesi gerekir. Bu değişkenlerin Autocad programından elde edilen DXF dosya formatının, varlık bölümünden yay değişkenlerine ait grup kodları taranarak çizimde yer alan yaylar Matlab programına aktarılır. DXF dosya formatında yaya ait grup kodları (10,20,30) nesne koordinat sisteminde yayın merkez noktalarını, (40) yarı çapını, (50) derece cinsinden başlangıç açısını, (51) derece cinsinden bitiş açısını , (210, 220, 230) ise yayın ekstrüzyon yön vektörünü temsil eder. (Şekil 4.7.) (Autodesk Inc, 2012)

```

case 'ARC'
    % ((X,Y,Z)Center (in OCS), Radius, Start angle (in degree), End
    % angle (in degree), (X,Y,Z) Extrusion Direction))
    c_Arc{iArc,1} = [str2double(ValGroupCode(10,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(20,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(30,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(40,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(50,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(51,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(210,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(220,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(230,GroupCodeEntity,EntityValues))];
    iArc = iArc+1;

```

Şekil 4.7. Yay değişkenlerinin DXF dosyasından okunmasını gösteren Matlab kodu

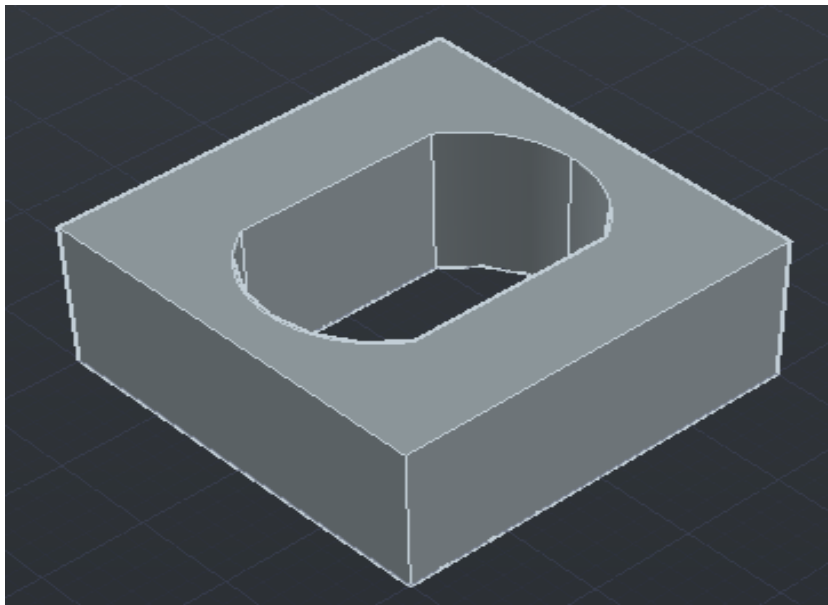
Yaya ait değişkenlerin Matlab programında okunmasından sonra, yaya ait yörünge Matlab programında, yay değişkenlerinden ve yaya ait denklemlerden (Şekil 4.8.) faydalanılarak çizdirilir. (Şekil 4.9., Şekil 4.10.) Yaya ait denklemler kullanılırken yayın merkez noktaları nesne koordinat sisteminde olduğundan dolayı merkez noktaları dünya koordinat sistemine; yayın başlangıç ve bitiş açıları derece cinsinden olduğundan da radyana dönüştürme işlemleri yapılır.

```

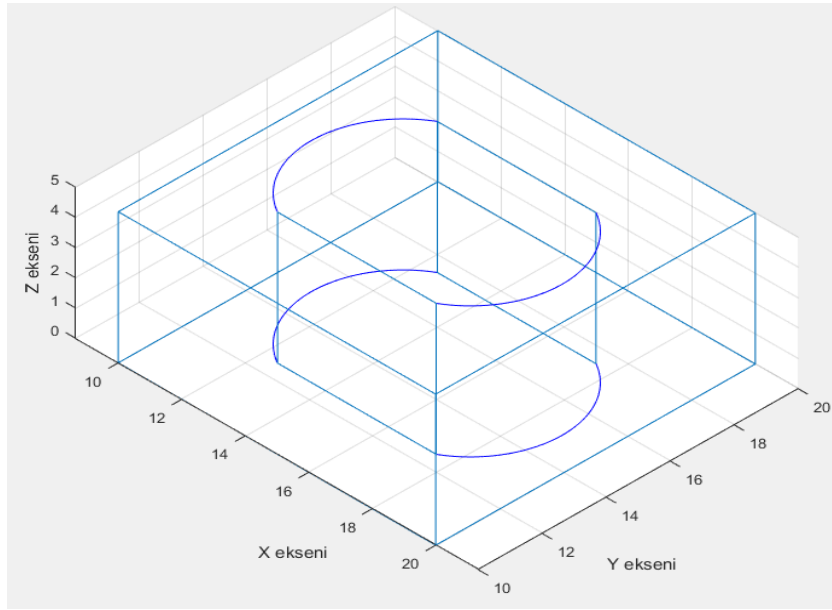
function out = plotArc3D(center,normal,radius,startangle,endangle,color)
%% Plotting ARC in 3D
if isempty(color)
    color = 'b';
end
start_rad = DR(startangle);
end_rad = DR(endangle);
N=(end_rad-start_rad)/0.01 +1;
x=linspace(0,5,N);
y=linspace(0,5,N);
z=linspace(0,5,N);
k=1;
if(issame(normal,[1 0 0]) && startangle>endangle)
    temp=start_rad;
    start_rad=end_rad;
    end_rad=temp;
end
for i=start_rad:0.01:end_rad
    x(k)=center(1,1)+cos(i)*radius;
    y(k)=center(1,2)+sin(i)*radius;
    z(k)=center(1,3);
    [x(k), y(k), z(k)]=ocs_to_wcs(x(k),y(k), z(k), normal(1,1), normal(1,2), normal(1,3));
    k=k+1;
end
out = [x(:),y(:),z(:)];
plot3(x,y,z,'Color',color);
end

```

Şekil 4.8. Matlab ortamında yay denklemlerinin kodlanması



Şekil 4.9. Autocad'te 3 boyutlu yaylardan oluşturulmuş cisim



Şekil 4.10. Şekil 4.9.'un Matlab'ta çizimi

#### 4.1.4. Daire değişkenlerinin belirlenmesi

Üç boyutlu bir dairenin çizilebilmesi için, Kartezyen koordinat sisteminde, daireye ait merkez noktalarının, yarıçapının ve ekstrüzyon yön vektörünün bilinmesi gerekir. Bu değişkenlerin Autocad programından elde edilen DXF dosya formatının, varlık bölümünden daire değişkenlerine ait grup kodları taranarak çizimde yer alan daireler Matlab programına aktarılır. DXF dosya formatında daireye ait grup kodları (10,20,30) dairenin nesne koordinat sisteminde merkez noktalarını, (40) yarı çapını, (210, 220, 230) ise dairenin ekstrüzyon yön vektörünü temsil eder. (Şekil 4.11.) (Autodesk Inc, 2012)

```

case 'CIRCLE'
    % ((X,Y,Z)Center (in OCS), Radius, (X,Y,Z) Extrusion
    % Direction))
    c_Cir{iCir,1} = [str2double(ValGroupCode(10,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(20,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(30,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(40,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(210,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(220,GroupCodeEntity,EntityValues)),...
        str2double(ValGroupCode(230,GroupCodeEntity,EntityValues))];
    iCir = iCir+1;

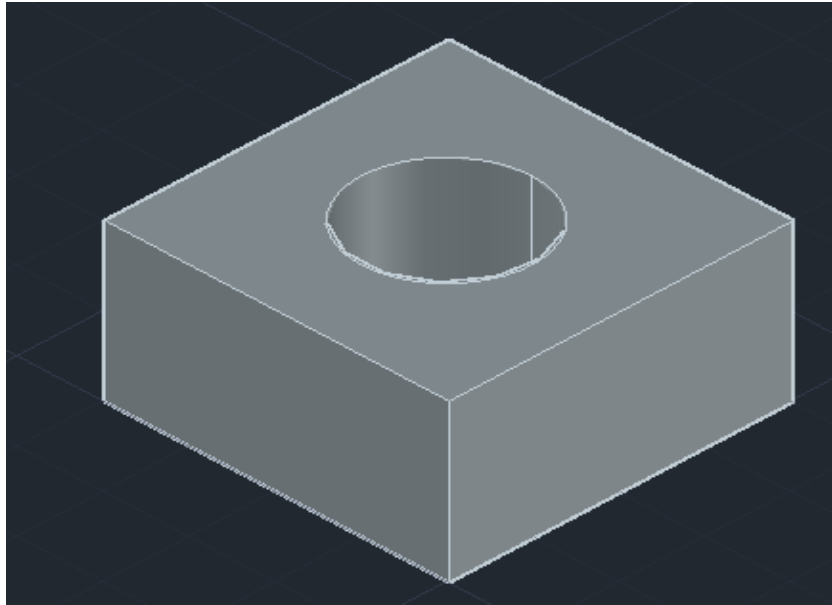
```

Şekil 4.11. Daire değişkenlerinin DXF dosyasından okunmasını gösteren Matlab kodu

Daireye ait deęişkenlerin Matlab programında okunmasından sonra, daireye ait yörünge Matlab programında, daire deęişkenlerinden ve daireye ait denklemlerden (Şekil 4.12.) faydalanılarak çizdirilir. (Şekil 4.13., Şekil 4.14.) Daireye ait denklemler kullanılırken dairenin merkez noktaları nesne koordinat sisteminde olduğundan dolayı merkez noktalarının dünya koordinat sistemine dönüştürme işlemleri yapılır.

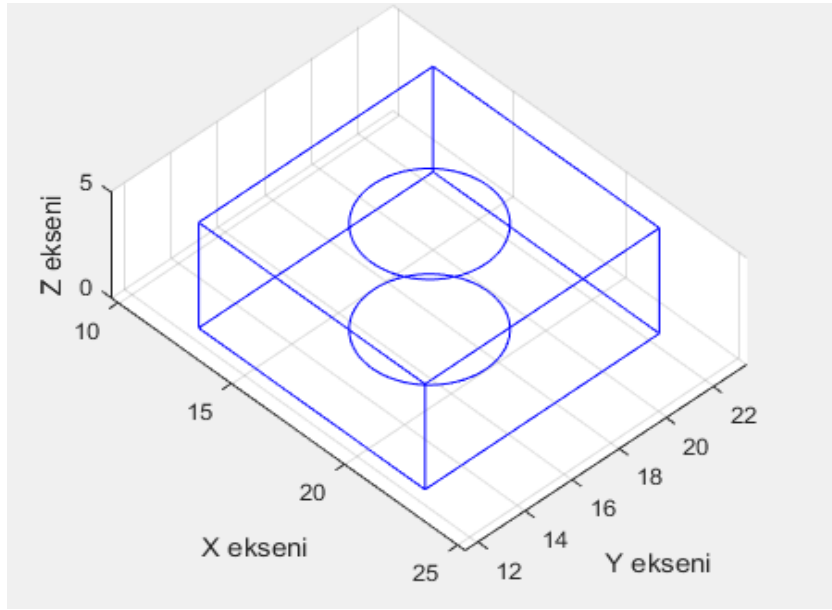
```
function out = plotCircle3D(center,normal,radius,color)
%% Plotting Circle in 3D
if isempty(color)
    color = 'b';
end
[center(1,1),center(1,2),center(1,3)]=ocs_to_wcs(center(1,1),...
    center(1,2),center(1,3),normal(1,1),normal(1,2),normal(1,3));
theta=0:0.01:2*pi;
v=null(normal);
points= repmat(center',1,size(theta,2))+radius*(v(:,1)*cos(theta)+...
    v(:,2)*sin(theta));
plot3(points(1,:),points(2,:),points(3,:),color);
out = [points(1,:) ',points(2,:) ',points(3,:) '];
end
```

Şekil 4.12. Matlab ortamında daire denklemlerinin kodlanması



Şekil 4.13. Autocad'te 3 boyutlu dairelerden oluşturulmuş cisim





Şekil 4.14. Şekil 4.13.'ün Matlab'ta çizimi

#### 4.1.5. Elips değişkenlerinin belirlenmesi

Üç boyutlu bir elipsin çizilebilmesi için, Kartezyen koordinat sisteminde, elipse ait merkez noktalarının, merkez noktasına göre büyük eksen ölçülerinin, ekstrüzyon yön vektörünün, küçük eksen ölçülerinin büyük eksen ölçülerine oranının, başlangıç ve bitiş açısının bilinmesi gerekir. (Tam elips için başlangıç açısı radyan cinsinden 0; bitiş açısı radyan cinsinden  $2\pi$ 'dir.) Bu değişkenlerin Autocad programından elde edilen DXF dosya formatının, varlık bölümünden elips değişkenlerine ait grup kodları taranarak çizimde yer alan elipsler Matlab programına aktarılır. DXF dosya formatında elipse ait grup kodları (10,20,30) elipsin dünya koordinat sisteminde merkez noktalarını, (11,21,31) dünya koordinat sisteminde merkez noktasına göre büyük eksen ölçülerini, (210, 220, 230) ekstrüzyon yön vektörünü, (40) küçük eksen ölçülerinin büyük eksen ölçülerine oranını, (41,42) başlangıç ve bitiş açısını temsil eder. (Şekil 4.15.) (Autodesk Inc, 2011)

```

case 'ELLIPSE'
% ((X,Y,Z)Center (in WCS), (X,Y,Z) Endpoint of major axis relative to the center (in WCS)
% (X,Y,Z) Extrusion Direction, Ratio of minor axis to major
% axis, Start parameter (this value is 0.0 for a full ellipse),
% End parameter (this value is 2pi for a full ellipse)
c_Ellipse{iEllipse,1} = [str2double(ValGroupCode(10,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(20,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(30,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(11,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(21,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(31,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(210,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(220,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(230,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(40,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(41,GroupCodeEntity,EntityValues)),...
    str2double(ValGroupCode(42,GroupCodeEntity,EntityValues))];
iEllipse = iEllipse + 1;

```

Şekil 4.15. Elips değişkenlerinin DXF dosyasından okunmasını gösteren Matlab kodu

Elipse ait değişkenlerin Matlab programında okunmasından sonra, elipse ait yörünge Matlab programında, elips değişkenlerinden ve elipse ait denklemlerden (Şekil 4.16.) faydalanılarak çizdirilir. (Şekil 4.20., Şekil 4.21.) Elipse ait denklemler kullanılırken elipsin uzaydaki dönüş açısının bulunabilmesi için elipse ait uzaydaki noktalar X,Y,Z eksen yönündeki dönüşüm matrisleri ile vektörel çarpılır. (Şekil 4.17., Şekil 4.18., Şekil 4.19.)

```

function out = plotEllipse3D(center,endPoint,normal,ratio,start_angle,end_angle,color)
%% Plotting ELLIPSE in 3D
if isempty(color)
    color='b';
end
phi=start_angle:0.01:end_angle;
div=(end_angle-start_angle)/0.01;
Ex=-endPoint(1);
Ey=-endPoint(2);
P=zeros(3,1);
E=[Ex Ey]';
a=-norm(E);
b=ratio*a;
theta=atan2(Ey,Ex);
if issame(normal,[1 0 0])
    R=rotationx(theta);
elseif issame(normal,[0 1 0])
    R=rotationy(theta);
else
    R=rotationz(theta);
end
for i=1:div
    P(1) = a*cos(phi(i));
    P(2) = b*sin(phi(i));
    P(3) = center(3);
    Pr = R*P;
    x(i)= center(1)+Pr(1);
    y(i)= center(2)+Pr(2);
    z(i)= center(3);
end
out = [x(:),y(:),z(:)];
plot3(x,y,z,color);
end

```

Şekil 4.16. Matlab ortamında elips denklemlerinin kodlanması

```

function R = rotationx(phi)
%Rotation matrix in the x direction
R = [ 1      0      0;
      cos(phi) -sin(phi) 0;
      sin(phi)  cos(phi) 0];
end

```

Şekil 4.17. X eksenindeki dönüşüm matrisi

```

function R = rotationy(phi)
%Rotation matrix in the y direction
R = [cos(phi) -sin(phi) 0;
      0      1      0;
      sin(phi)  cos(phi) 0];
end

```

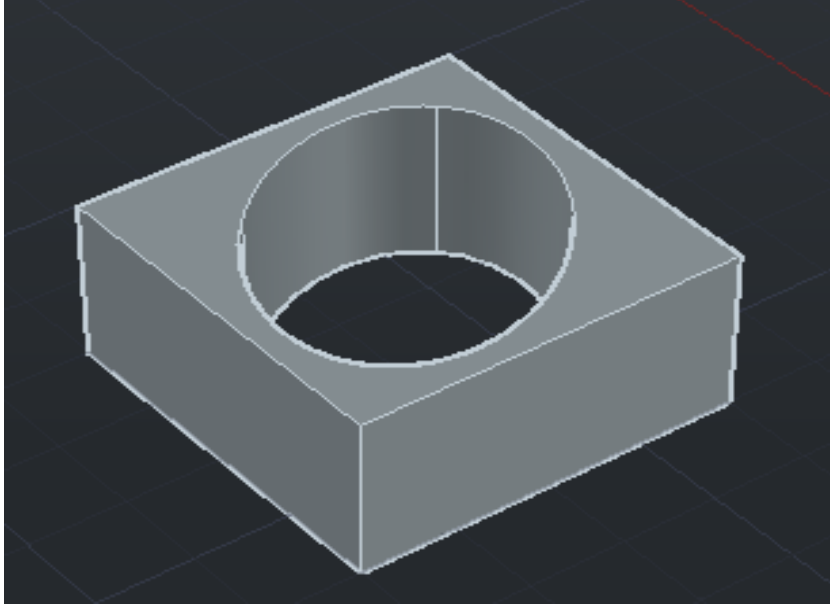
Şekil 4.18. Y eksenindeki dönüşüm matrisi

```

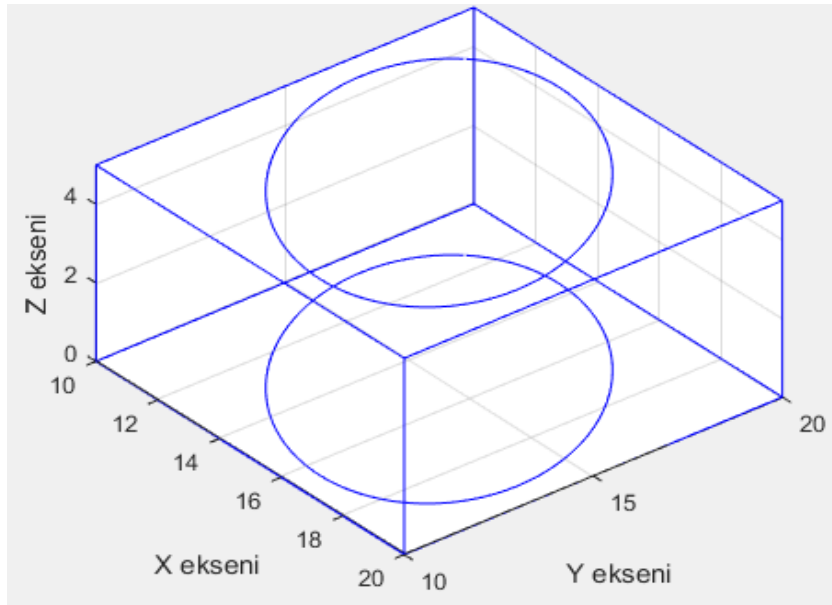
function R = rotation(phi)
%Rotation matrix in the z direction
    R = [cos(phi) -sin(phi) 0;
         sin(phi)  cos(phi) 0;
         0          0        1];
end

```

Şekil 4.19. Z eksenini yönündeki dönüşüm matrisi

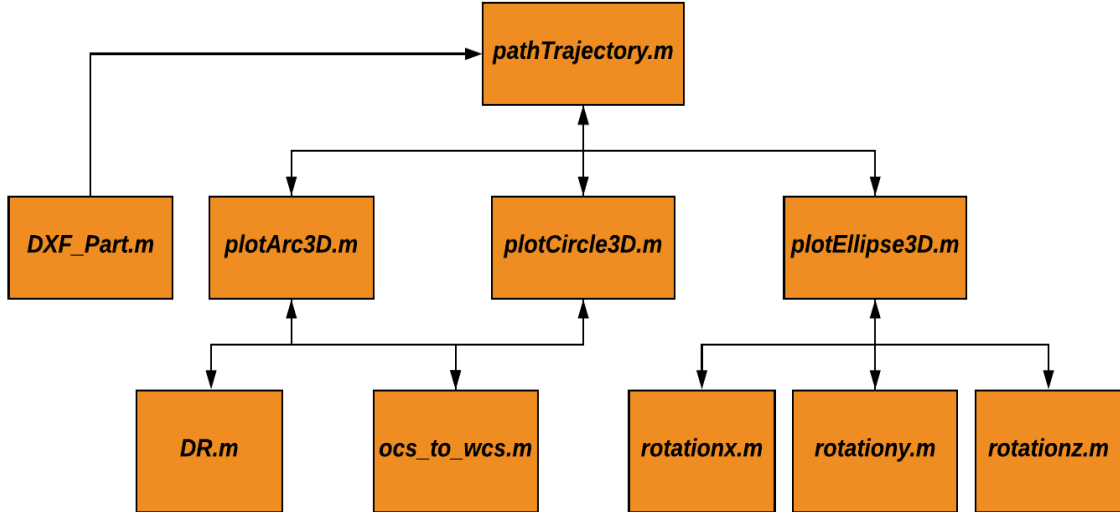


Şekil 4.20. Autocad'te 3 boyutlu elipslerden oluşturulmuş cisim



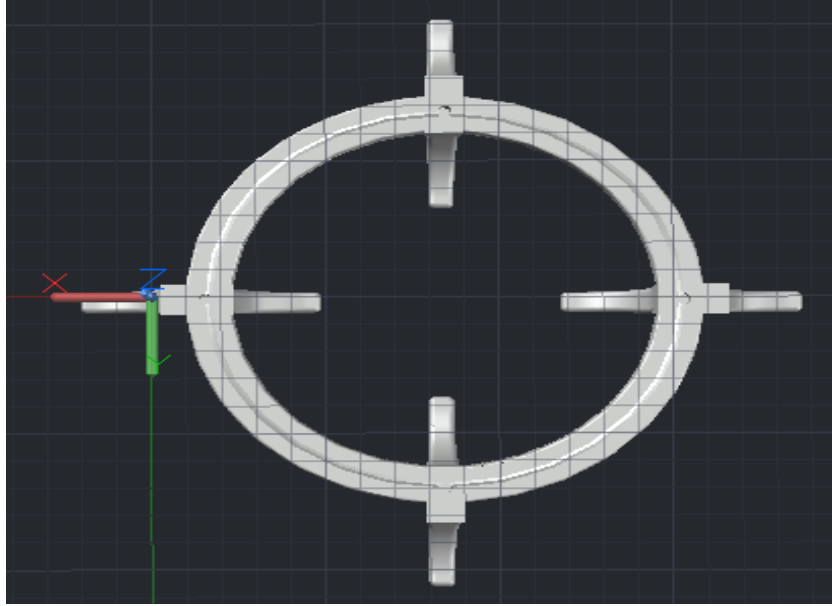
Şekil 4.21. Şekil 4.20.'nin Matlab'ta çizimi

DXF dosya formatı analizinde temel çizim şekilleri (çizgi, yay, daire, elips), Matlab programında yaratılan “*DXF\_Part.m*” fonksiyonu ile sınıflara ayrılır ve çizim şekillerinin değişkenlerine ait grup kodları bu fonksiyonda taranarak çizim şekillerinin yeniden çizdirilebilmesi için çizim şekillerine ait değişkenler elde edilir. “*pathTrajectory.m*” fonksiyonu ile çizim şekillerine ait değişkenler “*DXF\_Part.m*” fonksiyonundan okunarak her çizim şekli için yer açma ve atama işlemleri yapılır. Çizim şekillerinin üç boyutlu uzayda yeniden çizilebilmesi için “*plotCircle3D.m*”, “*plotArc3D.m*” ve “*plotEllipse3D.m*” alt fonksiyonları “*pathTrajectory.m*” fonksiyonundan çağrılarak çizim şekillerinin denklemlerine, çizim şekillerinin değişkenleri gönderilerek yeniden çizme işlemi yapılır. Daire ve yayla ait denklemlerinin yer aldığı “*plotCircle3D.m*”, “*plotArc3D.m*” fonksiyonlarından bir alt fonksiyon olan “*ocs\_to\_wcs.m*” fonksiyonu çağrılarak nesne koordinat sisteminden dünya koordinat sistemine geçiş yapılır. “*plotEllipse3D.m*” fonksiyonunun alt fonksiyonu olan eksenlere göre dönüşüm matris fonksiyonları da (*rotationx.m*, *rotationy.m* ve *rotationz.m*) yine “*plotEllipse3D.m*” fonksiyonunun içerisinde yer alır. “*DR.m*” alt fonksiyonuyla açıların derece ölçü biriminden radyan açı ölçü birime dönüştürme işlemleri yapılır. (Şekil 4.22.)

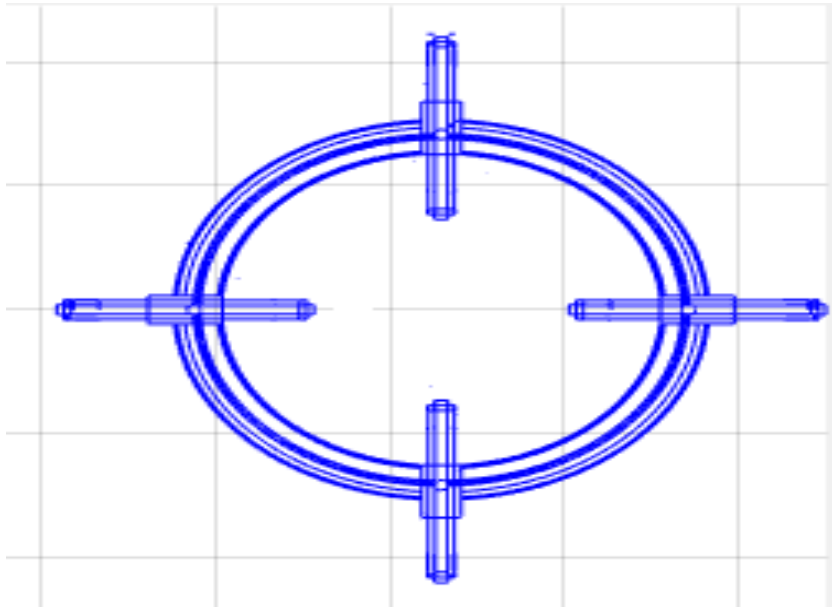


Şekil 4.22. DXF dosya formatı analizinde kullanılan fonksiyonların birbiriyle ilişkisi

Autocad'de çizilen ocak üstü ızgara çizimi, “*pathTrajectory.m*” fonksiyonuna gönderilerek Matlab ortamında yeniden çizdirilir. (Şekil 4.23., Şekil 4.24.)



Şekil 4.23. Autocad'de çizilen ocak üstü ızgara çizimi

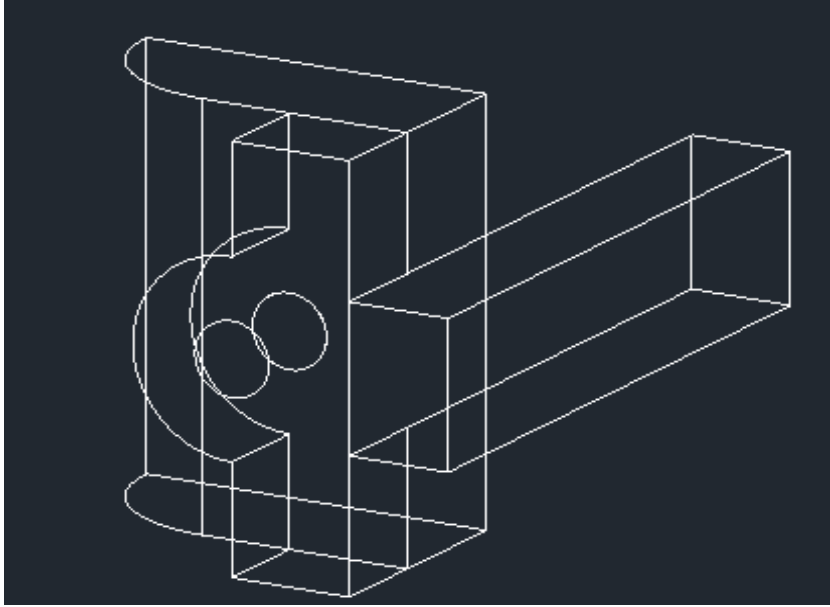


Şekil 4.24. Matlab'da çizilen ocak üstü ızgara çizimi

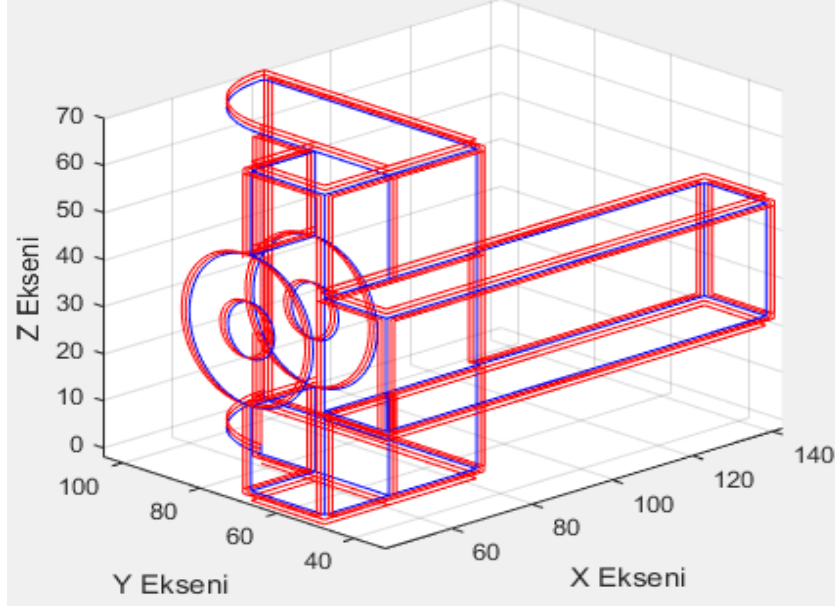
Matlab programında yeniden çizimi yapılan parçaların, kullanıcı tarafından istenilen robot kolunun uç elemanının çap bilgisi ile beraber yeniden çizim işlemi yapılarak robotun izleyeceği yörünge çıkartma işlemi tamamlanmış olur. Kullanıcı tarafından istenen robot

kolunun uç elemanının çap bilgisi cisim şekillerinin ekstrüzyon yön vektörü ile çarpılarak cismin uzaydaki konum bilgisine eklenir. Böylelikle robot kolunun uç elemanının, Cad çiziminde yer alan parçayı etrafından dolaşarak parçanın yüzeyinde var olan çapağı temizleyecek yörüngesi yaratılır. Yaratılan bu yörüngenin dünya koordinat sisteminde (Kartezyen koordinat sistemi) X, Y, Z noktaları TXT uzantılı bir dosyaya kaydedilerek robotun uç noktasının takip edeceği konum bilgisi olarak tutulur.

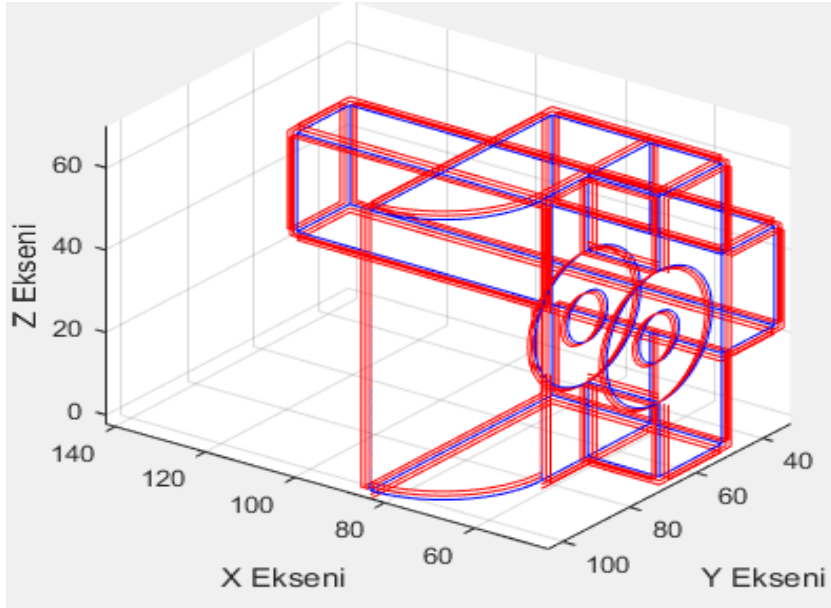
Autocad'de DXF dosya formatında kaydedilen bir makina parçasına ait çizim, Matlab'da "*pathTrajectory.m*" fonksiyonuna gönderilerek parçanın gerçek çizimi mavi renkle, robot kolunun uç elemanının izleyeceği yörünge kırmızı renkle çizdirilmiştir. Robot kolunun uç elemanının izleyeceği yörünge çizdirilirken uç elemanının çapı 4 cm olarak alınmıştır. (Şekil 4.25., Şekil 4.26. ve Şekil 4.27.) . Yaratılan bu yörüngenin dünya koordinat sisteminde (Kartezyen koordinat sistemi) X, Y, Z noktaları TXT uzantılı bir dosyaya kaydedilerek robotun uç noktasının takip edeceği konum bilgisi olarak tutulmuştur. (Şekil 4.28.)



Şekil 4.25. Autocad'de DXF dosya formatında bir makina çizimi

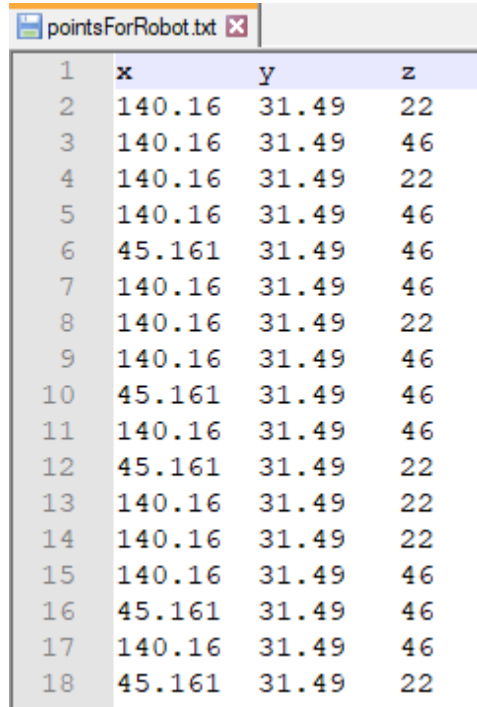


Şekil 4.26. Şekil 4.25.'e ait çizim için robot kolunun uç elemanın izleyeceği yörünge 1



Şekil 4.27. Şekil 4.25.'e ait çizim için robot kolunun uç elemanın izleyeceği yörünge 2



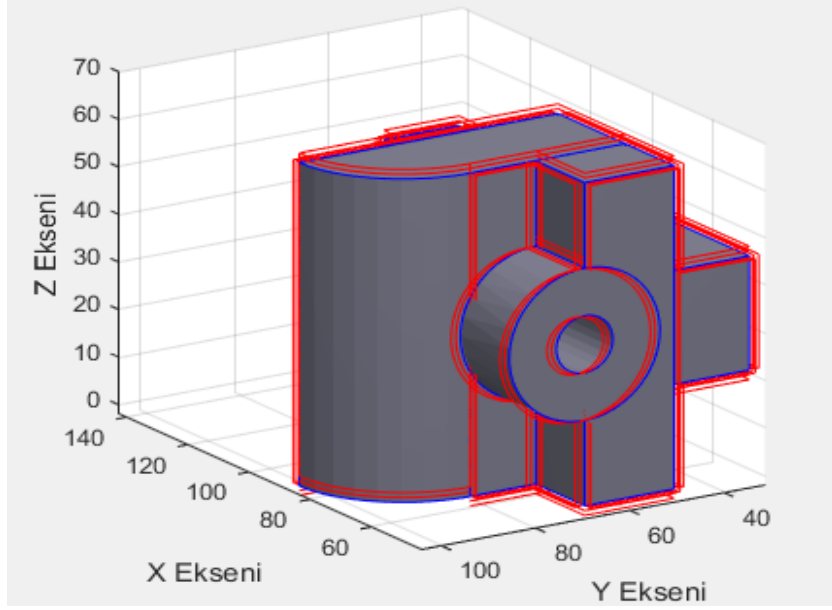


	x	y	z
1			
2	140.16	31.49	22
3	140.16	31.49	46
4	140.16	31.49	22
5	140.16	31.49	46
6	45.161	31.49	46
7	140.16	31.49	46
8	140.16	31.49	22
9	140.16	31.49	46
10	45.161	31.49	46
11	140.16	31.49	46
12	45.161	31.49	22
13	140.16	31.49	22
14	140.16	31.49	22
15	140.16	31.49	46
16	45.161	31.49	46
17	140.16	31.49	46
18	45.161	31.49	22

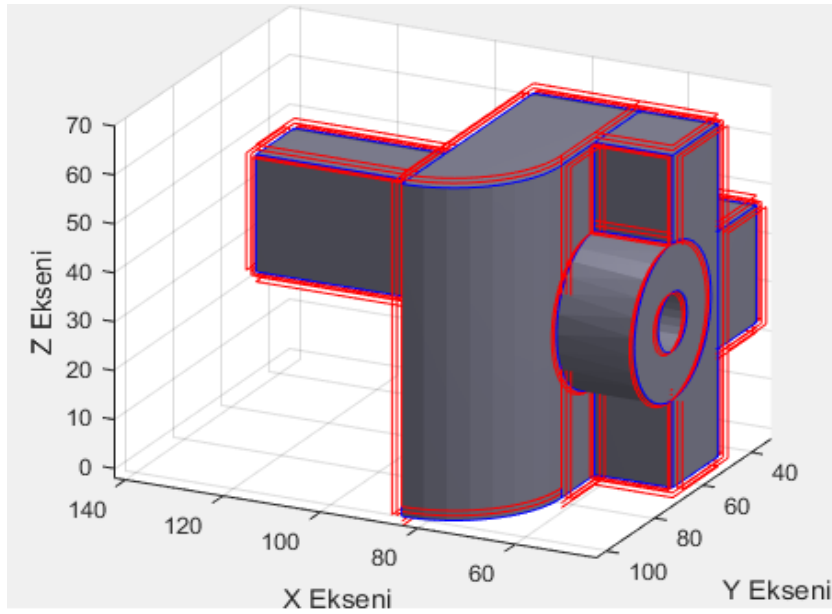
Şekil 4.28. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörüngenin X, Y ve Z noktalarının başlangıç kısmı

## 4.2. STL Dosya Formatının Analizi

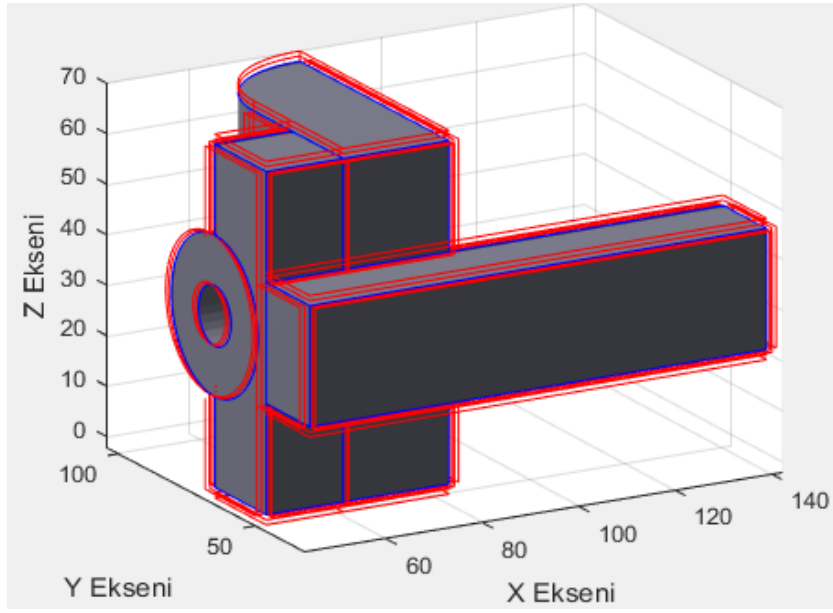
STL dosya formatı, DXF dosya formatından elde edilen çizim ve yörüngenin doğruluğunu kontrol etmek için kullanılır. STL dosya formatındaki çizim, DXF dosya formatında Matlab'da çizdirilen yörünge ile birleştirilerek yörüngenin doğruluğu test edilir. STL dosya formatı Matlab programında okunurken Matlab ortamında kodlanan "*stlread.m*" fonksiyonu "*pathTrajectory.m*" fonksiyonu tarafından çağrılır. "*stlread.m*" fonksiyonu STL dosya formatındaki bir çizimin yüzey, köşe ve ekstrüzyon yön vektörü bilgilerini geri döndürerek parçanın çizim işlemlerini gerçekleştirir. "*stlread.m*" fonksiyonundan geri döndürülen cisme ait köşe (dünya koordinat sisteminde X, Y, Z) noktaları TXT uzantılı bir dosyaya yazdırılır. (Şekil 4.29., Şekil 4.30., Şekil 4.31., Şekil 4.32., Şekil 4.33. ve Şekil 4.34.)



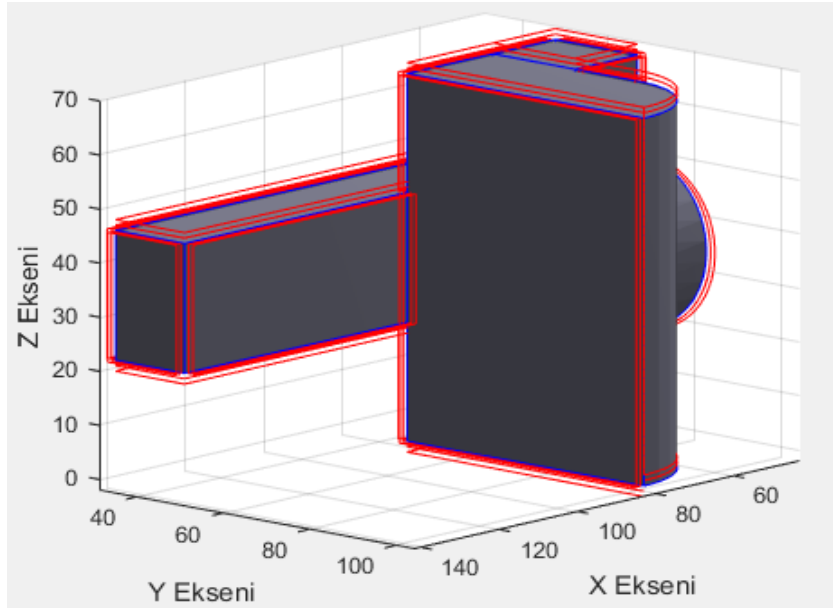
Şekil 4.29. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörüngeyi doğruluk testi 1



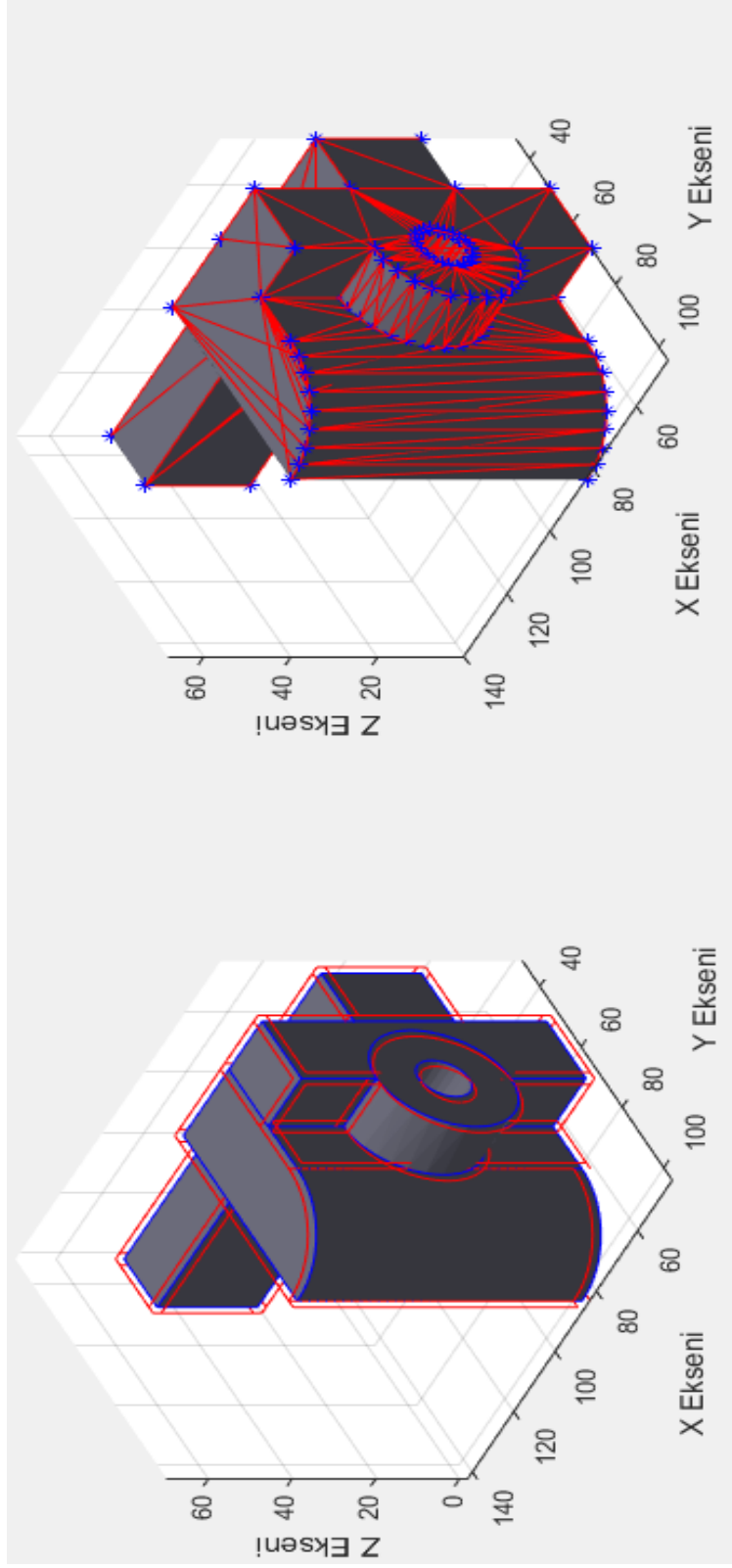
Şekil 4.30. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörüngeyi doğruluk testi 2



Şekil 4.31. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörüngenin doğruluk testi 3



Şekil 4.32. Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörüngenin doğruluk testi 4



Şekil 4.33. Sol tarafta Şekil 4.25'e ait çizim için robot kolunun uç elemanının izleyeceği yörünge ve sağ tarafta uç elemanının izleyeceği yörüngeyi köşe noktaları

Line	x	y	z
1			
2	140.16	31.49	22
3	140.16	31.49	46
4	140.16	31.49	22
5	140.16	31.49	46
6	45.161	31.49	46
7	140.16	31.49	46
8	140.16	31.49	22
9	140.16	31.49	46
10	45.161	31.49	46
11	140.16	31.49	46
12	45.161	31.49	22
13	140.16	31.49	22
14	140.16	31.49	22
15	140.16	31.49	46
16	45.161	31.49	46
17	140.16	31.49	46
18	45.161	31.49	22
19	140.16	31.49	22
20	45.161	31.49	46
21	45.161	31.49	22
22	140.16	31.49	22
23	140.16	31.49	46
24	45.161	31.49	46
25	140.16	31.49	46
26	45.161	31.49	22
27	140.16	31.49	22
28	45.161	31.49	46
29	45.161	31.49	22
30	140.16	49.49	20

Line	x	y	z
1			
2	61.161	49.49	22
3	61.161	49.49	3.4153e-06
4	83.161	49.49	22
5	61.161	49.49	3.4153e-06
6	83.161	49.49	3.4153e-06
7	83.161	49.49	22
8	83.161	49.49	46
9	83.161	49.49	68
10	61.161	49.49	46
11	83.161	49.49	68
12	61.161	49.49	68
13	61.161	49.49	46
14	61.161	66.937	28.204
15	61.161	66.937	39.796
16	61.161	68.49	28
17	61.161	66.937	39.796
18	61.161	66.937	28.204
19	61.161	65.49	39.196
20	61.161	64.248	29.757
21	61.161	64.248	38.243
22	61.161	65.49	28.804
23	61.161	64.248	38.243
24	61.161	64.248	29.757
25	61.161	63.294	37
26	61.161	62.695	32.447
27	61.161	62.695	35.553
28	61.161	63.294	31
29	61.161	62.695	32.447
30	61.161	62.49	34

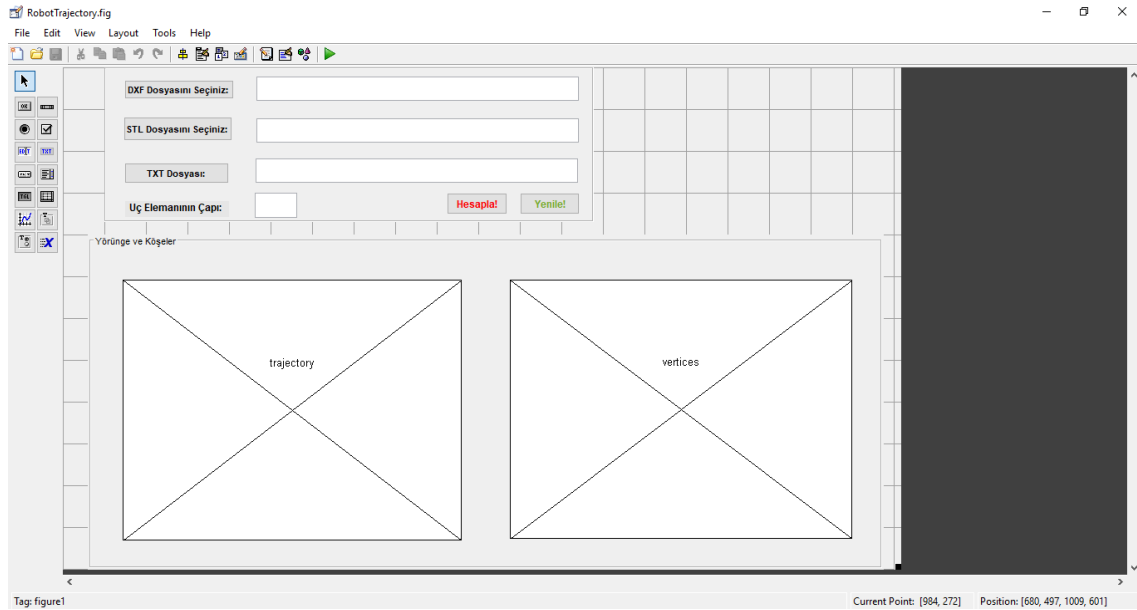
Şekil 4.34. Sol tarafta Şekil 4.25.'e ait çizim için robot kolunun uç elemanının izleyeceği yörüngenin X, Y ve Z noktalarının başlangıç kısmı, sağ tarafta bu yörüngenin köşe noktalarının başlangıç kısmı

Bu çalışma sırasında Matlab'da kodlanan fonksiyonlar robot kolunun uç elemanının, bir cismin etrafında dolaşacak yörüngeyi yaratabilmesi için dört adet girdiye ihtiyaç duyar. Bu girdiler sırasıyla, cisme ait DXF dosyasının bilgisayarda kaydedildiği yol ve DXF dosyasının ismi, STL dosyasının bilgisayarda kaydedildiği yol ve STL dosyasının ismi, yaratılan yörüngenin ve köşe noktalarının kaydedileceği TXT dosyalarının bilgisayarda kaydedileceği yol ve robot kolunun uç elemanının çap bilgileridir. Bu girdiler bir kullanıcı tarafından "*pathTrajectory.m*" fonksiyonunda bulunan bu girdilerin istendiği kod bloğuna eklenmesi gerekir. Kullanıcının bu yükünü hafifletmek için Matlab'da GUIDE (Graphical User

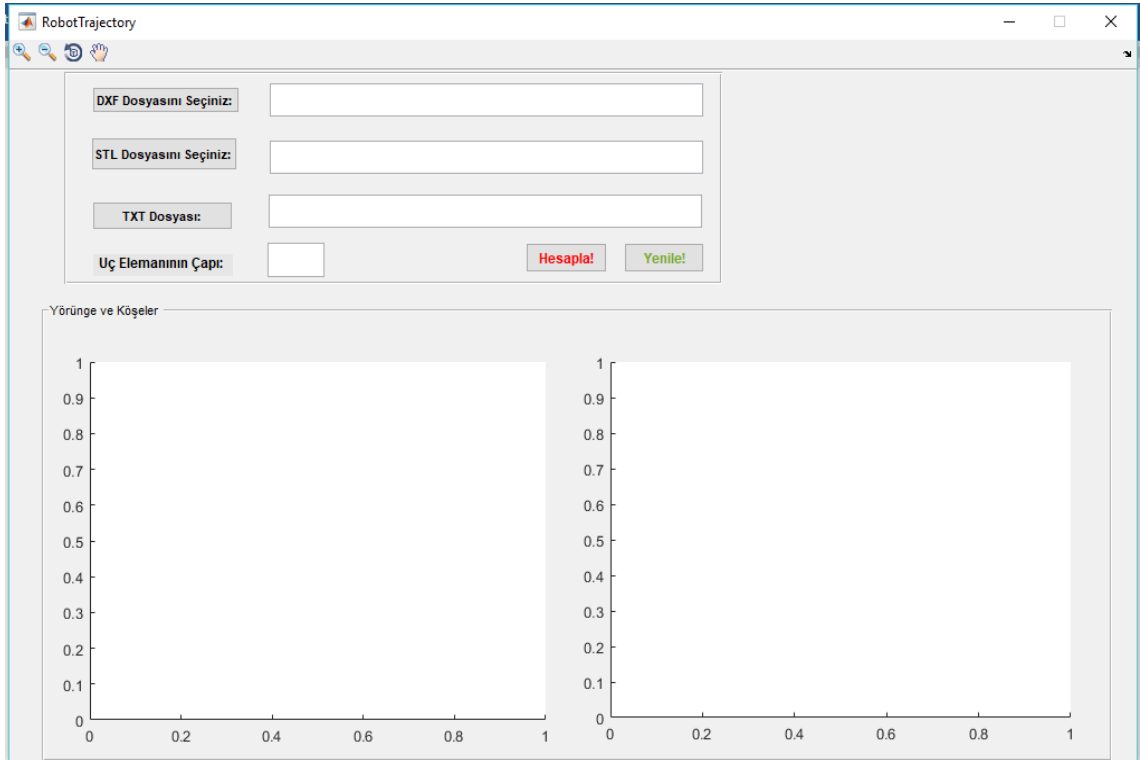
Interface Development Environment), grafiksel kullanıcı ara yüzü geliştirme ortamı kullanılarak bir ara yüz geliştirilmiştir. Bu ara yüz aracılığıyla, ara yüzde bulunan düğmelere tıklanarak bilgisayardan gerekli olan dosyalar seçilip robot kolunun uç elemanının çap bilgisi girildikten sonra “*Hesapla!*” düğmesine tıklanarak cisme ait yörünge çıkarılır. Kullanıcı yeni bir cisme ait yörünge yaratmak istediğinde “*Yenile!*” düğmesine tıklayarak bir önceki cisme ait girdileri temizler ve yeni cisme ait girdileri ara yüze ekler.

Kullanıcının yükünü hafifletmek için Matlab’da ara yüz tasarlanırken, Matlab komut satırına “*guide*” komutu yazılarak GUIDE yerleşim düzenleyicisi açılır. Yerleşim düzenleyicisi aracılığıyla ara yüz grafiksel olarak tasarlanır. (Şekil 4.35. ve Şekil 4.36.) Tasarlanan bu grafiksel ara yüz GUIDE tarafından, uygulamanın davranışını programlamak için değiştirilecek ara yüzü oluşturmak için Matlab kodunu otomatik olarak oluşturur. Oluşturulan bu Matlab koduna yerleşim düzenleyicisinde yer alan grafiklerin davranışlarını belirleyecek kodlar yazılır. Böylelikle ara yüz ve fonksiyonlar arasındaki ilişki tanımlanmış olur.

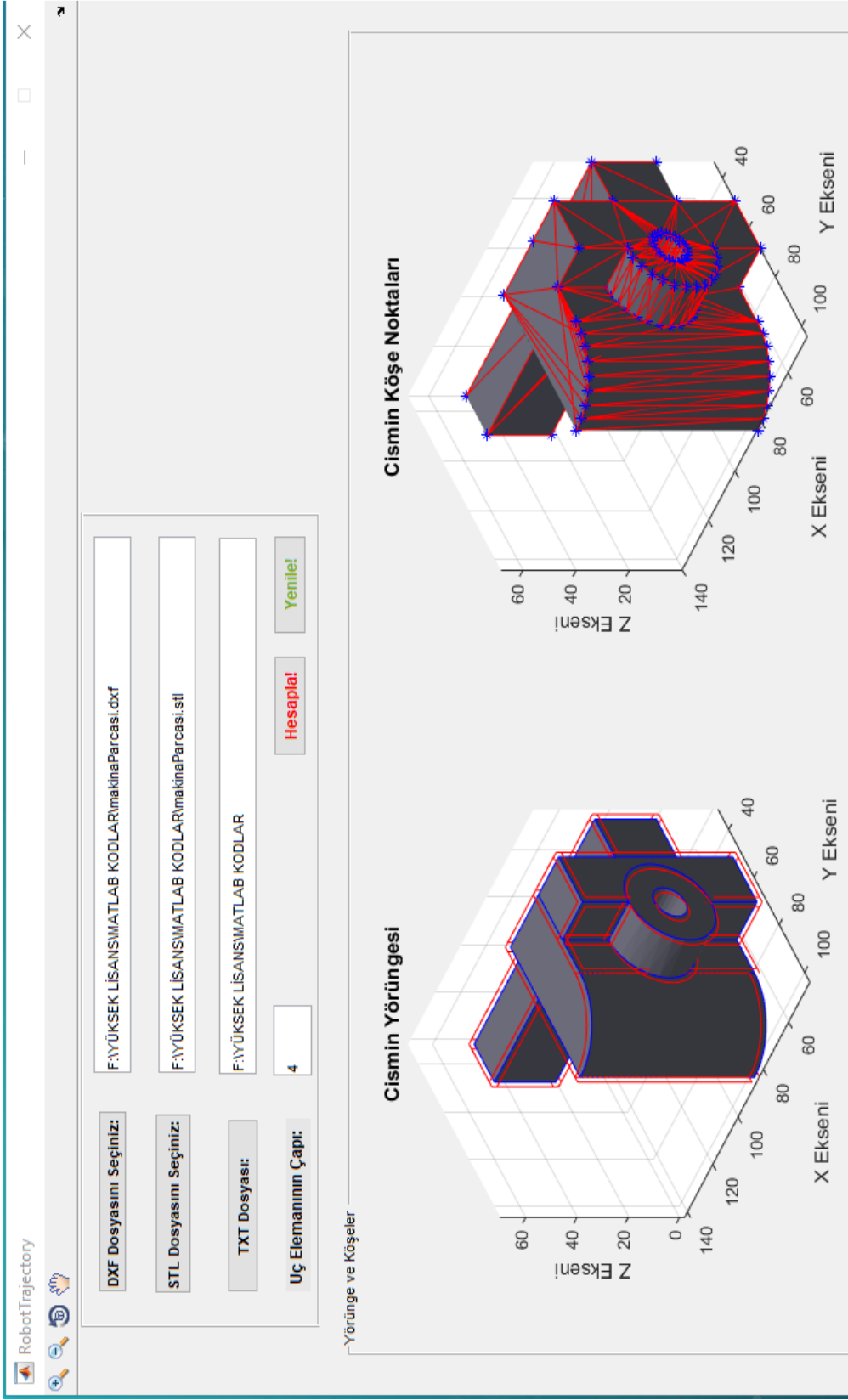
Sonuç olarak tasarlanan bu ara yüz aracılığıyla, kullanıcı tarafından girilen girdilerle birlikte robot kolunun uç elemanının izleyeceği yörünge yaratılmış olur. (Şekil 4.37.)



Şekil 4.35. Matlab’da yerleşim düzenleyicisi aracılığıyla tasarlanan grafiksel ara yüz 1



Şekil 4.36. Matlab'ta yerleşim düzenleyicisi aracılığıyla tasarlanan grafiksel ara yüz 2



Şekil 4.37. Tasarlanan ara yüz aracılığıyla yörünge oluşturulması



## 5. BULGULAR VE TARTIŞMA

Çalışmanın bu bölümünde, robot kolunun uç elemanının izleyeceği yörünge koordinatlarının robot kolunun üzerinde uygulanabilir olabilmesi için robot pano bağlantıları, robot üzerinde uygulanacak kontrol şekline göre yapılır. Çalışmanın son adımında servo sürücü parametreleri kontrol şekline uygun bir şekilde ayarlanarak her bir robot eklemi için hız, ivme, pozisyon, PID değişkenleri belirlenir. Böylelikle, 5 eksenli SCARA tipi çapak alma robotunun kontrolü sağlanmış olur.

### 5.1. Donanım

Şekil 5.1.'de gösterilmiş olan 5 eksenli çapak alma robot kolunun kontrol panelinde 5 adet ADTECH firmasına ait QS7 serisi AC 220 volt servo sürücü yer almaktadır. Sürücülerin içerisinde 2500 çizgi artımlı fotoelektrik enkoder bulunmaktadır. Robotun hareket komutlarının üretileceği hareket kontrol kartı Trio Motion Technology firmasına ait MC508'dir. 5 eksenli çapak alma robot kolunun üzerinde 5 adet servo motor bulunmaktadır. Lineer hareket eden motor üzerinde frenleme sistemi mevcuttur. Frenleme mekanizmasına 24 volt verilerek freni bırakması sağlanır. Bu motorun sürücüsünde frenleme işleminin gerçekleştirilebilmesi için gerekli bağlantılar pano üzerinde mevcuttur. Pano üzerinde motorların anlık durdurulabilmesi için bir adet acil butonu vardır. Sistemin gücünü açıp kapatmak için bir de anahtar bulunmaktadır.

Sürücüler:

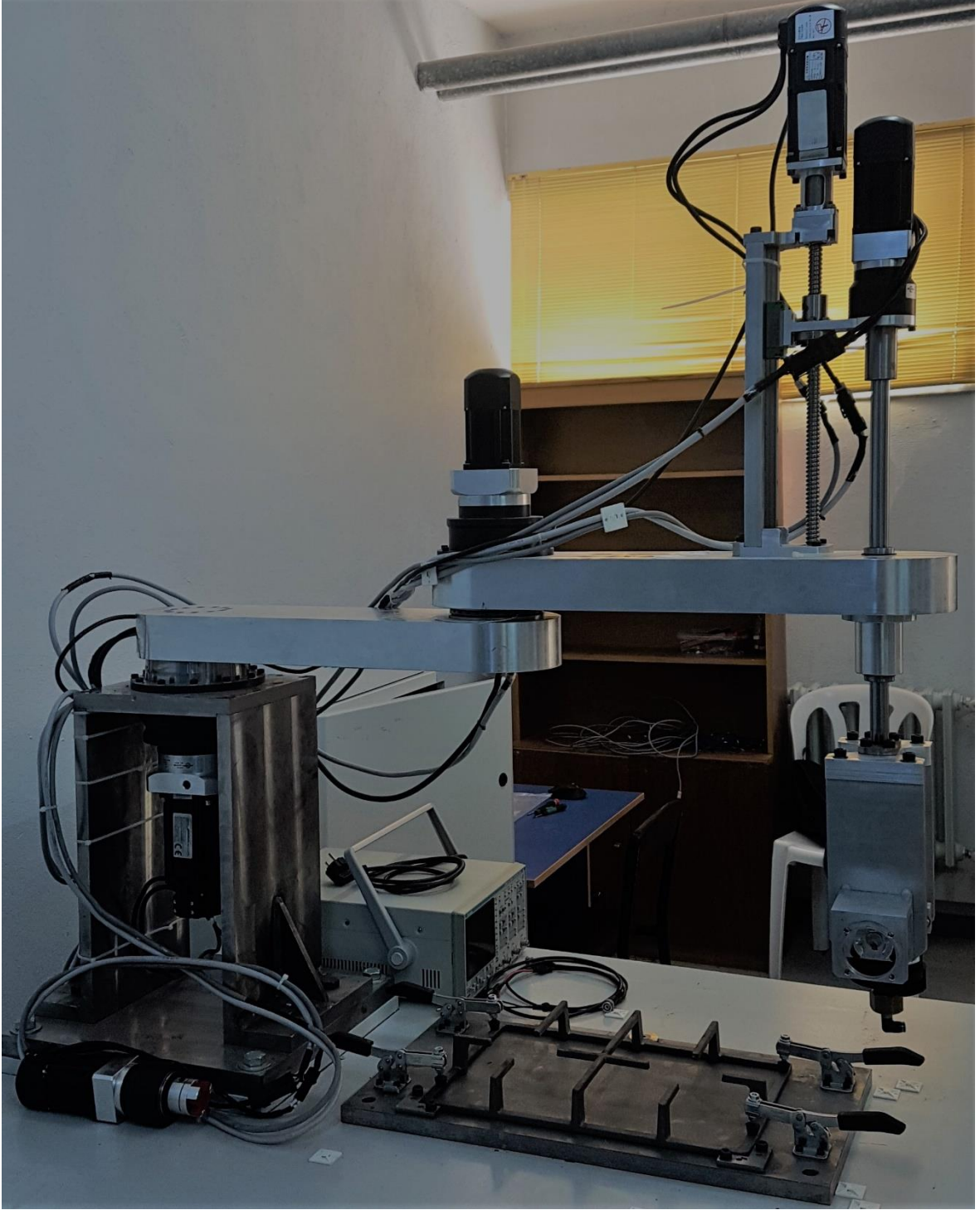
- 1 adet Adtech QS7AA030M servo sürücü (220 volt 30A)
- 4 adet Adtech QS7AA010M servo sürücü (220 Volt 10A)

Hareket kontrol kartı:

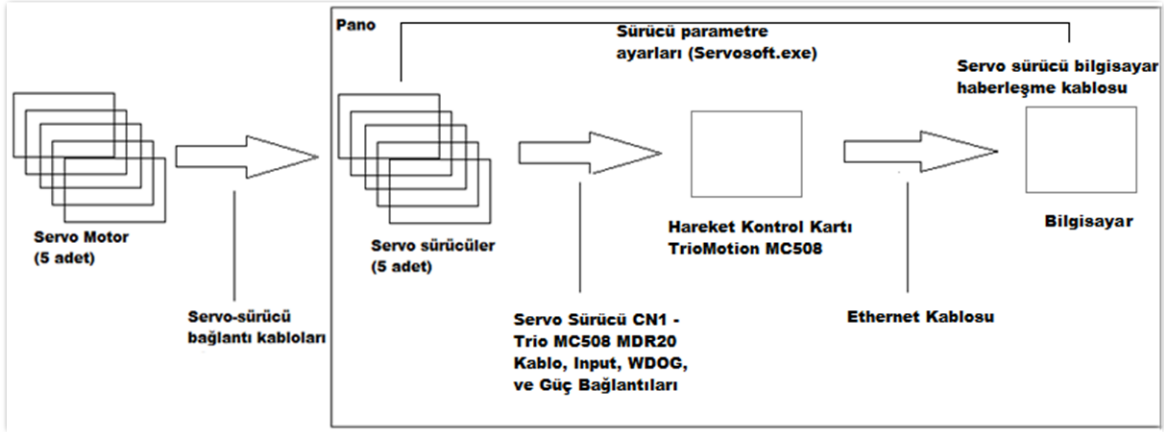
- 1 adet TrioMotion MC508 hareket kontrol kartı

Bilgisayar: Sürücülerle haberleşme bilgisayar üzerine kurulan *MotionPerfect* uygulaması üzerinden gerçekleştirilerek robot üzerinde bulunan servo motorların her birisi için pozisyon, hız, ivme, PID değerleri gibi değişkenlerin ayarlamaları sağlanır. Aynı

zamanda sürücü parametrelerinin bilgisayar üzerinden deęiştirilebilmesi için *Servosoft.exe* uygulaması kullanılmıřtır. Pano içerik tanımını Şekil 5.2.'de gösterilmiřtir.



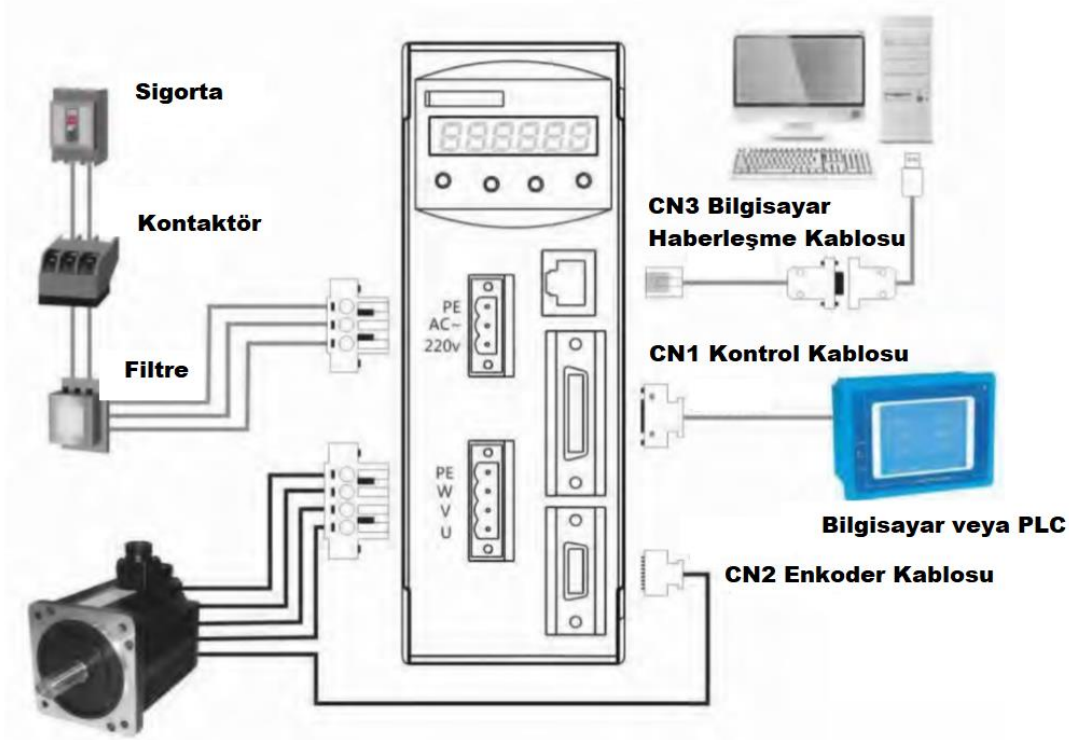
Şekil 5.1. 5 eksenli apak alma robot kolu



Şekil 5.2. Pano içerik tanımı

### 5.1.1. Servo sürücü bağlantıları

Şekil 5.3.'de servo sürücü üzerinde bulunan haberleşme kablosu, kontrol kablosu, enkoder kablosu, motoru besleyen güç kablosu ve sürücüyü besleyen güç kablosunun bağlantı şeması gösterilmiştir. Çizelge 5.1.'de sürücü üzerindeki bağlantı portlarının isim ve özellikleri verilmiştir.



Şekil 5.3. Sürücü bağlantı şeması (Adtech Technology Co., 2012)

Bağlantı Kısaltması	Bağlantı ismi	Özellikleri
R, T	Ana güç girişi	Sürücünün ana güç girişi
PE	Toprak bağlantısı	Motor harici toprak bağlantısı
U, V, W	Motor güç hattı	Motor akım sağlayıcısı
CN1	Kontrol bağlantısı	Kontrolcü veya PLC bağlantısı
CN2	Enkoder bağlantısı	Enkoder iade sinyali
CN3	Haberleşme kablosu	Bilgisayar haberleşme bağlantısı (RJ45 portu)

Çizelge 5.1. Sürücü üzerinde yer alan portlar ve özellikleri (Adtech Technology Co., 2012)

CN1 kontrol bağlantısı: CN1 portu sürücü ve hareket kontrol kartı arasındaki bağlantıların yapıldığı porttur. Bağlantıların yapılabilmesi için 36 pin MDR konektörün çoklu kablolarla lehim işlemlerinin özelliklerine göre yapılması gerekmektedir.(Şekil 5.4. ve Çizelge 5.2.)

18	16	14	12	10	8	6	4	2
17	15	13	11	9	7	5	3	1
36	34	32	30	28	26	24	22	20
35	33	31	29	27	25	23	21	19

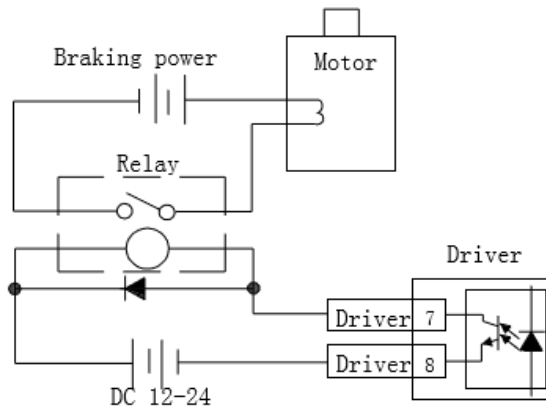
Şekil 5.4. 36 pin MDR konektörün pin dizilişi (Adtech Technology Co., 2012)

NO	İSİM	DURUM	NO	İSİM	DURUM
1	COIN+	POSITIONING COMPLETE +	19	VIN	ANALOG INPUT
2	COIN-	POSITIONING COMPLETE -	20	GND	ANALOG GROUND
3	ALM-	SERVO ALARM -	21	RESERVE	—————
4	ALM+	SERVO ALARM +	22	RESERVE	—————
5	SRDY+	SERVO READY +	23	RESERVE	INSIDE INTEGRATION 2K RES CONNECT PLC +
6	SRDY-	SERVO READY -	24	PULSE-	PULSE SIGNAL -
7	BRK+	BRAKE SIGNAL +	25	PULSE+	PULSE SIGNAL +
8	BRK-	BRAKE SIGNAL -	26	SIGN-	DIRECTION SIGNAL
9	INCOM+	V +	27	SIGN+	DIRECTION SIGNAL
10	EN-	SERVO EN	28	CZ+	

11	INTH-	COMMAND PULSE FORBID	29	CZ-	ENCODER Z SIGNAL
12	CW-	ARE TURNING LIMIT	30	OZ+	ENCODER Z +
13	CCW-	REVERSAL LIMIT	31	OZ-	ENCODER Z -
14	CLR-	CLR -	32	OB+	ENCODER B +
15	MODE-	FUNCTION SELECT	33	OB-	ENCODER B-
16	0V	Internal power for PLC 12V 100mA	34	OA+	ENCODER A +
17	12V+		35	OA-	ENCODER A-
18	RESERVE	—————	36	RESERVE	—————

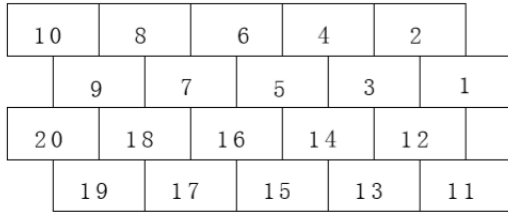
Çizelge 5.2. CN1 portundaki pinlerin numaraları ve görevleri (Adtech Technology Co., 2012)

Dikey konumlandırılmış motorun üzerinde frenleme mekanizması bulunmaktadır. Frenleme mekanizması dikey olan motorun düşmesini engellemek için servo sürücünün CN1 portunda yer alan 7. (Brake Signal+) ve 8. (Brake Signal-) pinlerinin 24 VDC Solid-State bir röle aracılığıyla güç kaynağı ile arasındaki bağlantılarının doğru bir şekilde yapılması gerekmektedir. (Şekil 5.5.)



Şekil 5.5. Frenleme pin bağlantısı (Adtech Technology Co., 2012)

CN2 enkoder bağlantısı: CN2 portu sürücü ve motor arasındaki bağlantıların yapıldığı porttur. Bağlantıların yapılabilmesi için 20 pin MDR konektörün çoklu kablolarla lehim işlemlerinin özelliklerine göre yapılması gerekmektedir. (Şekil 5.6. ve Çizelge 5.3.)

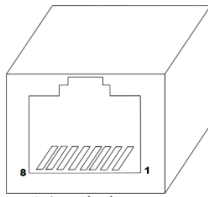


Şekil 5.6. 20 pin MDR konektörün pin dizilişi (Adtech Technology Co., 2012)

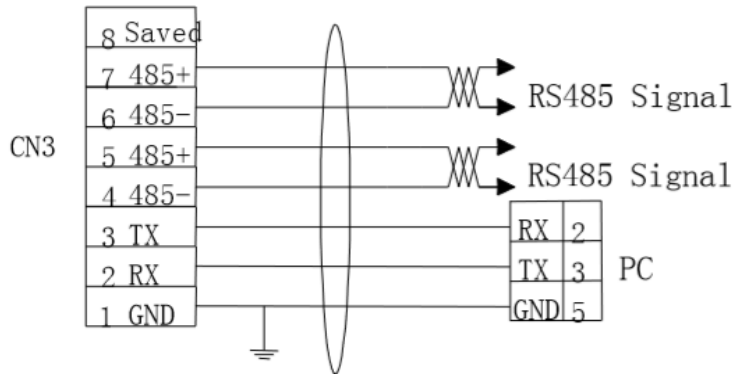
NO	İSİM	DURUM	NO	İSİM	DURUM
1	A+	PG INPUT A PHASE	11	U+	PG INPUT U PHASE
2	A-		12	U-	
3	B+	PG INPUT B PHASE	13	V+	PG INPUT V PHASE
4	B-		14	V-	
5	Z+	PG INPUT Z PHASE	15	W+	PG INPUT W PHASE
6	Z-		16	W-	
7	+5V	POWER 5V	17	0V	POWER 0V
8					
9					
10	RESERVE	—————	20	RESERVE	—————

Çizelge 5.3. CN2 portundaki pinlerin numaraları ve görevleri (Adtech Technology Co., 2012)

CN3 bilgisayar haberleşme bağlantısı: CN3 portu sürücü ve bilgisayar arasındaki bağlantıların yapıldığı porttur. Bağlantıların yapılabilmesi için 8 pin ethernet kablosunun (Şekil 5.7.) RS-232'den USB'ye dönüştürücü aparatına bağlantılarının yapılması gerekmektedir. (Şekil 5.8.)

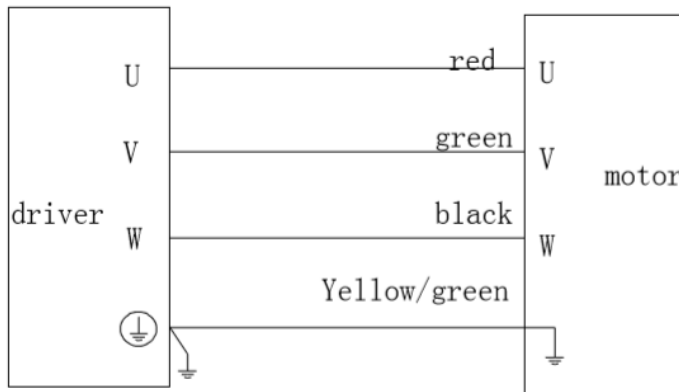


Şekil 5.7. Network Port (Adtech Technology Co., 2012)



Şekil 5.8. CN3 portu ve bilgisayar arasındaki bağlantı şeması (Adtech Technology Co., 2012)

Sürücü ve motor arasındaki güç bağlantısı: Robot üzerinde yer alan motorun beslemesi, sürücüsü üzerinde yer alan U, V, ve W portlarından yapılır. PE portu topraklama işleminin yapıldığı porttur. (Şekil 5.9.)



Şekil 5.9. Sürücü ve motor arasındaki güç bağlantısı (Adtech Technology Co., 2012)

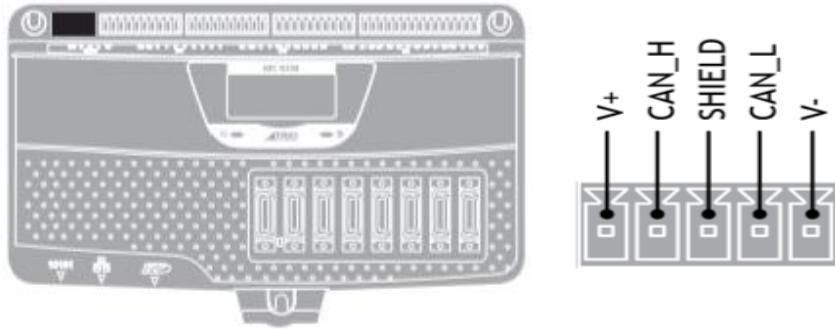
### 5.1.2. Hareket kontrol kartı bağlantıları

Hareket kontrol kartı MC508 (Şekil 5.10.), Trio'nun ARM Coretex-A9 ® çift duyarlıklılı teknolojisine dayanır ve 8 eksenli servo veya 8 - 16 eksenli step motorlar veya "pulse-input" servo sürücüler için sinyal ve yön kontrolü sağlar. Trio'nun Motion Perfect uygulama yazılımını çalıştıran bir bilgisayar kullanarak robot programlanır. Programlar ve veriler doğrudan bir flash belleğe kaydedilir, böylece batarya destekli depolama ihtiyacını ortadan kaldırılmış olur. Ethernet kablosu ile doğrudan programlanabilir. (Trio Motion Technology Ltd, 2017)



Şekil 5.10. Hareket kontrol kartı MC508 (Trio Motion Technology Ltd, 2017)

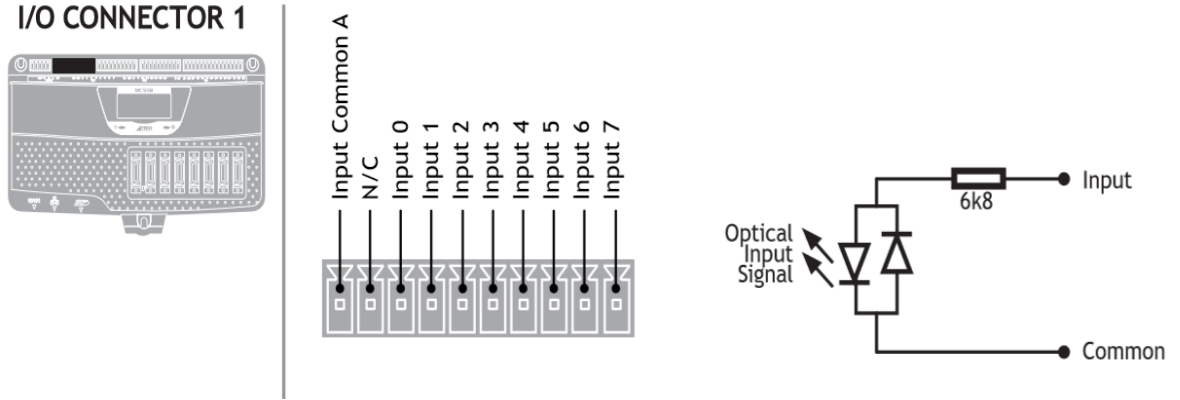
5 kanallı konektör: 5 kanallı 3.5 mm pitch konektörü MC508'e 24 Volt güç sağlamak için kullanılır. Bu 24 VDC giriş tamamen izole edilmiştir. (Trio Motion Technology Ltd, 2017) (Şekil 5.11.)



Şekil 5.11. 5 kanallı konektör ve özellikleri (Trio Motion Technology Ltd, 2017)

I/O konektör 1: 0-7 arasındaki herhangi bir giriş herhangi bir eksenin girişi olarak eşlenebilir. PNP işlemi için ortak (common) uç 0 Volt'a bağlanır. NPN işlemi için ortak (common) uç 24 Volt'a bağlanır. (Trio Motion Technology Ltd, 2017) (Şekil 5.12.)

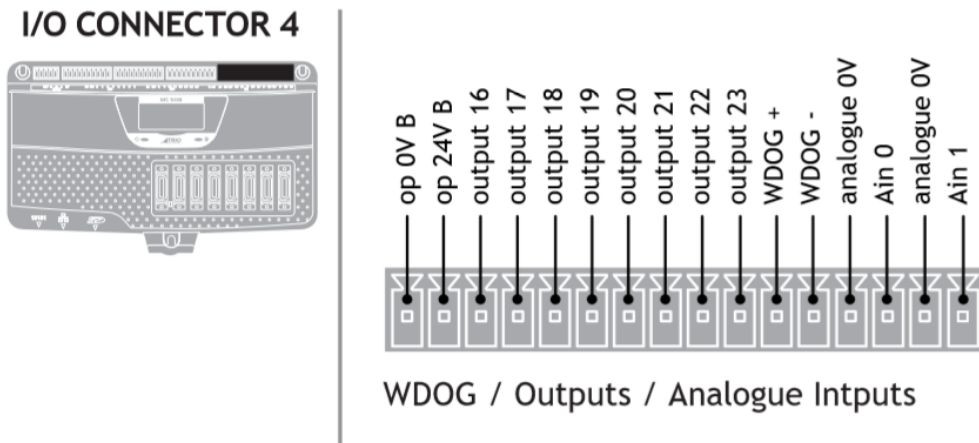




Şekil 5.12. I/O konektör 1 özellikleri (Trio Motion Technology Ltd, 2017)

I/O Konektör 4: Çıkışları, analog girişleri ve eksen konektörü üzerindeki DAC çıkışlarını çalıştırmak için 24V uygulanmalıdır. (Şekil 5.13.) (Trio Motion Technology Ltd, 2017)

Denetleyici doğru şekilde açıldığında ve sistem ve uygulama yazılımı hazır olduğunda harici amplifikatörleri etkinleştirmek için dahili bir solid-state rölesi kullanılır. Kontrol rölesinde güç yoksa, röle kontağı açık devre olacaktır veya kullanıcı programı *WDOG = OFF* komutu ile açık olarak ayarlar. Her eksen ayrıca, uygulama programı tarafından belirlenen ana röle ile birlikte çalışan bağımsız bir röle çıkışına sahiptir. (Trio Motion Technology Ltd, 2017)

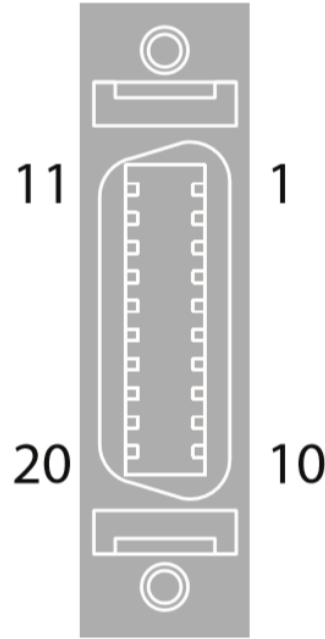


Şekil 5.13. I/O konektör 4 özellikleri (Trio Motion Technology Ltd, 2017)

**MDR20 Konektörü:** MDR20 konektöründeki pinler doğrudan sürücü ile ilişkilendirilir. MDR20 konektörünün pin özelliklerine göre sürücü ile bağlantısı yapılır. (Şekil 5.14.)

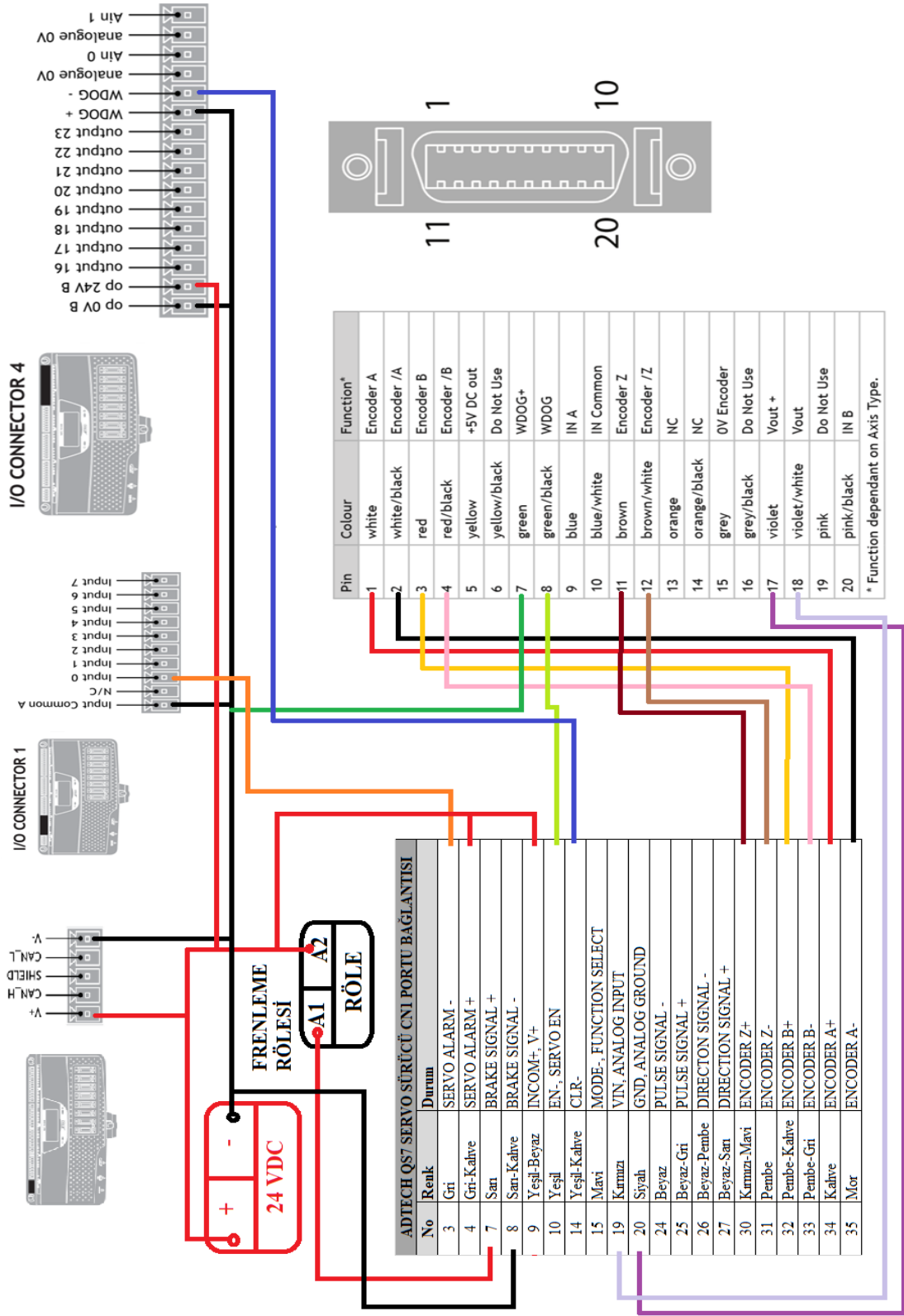
Pin	Colour	Function*
1	white	Encoder A
2	white/black	Encoder /A
3	red	Encoder B
4	red/black	Encoder /B
5	yellow	+5V DC out
6	yellow/black	Do Not Use
7	green	WDOG+
8	green/black	WDOG
9	blue	IN A
10	blue/white	IN Common
11	brown	Encoder Z
12	brown/white	Encoder /Z
13	orange	NC
14	orange/black	NC
15	grey	0V Encoder
16	grey/black	Do Not Use
17	violet	Vout +
18	violet/white	Vout
19	pink	Do Not Use
20	pink/black	IN B

\* Function dependant on Axis Type.

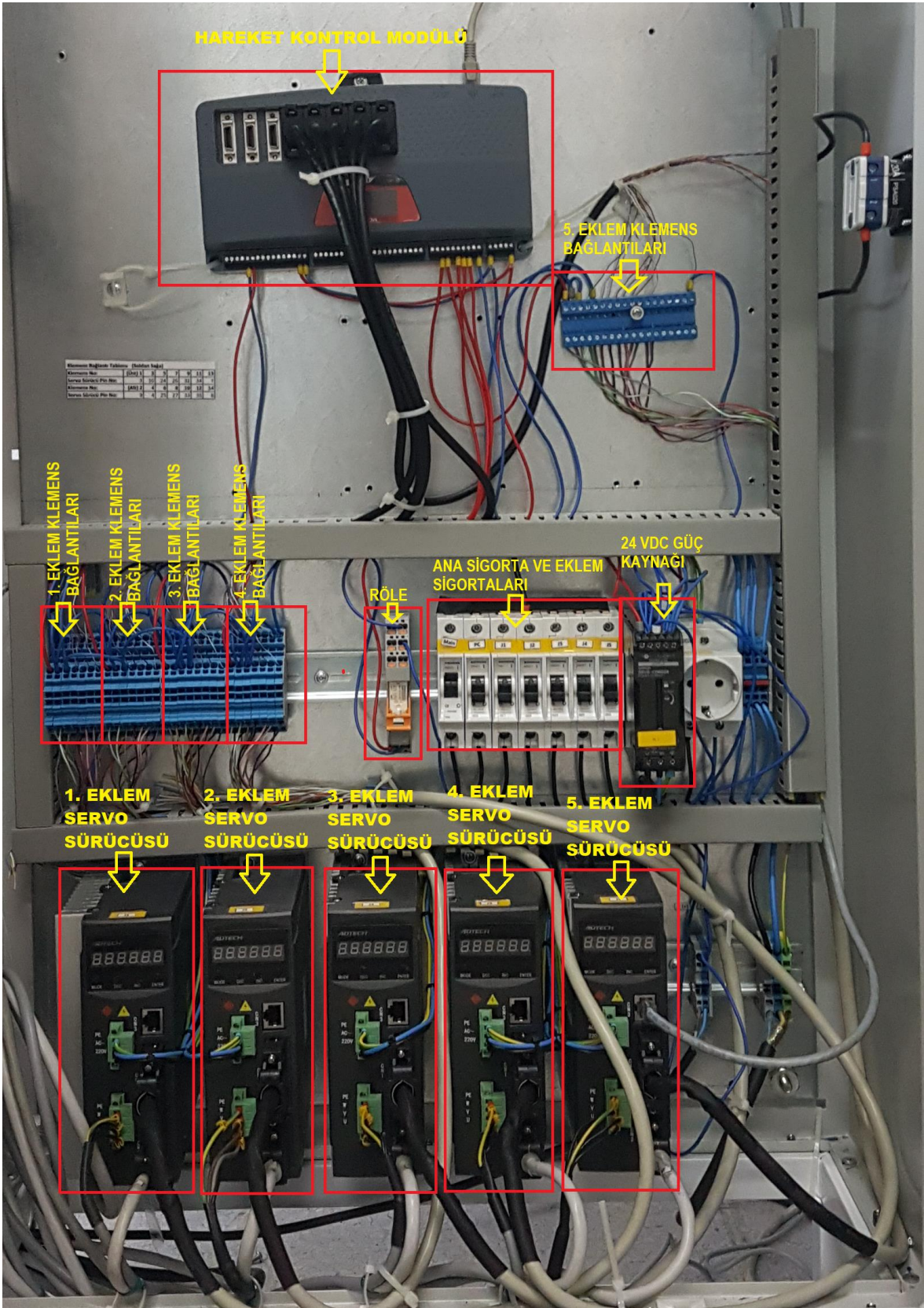


Şekil 5.14. MDR20 konektörü pin ve özellikleri (Trio Motion Technology Ltd, 2016)

5 eksenli çapak alma robotunun servo sürücü ve hareket kontrol kartı ile yapılmış olan tek eklem için bağlantıları Şekil 5.15.'de gösterilmiştir. Robotun kontrol pano elemanları Şekil 5.16.'da sınıflandırılmış ve robota ait her bir eklem için klemens bağlantıları Şekil 5.17. ve Şekil 5.18'de görselleştirilmiştir.

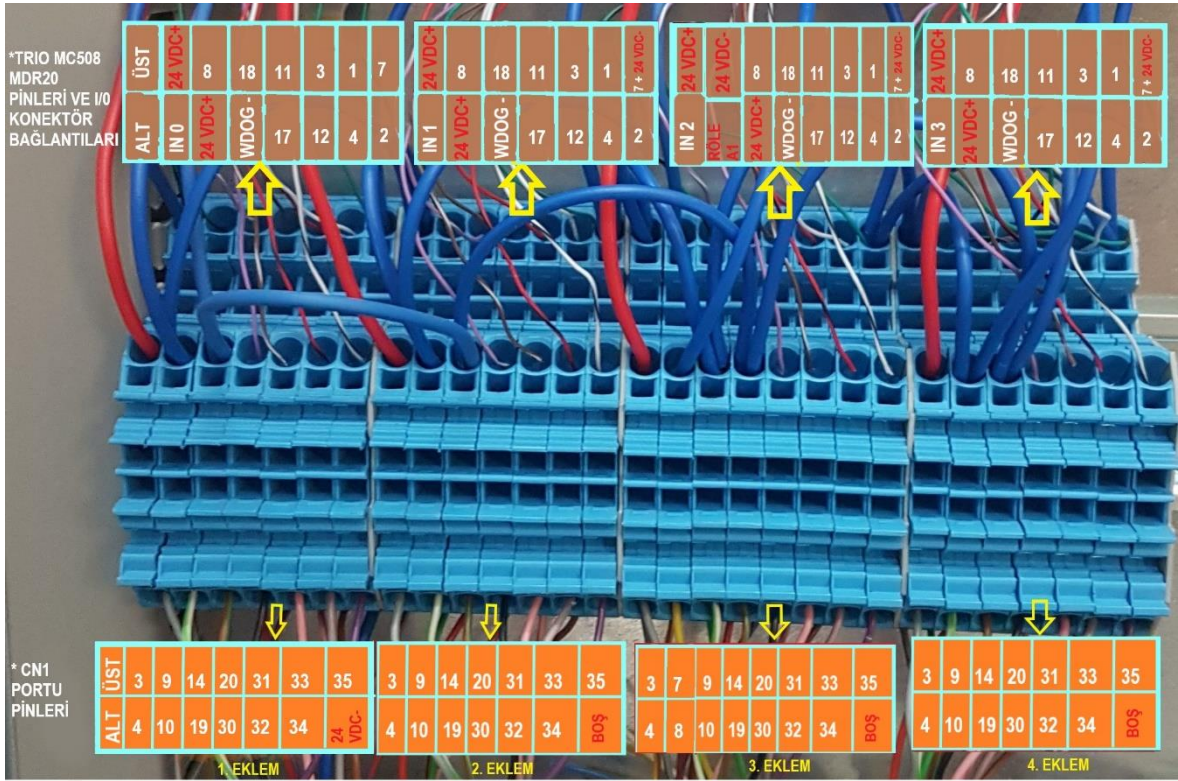


Şekil 5.15. 5 eksenli çapak alma robotunun tek eklem için bağlantı şeması

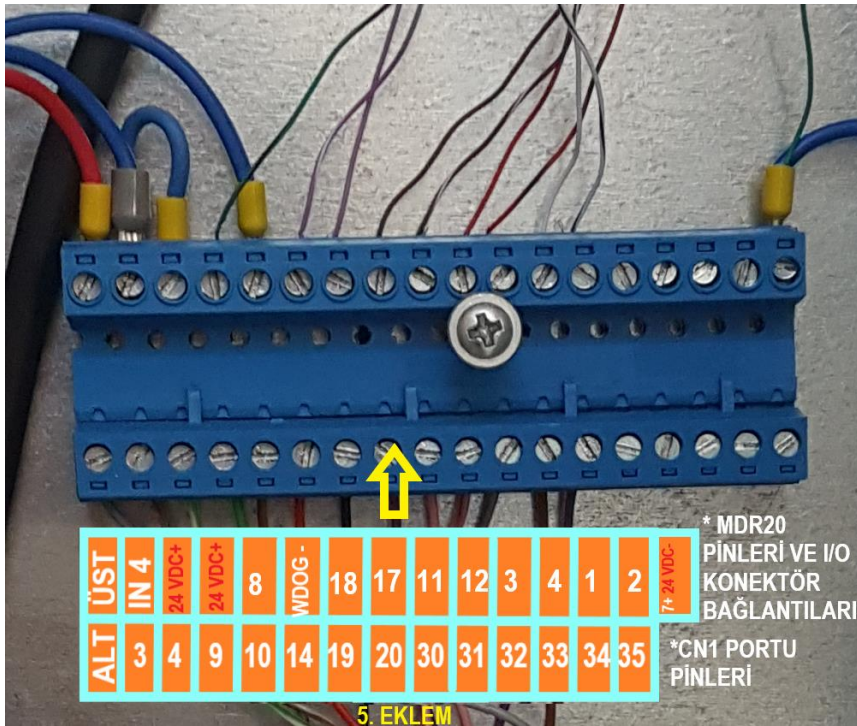


Şekil 5.16. 5 eksenli çapak alma robotunun kontrol pano elemanları





Şekil 5.17. 1., 2., 3. ve 4. eklem için klemens bağlantıları



Şekil 5.18. 5. eklem için klemens bağlantıları

## 5.2. Sürücü Değişkenleri

Sürücü parametreleri iki şekilde kullanıma göre ayarlanabilir. Birincisi, servo motor sürücüsü üzerinden parametre ayarları teker teker girilip kaydedilir. İkincisi, *Servosoft.exe* uygulaması ile sürücünün CN3 portuyla bilgisayardan haberleşmesi sağlanarak parametreler değiştirilebilir. Sürücü parametreleri aşağıdaki tablolarda parametre numarası, parametre ismi, uygulama modu, parametre aralığı, fabrika ayarı, parametre birimi gibi özellikleri dikkate alınarak sınıflandırılmıştır. (Şekil 5.19.)

Parameter No.	Parameter Name	Application Mode	Parameter Range	Factory Default	Unit	Remark
P0	Software version	P, S, T	2015—2050			②
P1	Parameter password	P, S, T	0-9999	0		①
P2	Motor model	P, S, T		400	Motor power	①
P3	Boot display	P, S, T	0—10	0		①
P4	Control mode	P, S, T	0—7	0		①
P5	Servo enable control	P, S, T	0—1	0		①
P6	Servo input signal INTH function	P, S, T	0—1	0		①
P7	Limit input control	P	0—4	0		①
P8	Coin output mode	P, S, T	0—1	0		①
P9	Alarm output mode	P, S, T	0—1	0		①
P10	Pulse mode	P	0—2	0		①
P11	Motor direction	P, S	0—1	0		①
P12	Electronic gear numerator	P, S	1—32000	1		①
P13	Electronic gear denominator	P, S	1—32000	1		①
P14	Positioning completion scope	P, S	0—32000	5	Pulse	①
P15	Position deviation alarm range	P, S	0—32000	0	Pulse	①
P16	Position gain	P, S	1—2000	100		①
P17	Position feed-forward	P, S	0—32000	0		①
P18	Position smoothing constant	P, S	0—1000	0		①
P19	Position acceleration time	P, S	0—32000	0		①
P20	Position deceleration time	P, S	0—32000	0		①
P21	Speed gain	P, S	1—1000	50		①
P22	Speed integral	P, S	1—32000	10		①
P23	Acceleration time (speed)	S	0 — 32000(ms)	100	ms	①
P24	Deceleration time	S	0 — 32000(ms)	100	ms	①

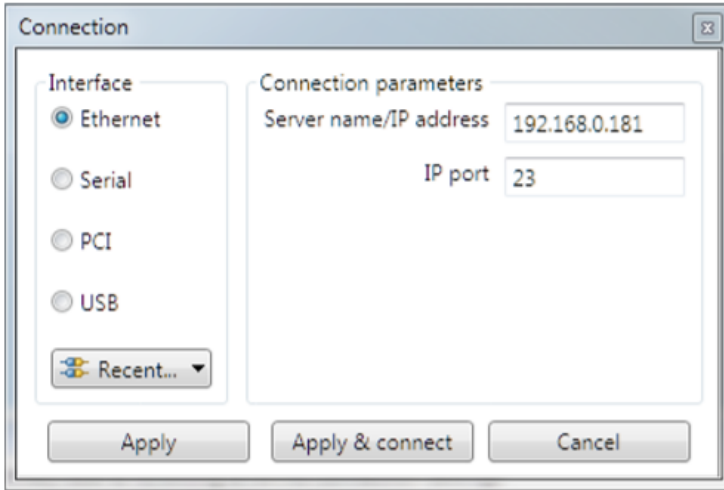
P25	Analog input method	S, T	0-1	0		①
P26	Analog Max. speed	S	1-5000	2000	r/min	①
P27	Torque Max. speed	T	1-5000	2000	r/min	①
P28	Analog input filter coefficient	S, T	0-1000	0		①
P29	Analog input voltage at zero	S, T		0		①
P30	Inertia ration	P, S, T	0-1000	0		①
P31	Analog input percentage	S, T	0-500	0	%	①
P32	Encoder lines frequency splitting	P, S, T	0-127	0		③
P33	Encoder alarm permit	P, S, T	0-1	0		①
P34	JOG speed	S	0-5000	1000		①
P35	Internal speed 1	S	0-5000	100	r/min	①
P36	Internal speed 2	S	0-5000	200	r/min	①
P37	Internal speed 3	S	0-5000	300	r/min	①
P38	Internal speed 4	S	0-5000	400	r/min	①
P39	Internal position 1	P	0-±32000	100	Pulse	①
P40	Internal position 2	P	0-±32000	200	Pulse	①
P41	Internal position 3	P	0-±32000	300	Pulse	①
P42	Internal position 4	P	0-±32000	400	Pulse	①
P43	Communication address	P, S, T	0-255	0		①
P44	Communication baud rate	P, S, T		9600		①
P45	Torque reaching percentage	P, S, T	0-100	100	%	①
P46	Torque percentage of motor stationary	P,	0-100	0	%	①
P47	Start delay of electromagnetic brake	P, S, T	0-3200 (ms)	0	ms	①
P48	Stop delay of electromagnetic brake	P, S, T	0-3200 (ms)	0	ms	①
P49	Zero speed clamp-on					
P50	Current loop gain	P, S, T	10-4000	600		①
P51	Current loop integral	P, S, T	1-2000	150		①
P52	Encoder lines	P, S, T	1000-6000	2500		③
P53	Encoder type	P, S, T	0-1	0		③
P54	Pole-pairs	P, S, T	2-6	4		③
P55	Drift angle	P, S, T	0-2500	2360		③
P56	Rated current	P, S, T	0-100	28	0.1A	③
P57	Rated torque	P, S, T	0-200	13	0.1NM	③
P58	second electronic gear ratio numerator	P	0-32000			①
P59	second electronic gear ratio denominator	P	0-32000			①

Şekil 5.19. Sürücü parametreleri (Adtech Technology Co., 2012)

### 5.3. Robot Üzerinde Hız Kontrolünün Uygulanması

Sistem parametreleri  $P4 = 2$  veya  $P4 = 4$  ve MODE geçerli olduğunda, servo sistemi hız kontrol modundadır. Maksimum çalışma hızı P26 ve P31 parametreleri ile belirlenir. Maksimum çalışma hızı, giriş voltajı 10V olduğunda çalışma hızını ifade eder. Çalışma hızı Vin1 voltajı ile belirlenir ve yön Vin1 ve P11'in sembolleri ile belirlenir.  $P15 = 2$  olduğunda, yön CW ve CCW ile belirlenir, burada CW ve CCW sırasıyla motor dönüşünü ileriye ve tersine işaret eder. Hız kontrolünün sıfırlanması P29 parametresi ile ayarlanır. Giriş voltajı 0V olduğunda motor hızını 0 olarak ayarlamak için bu parametrenin ayarlanması gerekmektedir. Hız kontrol ivme zaman sabiti P23 parametresi ile ayarlanır; hız kontrolü yavaşlama süresi sabiti P24 parametresi üzerinden ayarlanır. (Adtech Technology Co., 2012)

Bu çalışmada robot üzerinde hız kontrolünün yapılabilmesi için sürücü üzerindeki bu parametrelerin ayarlanması gerekmektedir. Bu parametreler doğru bir şekilde sürücüye kaydedildiğinde *MotionPerfect* uygulamasının hareket kontrol kartı ile ethernet kablosu üzerinden haberleşebilmesi için gerekli bağlantı değerleri girilmelidir. (Şekil 5.20.)



Şekil 5.20. IP adres ve IP port

Bilgisayar ve Trio MC508 hareket kontrol kartının bağlantısı kurulduktan sonra her bir eklemin kontrolünün sağlanabilmesi için *MotionPerfect* uygulamasında TrioBasic dili kullanılarak programlar yazılmıştır. Bunlar;



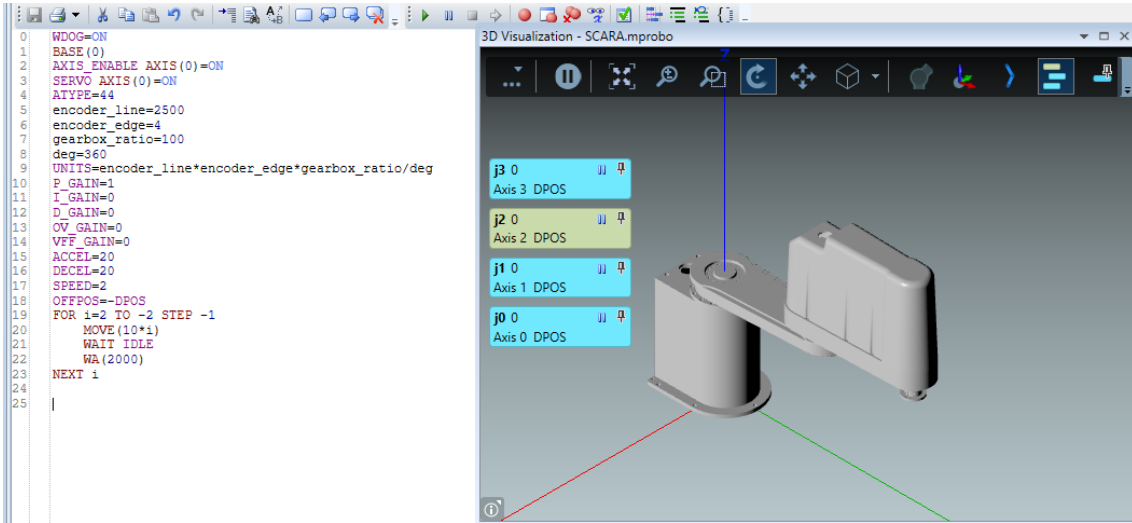
Birinci eklemin örneğin 10 derece dönebilmesi için derece cinsinden bir birim hesaplama işlemine ihtiyacı vardır. Birim dönüşümü; enkoder çözünürlüğü, enkoder kenarı, ve motor adım sayısının birbiriyle çarpılmasından sonra 360 dereceye bölünmesi ile bulunur. 5 eksenli çapak alma robotunun birinci eklemi için bu değerler aşağıdaki gibidir.

- Enkoder çözünürlüğü: 2500
- Enkoder kenarı: 4
- Motor adım sayısı: 100

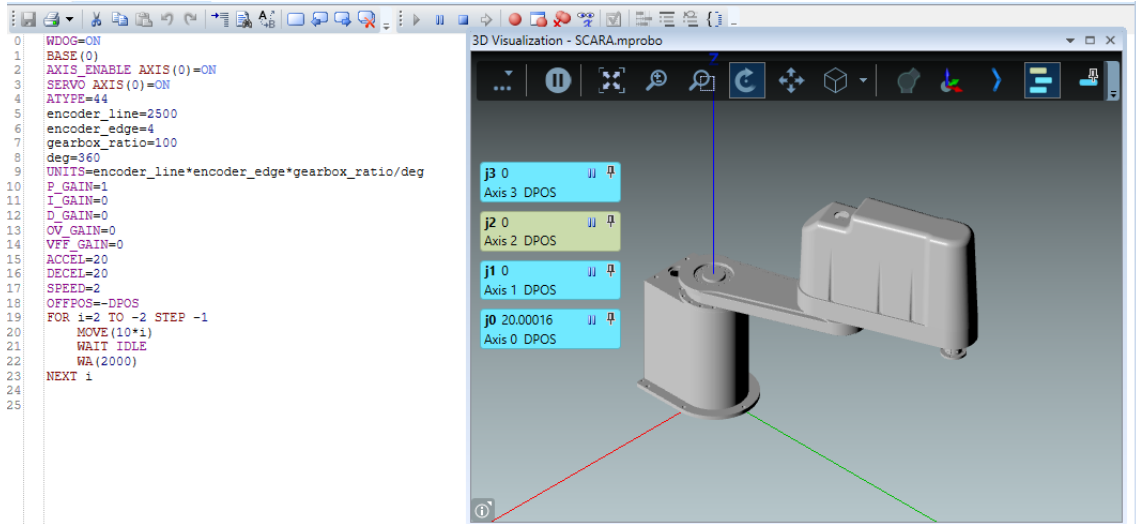
Birim dönüşümü yapıldıktan sonra motor hareketini gerçekleştirirken hangi PID, hız ve ivme değerlerini kullanacağı yazılır. Bu değerler birinci eklem için aşağıdaki gibidir.

- P\_GAIN=1
- I\_GAIN=0
- D\_GAIN=0
- OV\_GAIN=0
- VFF\_GAIN=0
- ACCEL=20
- DECEL=20
- SPEED=2

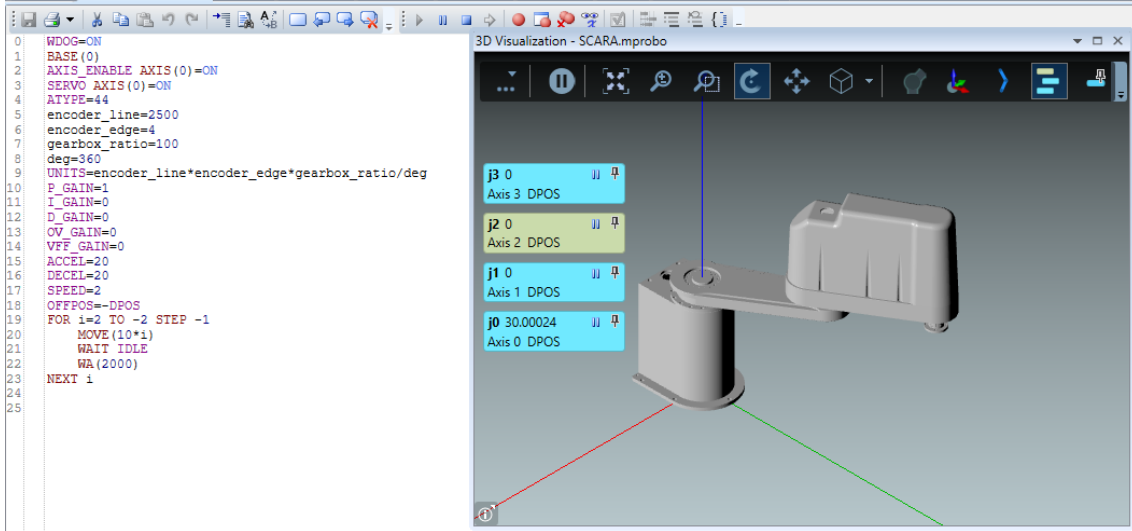
Birinci eklem için bu değerler girildikten sonra, birinci eklemden yer alan motora ev konumundan sırasıyla +20, +30, +20 ve 0 derece dönecek şekilde komut gönderilir. (Şekil 5.21., Şekil 5.22. ve Şekil 5.23.)



Şekil 5.21. Birinci eklemin ev konumu



Şekil 5.22. Birinci eklemin saat yönünde 20 derece dönüşü



Şekil 5.23. Birinci eklem için saat yönünde 30 derece dönüşü

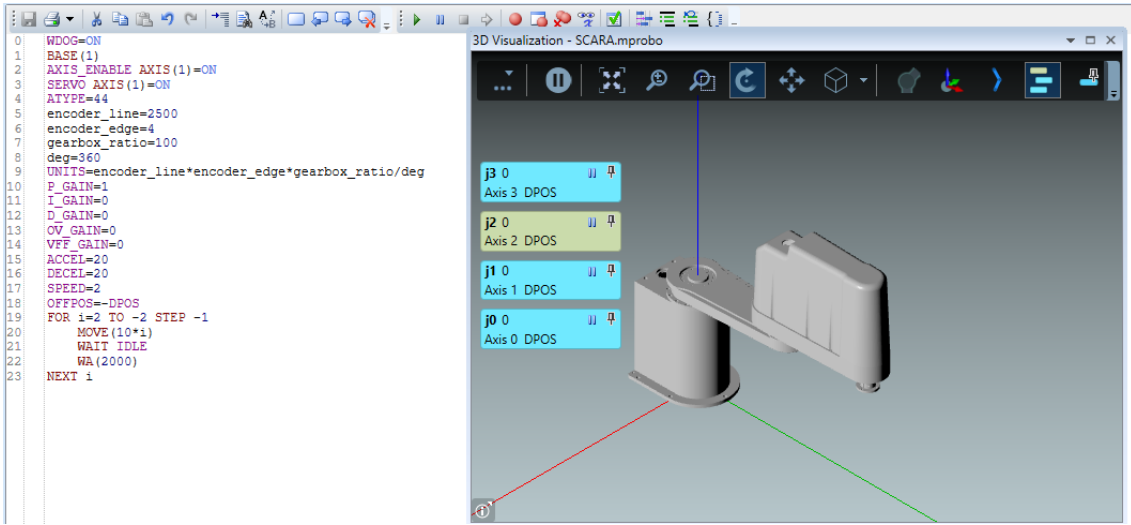
İkinci eklem için robotun birim dönüşüm değerleri aşağıdaki gibidir.

- Enkoder çözünürlüğü: 2500
- Enkoder kenarı: 4
- Motor adım sayısı: 100

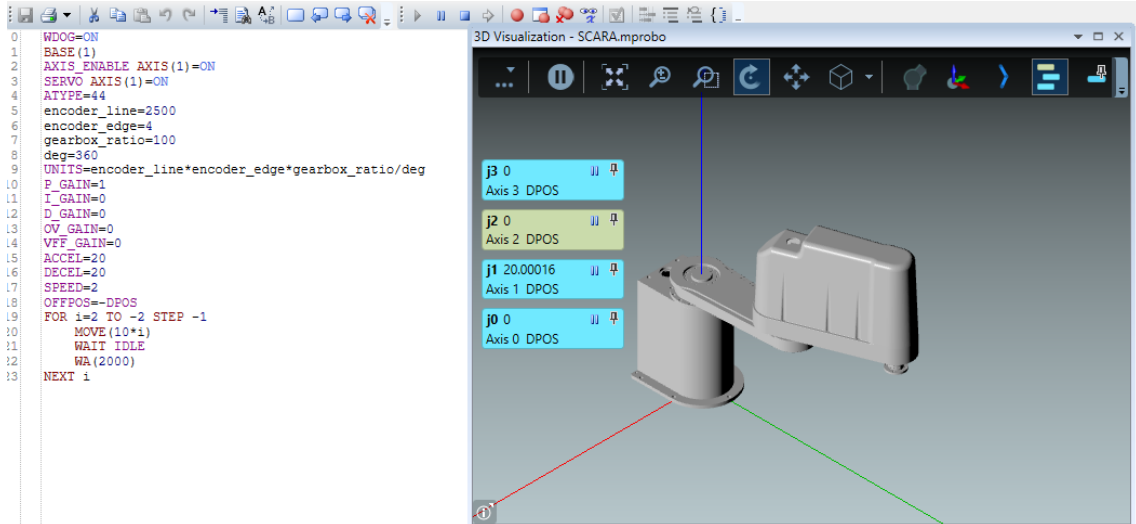
Birim dönüşümü yapıldıktan sonra motor hareketini gerçekleştirirken hangi PID, hız ve ivme değerlerini kullanacağı yazılır. Bu değerler ikinci eklem için aşağıdaki gibidir.

- P\_GAIN=1
- I\_GAIN=0
- D\_GAIN=0
- OV\_GAIN=0
- VFF\_GAIN=0
- ACCEL=20
- DECEL=20
- SPEED=2

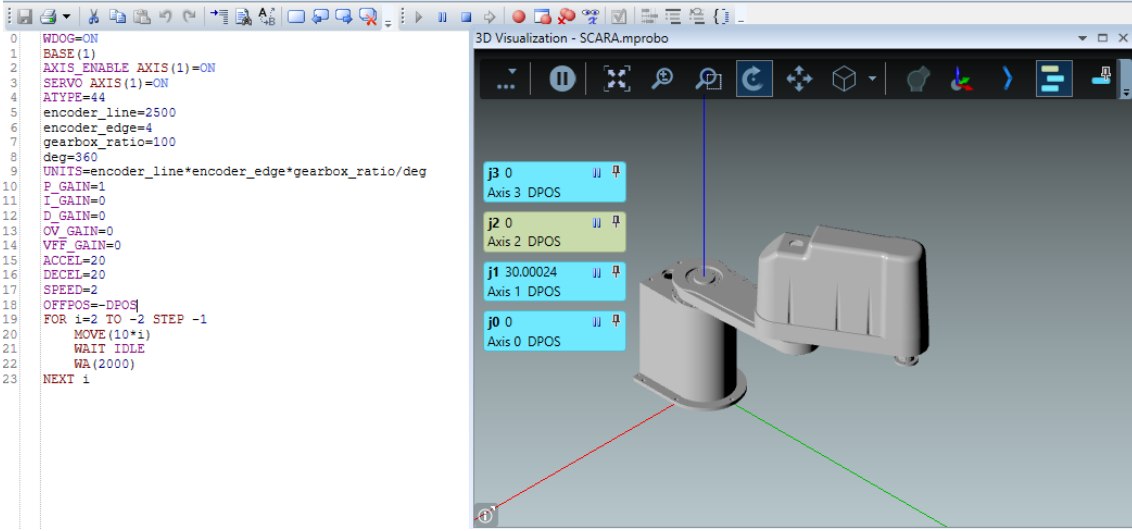
İkinci eklem için bu değerler girildikten sonra, ikinci ekleme yer alan motora ev konumundan sırasıyla +20, +30, +20 ve 0 derece dönecek şekilde komut gönderilir. (Şekil 5.24., Şekil 5.25. ve Şekil 5.26.)



Şekil 5.24. İkinci eklemin ev konumu



Şekil 5.25. İkinci eklemin saat yönünde 20 derece dönüşü



Şekil 5.26. İkinci eklemin saat yönünde 30 derece dönüşü

Üçüncü eklem lineer hareket gerçekleştirdiği için birim dönüşümü yapılırken derece yerine bir turda aldığı yola (mm) bölünür. 5 eksenli çapak alma robotu için bu değer 10 mm'dir. Robotun diğer birim dönüşüm değerleri aşağıdaki gibidir.

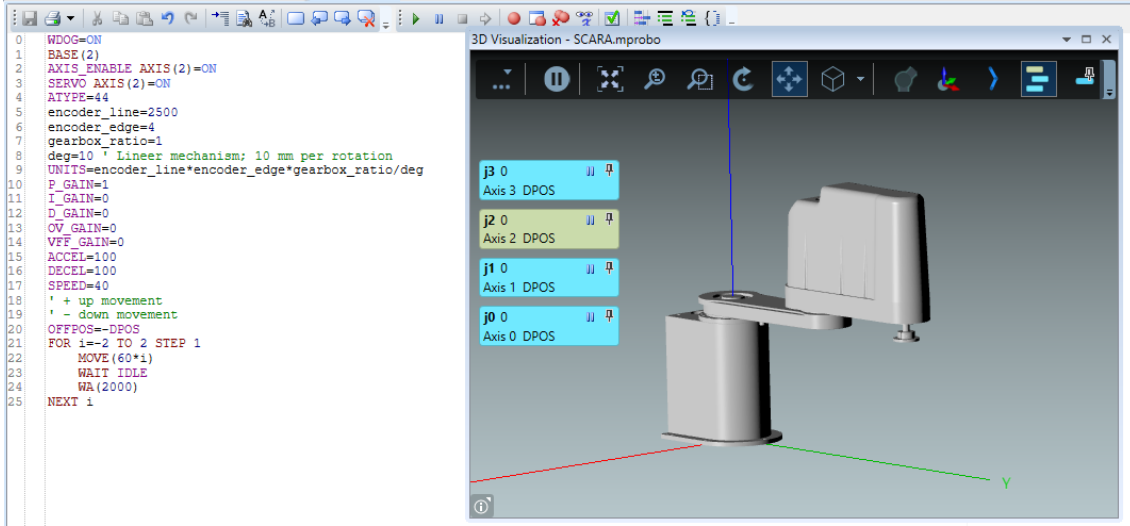
- Enkoder çözünürlüğü: 2500
- Enkoder kenarı: 4
- Motor adım sayısı: 1

Birim dönüşümü yapıldıktan sonra motor hareketini gerçekleştirirken hangi PID, hız ve ivme değerlerini kullanacağı yazılır. Bu değerler üçüncü eklem için aşağıdaki gibidir.

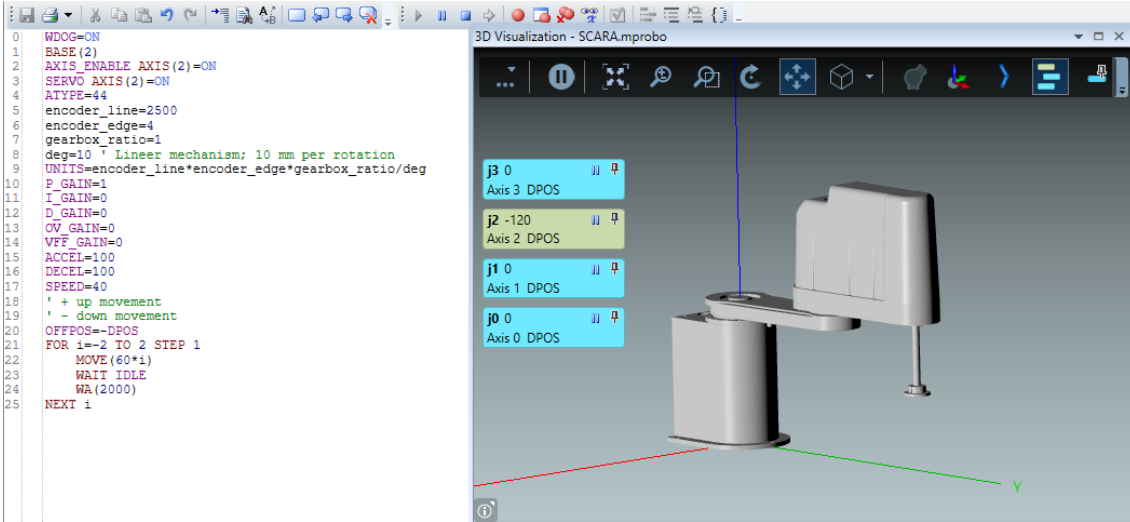
- P\_GAIN=1
- I\_GAIN=0
- D\_GAIN=0
- OV\_GAIN=0
- VFF\_GAIN=0
- ACCEL=100
- DECEL=100
- SPEED=40

Üçüncü eklem için bu değerler girildikten sonra, üçüncü ekleme yer alan motora ev konumundan sırasıyla -120, -150, -120 ve 0 mm yol alacak şekilde komut gönderilir.

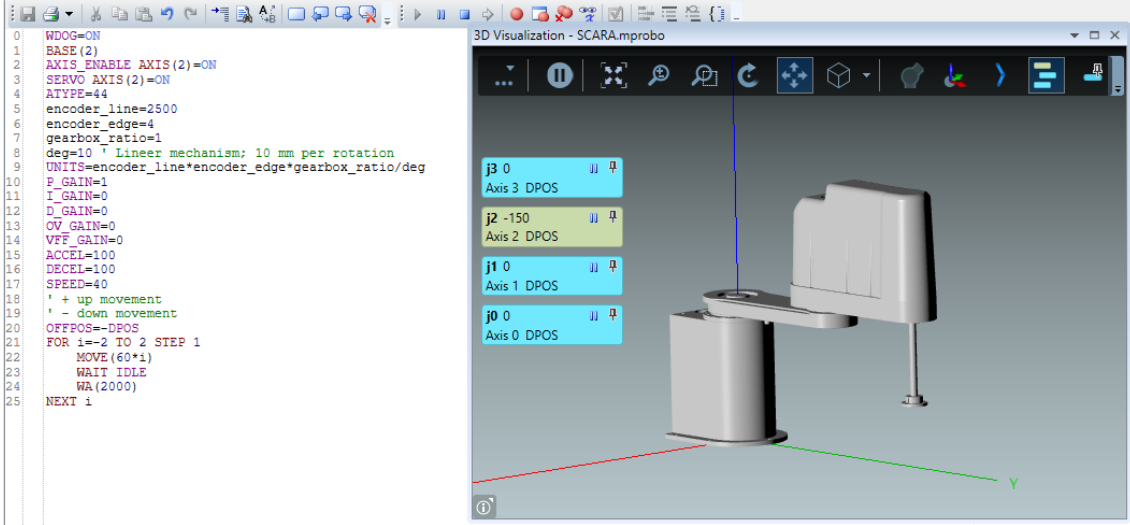
Buradaki – işareti Z ekseninde negatif yönde (aşağı) hareketi göstermektedir. (Şekil 5.27., Şekil 5.28. ve Şekil 5.29.)



Şekil 5.27. Üçüncü eklemin ev konumu



Şekil 5.28. Üçüncü eklemin -120 mm yol alışı



Şekil 5.29. Üçüncü eklem için -150 mm yol alışı

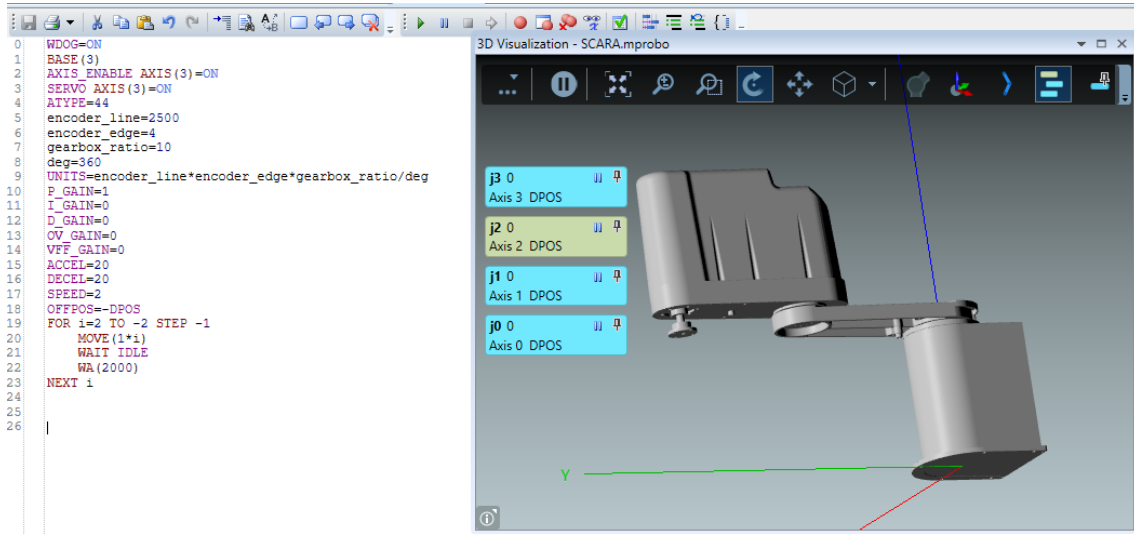
Dördüncü eklem için robotun birim dönüşüm değerleri aşağıdaki gibidir.

- Enkoder çözünürlüğü: 2500
- Enkoder kenarı: 4
- Motor adım sayısı: 10

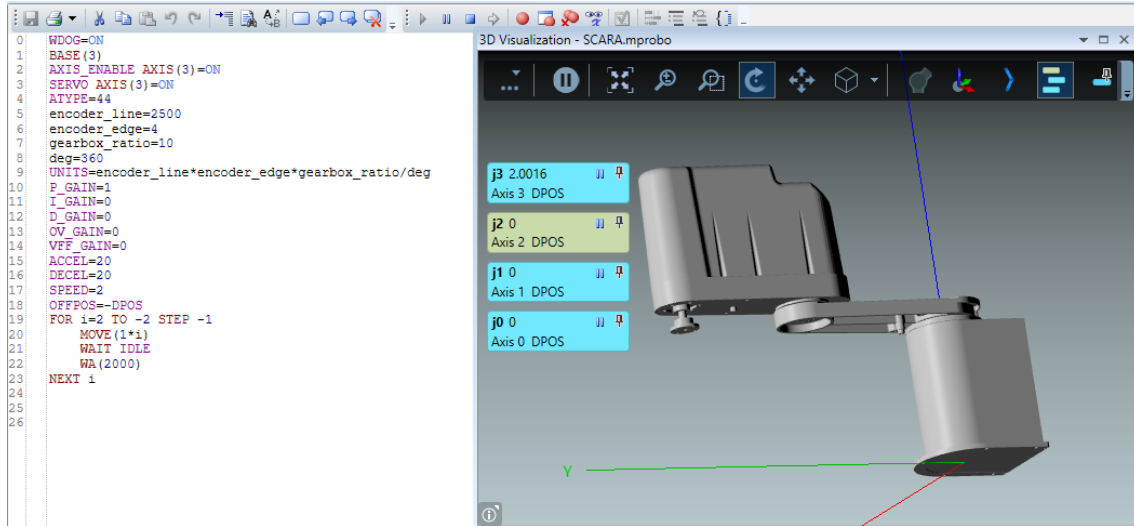
Birim dönüşümü yapıldıktan sonra motor hareketini gerçekleştirirken hangi PID, hız ve ivme değerlerini kullanacağı yazılır. Bu değerler dördüncü eklem için aşağıdaki gibidir.

- P\_GAIN=1
- I\_GAIN=0
- D\_GAIN=0
- OV\_GAIN=0
- VFF\_GAIN=0
- ACCEL=20
- DECEL=20
- SPEED=2

Dördüncü eklem için bu değerler girildikten sonra, dördüncü ekleme yer alan motora ev konumundan sırasıyla +2, +3, +2 ve 0 derece dönecek şekilde komut gönderilir. (Şekil 5.30., Şekil 5.31. ve Şekil 5.32.)

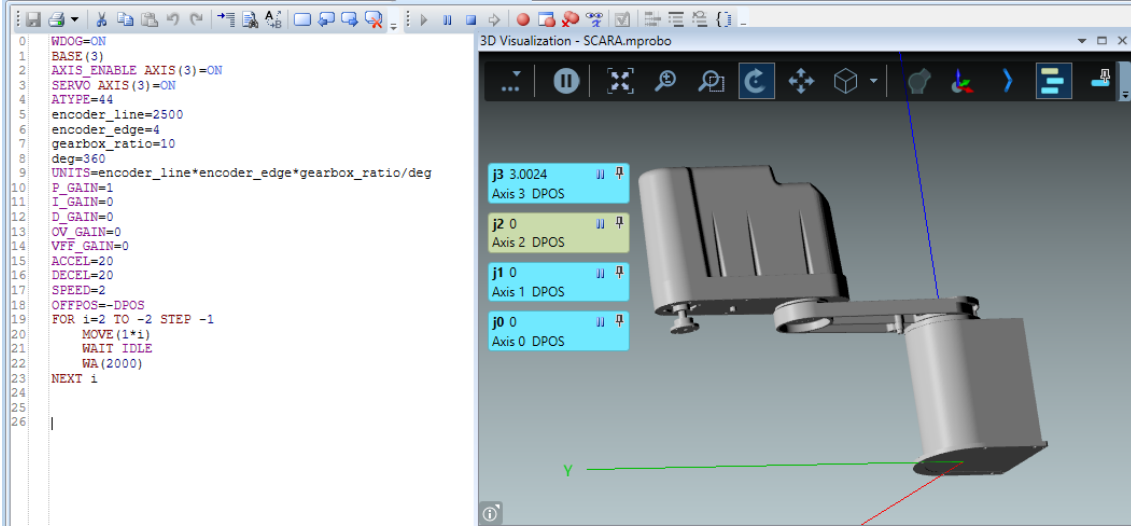


Şekil 5.30. Dördüncü eklemin ev konumu



Şekil 5.31. Dördüncü eklemin saat yönünde 2 derece dönüşü





Şekil 5.32. Dördüncü eklem için saat yönünde 3 derece dönüşü

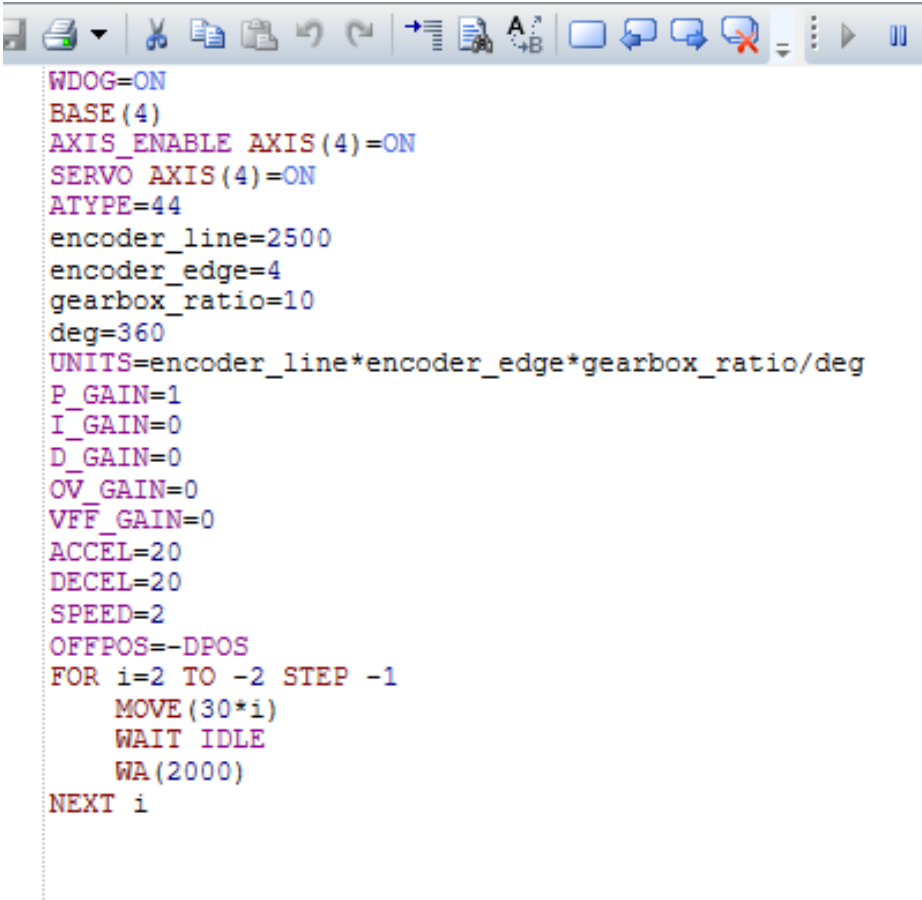
Beşinci eklem için robotun birim dönüşüm değerleri aşağıdaki gibidir.

- Enkoder çözünürlüğü: 2500
- Enkoder kenarı: 4
- Motor adım sayısı: 10

Birim dönüşümü yapıldıktan sonra motor hareketini gerçekleştirirken hangi PID, hız ve ivme değerlerini kullanacağı yazılır. Bu değerler beşinci eklem için aşağıdaki gibidir.

- P\_GAIN=1
- I\_GAIN=0
- D\_GAIN=0
- OV\_GAIN=0
- VFF\_GAIN=0
- ACCEL=20
- DECEL=20
- SPEED=2

Beşinci eklem için bu değerler girildikten sonra, beşinci ekleme yer alan motora ev konumundan sırasıyla +60, +90, +60 ve 0 derece dönecek şekilde komut gönderilir. (Şekil 5.33. ve Şekil 5.34.)



```
WDOG=ON
BASE (4)
AXIS_ENABLE AXIS (4)=ON
SERVO AXIS (4)=ON
ATYPE=44
encoder_line=2500
encoder_edge=4
gearbox_ratio=10
deg=360
UNITS=encoder_line*encoder_edge*gearbox_ratio/deg
P_GAIN=1
I_GAIN=0
D_GAIN=0
OV_GAIN=0
VFF_GAIN=0
ACCEL=20
DECEL=20
SPEED=2
OFFPOS=-DPOS
FOR i=2 TO -2 STEP -1
    MOVE (30*i)
    WAIT IDLE
    WA(2000)
NEXT i
```

Şekil 5.33. Beşinci eklemin kontrol programı

Parameter	Axis (0)	Axis (1)	Axis (2)	Axis (3)	Axis (4)
ATYPE	Inc Servo	Inc Servo	Inc Servo	Inc Servo	Inc Servo
UNITS	2777.77778	2777.77778	1000.0	277.77778	277.77778
<b>▲ Gains</b>					
P_GAIN	1.0	1.0	1.0	1.0	1.0
I_GAIN	0.0	0.0	0.0	0.0	0.0
D_GAIN	0.0	0.0	0.0	0.0	0.0
OV_GAIN	0.0	0.0	0.0	0.0	0.0
VFF_GAIN	0.0	0.0	0.0	0.0	0.0
<b>▲ Velocity profile</b>					
ACCEL	20.0	20.0	100.0	20.0	20.0
CREEP	0.03600	0.03600	0.10000	0.36000	0.36000
DECEL	20.0	20.0	100.0	20.0	20.0
MERGE	0	0	0	0	0
SPEED	2.0	2.0	40.0	2.0	2.0
SRAMP	0	0	0	0	0
VP_SPEED	0.0	0.0	0.0	0.0	2.0
MSPEED	0.0	0.0	0.0	0.0	0.0
<b>▲ Limits</b>					
DATUM_IN	-1	-1	-1	-1	-1
FE_LIMIT	7.20000	7.20000	20.0	72.0	72.0
FE_RANGE	3.60000	3.60000	10.0	36.0	36.0
FHOLD_IN	-1	-1	-1	-1	-1
FS_LIMIT	144000000.0	144000000.0	400000000.0	1440000000.0	1440000000.0
FWD_IN	-1	-1	-1	-1	-1
REP_DIST	72000000.0	72000000.0	200000000.0	720000000.0	720000000.0
REP_OPTION	0	0	0	0	0
REV_IN	-1	-1	-1	-1	-1
RS_LIMIT	-144000000.0	-144000000.0	-400000000.0	-1440000000.0	-1440000000.0
<b>▲ Positions</b>					
DPOS	0.0	0.0	0.0	0.0	46.62000
ENCODER	0	0	0	0	0
ENDMOVE	0.0	0.0	0.0	0.0	60.00120
MPOS	0.0	0.0	0.0	0.0	46.62000
REMAIN	0.0	0.0	0.0	0.0	13.38120
<b>▲ Axis output</b>					
DAC	0	0	0	0	0
SERVO	1	1	1	1	1
<b>▲ Status</b>					
AXISSTATUS	None	None	None	None	None
FE	0.0	0.0	0.0	0.0	0.0
LINK_AXIS	-1	-1	-1	-1	-1
MTYPE	Idle	Idle	Idle	Idle	MOVE
NTYPE	Idle	Idle	Idle	Idle	Idle
TABLE_POINTER	0.0	0.0	0.0	0.0	0.0
<b>▲ Registration</b>					
MARK	0	0	0	0	0
REG_POS	0.0	0.0	0.0	0.0	0.0
<b>▲ Jogging</b>					
FAST_JOG	-1	-1	-1	-1	-1
FWD_JOG	-1	-1	-1	-1	-1
REV_JOG	-1	-1	-1	-1	-1

Şekil 5.34. Beşinci eklemin hareketi sırasındaki eksen değişkenleri

Eklemlerin bir arada çalışması Şekil 5.35. ve Şekil 5.36.'de gösterilmiştir.

The screenshot displays the Motion Perfect v4.14 software interface. The main window shows a 3D visualization of a robotic arm with four axes. The interface includes a top menu bar, a toolbar, and several panels:

- Controller Panel:** Shows the controller name 'MC400' and various status indicators like 'Axis Status OK', 'System OK', and 'Motion stop'.
- Programs Panel:** Lists several programs including 'ELEM1', 'ELEM2', 'ELEM3', 'ELEM4', 'ELEM5', 'ELEM6', 'ELEM7', 'ELEM8', 'ELEM9', 'ELEM10', 'ELEM11', 'ELEM12', 'ELEM13', 'ELEM14', 'ELEM15', 'ELEM16', 'ELEM17', 'ELEM18', 'ELEM19', 'ELEM20', 'ELEM21', 'ELEM22', 'ELEM23', 'ELEM24', 'ELEM25', 'ELEM26', 'ELEM27', 'ELEM28', 'ELEM29', 'ELEM30', 'ELEM31', 'ELEM32', 'ELEM33', 'ELEM34', 'ELEM35', 'ELEM36', 'ELEM37', 'ELEM38', 'ELEM39', 'ELEM40', 'ELEM41', 'ELEM42', 'ELEM43', 'ELEM44', 'ELEM45', 'ELEM46', 'ELEM47', 'ELEM48', 'ELEM49', 'ELEM50', 'ELEM51', 'ELEM52', 'ELEM53', 'ELEM54', 'ELEM55', 'ELEM56', 'ELEM57', 'ELEM58', 'ELEM59', 'ELEM60', 'ELEM61', 'ELEM62', 'ELEM63', 'ELEM64', 'ELEM65', 'ELEM66', 'ELEM67', 'ELEM68', 'ELEM69', 'ELEM70', 'ELEM71', 'ELEM72', 'ELEM73', 'ELEM74', 'ELEM75', 'ELEM76', 'ELEM77', 'ELEM78', 'ELEM79', 'ELEM80', 'ELEM81', 'ELEM82', 'ELEM83', 'ELEM84', 'ELEM85', 'ELEM86', 'ELEM87', 'ELEM88', 'ELEM89', 'ELEM90', 'ELEM91', 'ELEM92', 'ELEM93', 'ELEM94', 'ELEM95', 'ELEM96', 'ELEM97', 'ELEM98', 'ELEM99', 'ELEM100'.
- Code Editor:** Displays the G-code program for the robot, including comments and axis movements.
- Axis Parameters Panel:** Provides detailed settings for each axis, including position, velocity, and acceleration limits.
- Output Panel:** Shows the execution log of the programs, including timestamps and status messages.

The G-code program in the code editor includes the following key sections:

```

0 DPOS=ON
1 BASE (1,1,2,3)
2 FWDJOG TO 3
3 FWDJOG TO 3
4 FWDJOG TO 3
5 SERVO AXIS(1)=ON
6 SERVO AXIS(1)=ON
7 NEXT 1
8
9 ACCEL=44
10 UNITS AXIS(0)=4500*4+110/360
11 UNITS AXIS(1)=4500*4+110/360
12 UNITS AXIS(2)=4500*4+110/360
13 UNITS AXIS(3)=4500*4+110/360
14 UNITS AXIS(4)=4500*4+110/360
15 FOR 1=0 TO 3
16   UNITS AXIS(1)=DPOS AXIS(1)
17 NEXT 1
18
19 FOR 1=0 TO 3
20   G-DRIN AXIS(1)=0
21   G-DRIN AXIS(2)=0
22   G-DRIN AXIS(3)=0
23   G-DRIN AXIS(4)=0
24   VFF GAIN AXIS(1)=0
25   VFF GAIN AXIS(2)=0
26   VFF GAIN AXIS(3)=0
27   VFF GAIN AXIS(4)=0
28   ACCEL AXIS(1)=20
29   ACCEL AXIS(2)=20
30   ACCEL AXIS(3)=20
31   ACCEL AXIS(4)=20
32   DECEL AXIS(1)=20
33   DECEL AXIS(2)=20
34   DECEL AXIS(3)=20
35   DECEL AXIS(4)=20
36 NEXT 1
37
38 MPOS 45, 45, -80, 10
39 WAIT 1000
40 RA (3000)

```

The output panel shows the following execution log:

```

%Process 2:Program [ON]S1 - Running
%Process 2:Program [ON]S1 - Program is stopped
%Process 2:Program [ON]S1 - Running
%Process 2:Line 27 (31) - Program is stopped
%Process 2:Program [ON]S1 - Running
%Process 2:Line 32 (31) - Program is stopped
%Process 2:Program [ON]S1 - Running
%Process 2:Program [ON]S1 - Program is stopped
%Process 2:Line 36 (31) - Program is stopped
%Process 2:Program [ON]S1 - Running
%Process 2:Line 32 (31) - Program is stopped
%Process 2:Program [ON]S1 - Running
%Process 2:Line 18 (31) - Program is stopped

```

Şekil 5.35. Eklemlerin ev konumu

The screenshot displays the Motion Perfect v4.1.4 software interface. At the top, a 3D visualization shows a robot arm with a gripper. Below it, a G-code program is visible, containing various axis movement and velocity commands. The bottom right section shows the 'Axis Parameters' table, which lists parameters for four axes (Axis 0 to Axis 4) and their corresponding units, gains, and limits.

**Axis Parameters Table:**

Parameter	Axis (0)	Axis (1)	Axis (2)	Axis (3)	Axis (4)
ATYPE	2777.77778	2777.77778	1000.0	2777.77778	277.77778
UNITS	2777.77778	2777.77778	1000.0	2777.77778	277.77778
<b>Gain</b>					
P_GAIN	1.0	1.0	1.0	1.0	1.0
I_GAIN	0.0	0.0	0.0	0.0	0.0
D_GAIN	0.0	0.0	0.0	0.0	0.0
OV_GAIN	0.0	0.0	0.0	0.0	0.0
VFF_GAIN	0.0	0.0	0.0	0.0	0.0
<b>Velocity profile</b>					
ACCEL	200	200	200	200	200
DECEL	0.05900	0.05900	0.10000	0.39000	0.36000
TRAP	200	200	200	200	200
MERGE	0	0	0	0	0
SPEED	2.0	2.0	2.0	2.0	2.0
SRAMP	0	0	0	0	0
V6_SPEED	0.0	0.0	0.0	0.0	0.0
V7_SPEED	0.0	0.0	0.0	0.0	0.0
V8_SPEED	0.0	0.0	0.0	0.0	0.0
<b>Limits</b>					
DATUM_IN	-1	-1	-1	-1	-1
FE_LIMIT	720000	720000	200	720	720
FE_RANGE	360000	360000	100	360	360
FF_ODD_IN	-1	-1	-1	-1	-1
FF5_LIMIT	14400000.0	14400000.0	40000000.0	144000000.0	144000000.0
FWD_IN	-1	-1	-1	-1	-1
REP_DIST	7200000.0	7200000.0	20000000.0	72000000.0	72000000.0
REP_OPTION	-1	-1	-1	-1	-1
REV_IN	-1	-1	-1	-1	-1
RS_LIMIT	-14400000.0	-14400000.0	-40000000.0	-144000000.0	-144000000.0
<b>Positions</b>					
DPOS	-50.00004	29.99988	-80.0	10.00080	0.0
ENCODER	0	0	0	0	0
ENDMOVE	-50.00004	29.99988	-80.0	10.00080	0.0
MPOS	-50.00004	29.99988	-80.0	10.00080	0.0
REMAIN	0.0	0.0	0.0	0.0	0.0
<b>Axis output</b>					
DAC	0	0	0	0	0
SERVO	1	1	1	1	1
<b>Status</b>					
AMISSTATUS	None	None	None	None	None
FE	0.0	0.0	0.0	0.0	0.0
LINK_AXIS	-1	-1	-1	-1	-1
MTYPE	Idle	Idle	Idle	Idle	Idle
NTYPE	Idle	Idle	Idle	Idle	Idle
TABLE_JOINTER	0.0	0.0	0.0	0.0	0.0
<b>Registration</b>					
MARK	0	0	0	0	0
REG_POS	0.0	0.0	0.0	0.0	0.0
<b>Jogging</b>					
FAST_JOG	-1	-1	-1	-1	-1
FWD_JOG	-1	-1	-1	-1	-1
REV_JOG	-1	-1	-1	-1	-1

Şekil 5.36. Eklemlerin bir arada hareketi ve eksen parametreleri

## 6. SONUÇ VE ÖNERİLER

Bu çalışmada, set üstü ocaklar için üretilen döküm ızgaraların üzerindeki çapakları almak üzere tasarlanan 5 eksenli SCARA tipi robot kolunun uç elemanının, döküm parçasının CAD çizimlerinden aldığı verileri kullanarak parçanın etrafında dolaşacak şekilde otomatik yörüngesi yaratılmıştır. Robot kolunun uç elemanının izleyeceği konum bilgisi ve köşe bilgisi (X,Y,Z) dünya koordinat sistemine göre hesaplanmış ve geliştirilen grafiksel kullanıcı ara yüzü ile robot kolunun uç elemanının izleyeceği yörünge görselleştirilmiştir. Konum bilgisi ve köşe bilgisi TXT uzantılı iki ayrı dosya şeklinde kaydedilmiştir. Bu koordinatların robot kolunun üzerinde uygulanabilir olabilmesi için robot pano bağlantılarının, robot üzerinde uygulanan hız kontrol şekline göre yapılmıştır. Son olarak, servo sürücü parametreleri kontrol şekline uygun bir şekilde ayarlanarak ve her bir robot eklemi için hız, ivme, pozisyon, PID değişkenleri belirlenmiştir. Böylelikle, 5 eksenli SCARA tipi çapak alma robotunun kontrolü sağlanmıştır.

Çalışma sırasında yörünge yaratılırken ilk önce, DWG formatındaki Autocad çizimleri DXF ve STL dosya formatlarında kaydedilir. DXF dosyası Matlab programına aktarılırken varlık bölümünde sınıflandırılan grafiksel çeşitliliklere göre ayrıştırılır. Bu uzantıdaki çizim komutlarını sembolize eden grup kodlar saptanıp çizilen verinin şekline göre gerekli olan özellikleri ayıklanıp kaydedilir. Çizim programlarından alınan üç boyutlu verilerin bilgileri Matlab ortamına aktarılıp ayrıştırılması yapıldıktan sonra komutları sembolize eden geometrik sınırların belirlenmesi sağlanır. Matlab programında fonksiyonlar oluşturularak üç boyutlu şekillerin çizimleri gerçekleştirilir. STL dosya formatı ise bu çalışmada, DXF dosya formatından elde edilen çizim ve yörüngeyi doğruluğunu kontrol etmek için kullanılır. Matlab ortamında yeniden çizimi ve doğruluğu test edilmiş olan parçaların, kullanıcı tarafından girilen robot kolunun uç elemanının çap bilgisi ile beraber yeniden çizim işlemi yapılarak robotun izleyeceği yörünge çıkartma işlemi tamamlanmış olur. Kullanıcı tarafından girilen robot kolunun uç elemanının çap bilgisi cisim şekillerinin ekstrüzyon yön vektörü ile çarpılarak cismin uzaydaki konum bilgisine eklenir. Böylelikle robot kolunun uç elemanının, cad çiziminde yer alan parçanın etrafından dolaşarak parçanın yüzeyinde var olan çapağı temizleyecek yörüngesi yaratılır. Yaratılan bu yörüngeyi dünya

koordinat sisteminde (Kartezyen koordinat sistemi) X, Y, Z noktaları TXT uzantılı bir dosyaya kaydedilerek robot kolunun uç elemanının takip edeceği konum bilgisi olarak tutulur. Kullanıcının yükünü hafifletmek için bu işlemleri kullanıcıdan aldığı girdilerle sırasıyla gerçekleştirecek bir ara yüz tasarlanmıştır. Arayüz aracılığıyla elde edilen bu koordinatların robot kolunun üzerinde uygulanabilir olabilmesi için robot pano bağlantılarının içerik tanımlaması yapılmıştır. Tanımlanan bu bağlantılara göre servo sürücü ve hareket kontrol kartı arasındaki donanımsal bağlantı ilişkileri belirlenmiş ve bu bağlantılar robot kolu üzerinde gerçekleştirilmiştir. Kontrol yöntemine uygun sürücü parametreleri incelenmiştir. Son olarak, servo sürücü parametreleri kontrol şekline uygun bir şekilde ayarlanarak her bir robot eklemi için hız, ivme, pozisyon, PID değişkenleri belirlenmiştir. *MotionPerfect* uygulaması üzerinde her eklem için programlar yazılarak robot kolu üzerinde denenmiştir. Böylelikle, 5 eksenli SCARA tipi çapak alma robotunun kontrolü sağlanmıştır.

Gelecekte bu çalışmadan faydalanarak, kontrolü sağlanan çapak alma robotuna yönelik bütün işlem basamaklarını gerçekleyecek bir arayüz tasarımı yapılabilir. Cad çizimlerinden çekilen parçaya ait koordinatlar robot kolu üzerinde kinematik ve ters kinematik işlem süreçlerinden geçtikten sonra robota ait eklemlerin belirlenen çalışma uzayında çalıştırılması sağlanabilir. Bu bağlamda sanayide, döküm parçaları üretiminde oluşan çapakların insan kullanımı olmadan temizleyecek bir robot kolu yerel olarak üretilerek hem maliyet hem de parça üretim işlemi azaltılabilir.

## KAYNAKLAR DİZİNİ

- Adtech Technology Co., 2012, Single-channel all-digital AC servo drives QS7 Series User Manual, <http://www.robosan.com.tr/pdfs/QS7%20MANUAL.pdf>, erişim tarihi: 11.8.2018
- Anonim1, 2016, AutoCAD Nedir? Kullanım Alanları ve Avantajları, <https://www.techworm.com/autocad-nedir-kullanim-alanlari-avantajlari/>, erişim tarihi: 11.5.2018.
- Anonim2, 2017, STL File Format for 3D Printing – Simply Explained, <https://all3dp.com/what-is-stl-file-format-extension-3d-printing/#pointone>, erişim tarihi: 11.5.2018.
- Anonim3, 2010, AutoCAD 2010 User Documentation, <http://docs.autodesk.com/ACD/2010/ENU/AutoCAD%202010%20User%20Documentation/index.html?url=WS1a9193826455f5ffa23ce210c4a30acaf-71e1.htm,topicNumber=d0e50148>, erişim tarihi: 12.5.2018.
- 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.
- Autodesk Inc., 2012, DXF Reference, [http://images.autodesk.com/adsk/files/autocad\\_2012\\_pdf\\_dxf-reference\\_enu.pdf](http://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf), erişim tarihi: 12.5.2018.
- F.M. Proctor and K.N. Murphy, “Advanced Deburring System Technology,” ASME Winter Annual Meeting, PED 38, San Francisco, CA, (December 10-15, 1989).
- 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.
- Mohammad, M., 2008, Modeling a Deburring Process, Using DELMIA V5®, School of Engineering, University of Bridgeport, Bridgeport CT 06604, USA, *J. Eng. Applied Sci.*, 3 (11): 835-847 p.
- Murphy, K. M., Norcross, R. J. and Proctor, F. M., 1998, Cad Directed Robotic Deburring, 1-8 p.
- Rososhansky, M., Xi, F., 2011, Coverage based tool-path planning for automated polishing using contact mechanics theory, *Journal of Manufacturing Systems* 30, 144–153.



**KAYNAKLAR DİZİNİ (devam)**

Stouffer, K.A. and R. Robert Jr., 1995. ADACS: An Advanced Deburring and Chamfering System. National Conference Publication-Institution of Engineers Australia NCP, pp: 623-628

Trio Motion Technology Ltd, 2017, Hardware Overview, <http://www.triomotion.uk/system/downloadPage.php?location=%2Fdocuments%2Fmanuals%2Fmanual7.7%2Fhardware>, erişim tarihi: 11.8.2018

Trio Motion Technology Ltd, 2016, Quick Start Guide, <http://www.triomotion.uk/public/products/p849.php>, erişim tarihi: 11.8.2018