

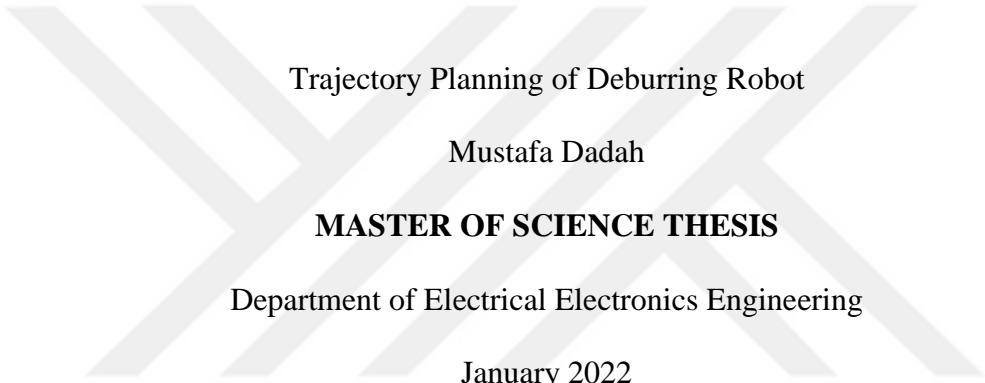
Çapak Alma Robotu için Yörünge Planlaması

Mustafa Dadah

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliği Anabilim Dalı

Ocak 2022



Trajectory Planning of Deburring Robot

Mustafa Dadah

MASTER OF SCIENCE THESIS

Department of Electrical Electronics Engineering

January 2022

Çapak Alma Robotu için Yörünge Planlaması

Mustafa Dadah

Eskişehir Osmangazi Üniversitesi

Fen Bilimleri Enstitüsü

Lisansüstü Yönetmeliği Uyarınca

Elektrik Elektronik Mühendisliği Anabilim Dalı

Kontrol ve Kumanda Sistemleri Bilim Dalında

YÜKSEK LİSANS TEZİ

Olarak Hazırlanmıştır

Danışman: Prof. Dr. Osman Parlaktuna

Ocak 2022

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Prof. Dr. Osman Parlaktuna danışmanlığında hazırlamış olduğum “**Çapak Alma Robotu için Yörünge Planlaması**” başlıklı 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.17/01/2022

Mustafa Dadah

ÖZET

Bu çalışmanın amacı çapak alma işlemi yapan robotların hareket planını hazırlamasını kolaylaştırmak için otonom bir mekanizma tasarlamaktır. Bu amaç doğrultusunda robotun yörünge planlaması için sanayi ortamlarında genelde bulunan iş parçasının 2 boyutlu Autocad çiziminden faydalanılmaktadır. İş parçasının Autocad çiziminin DXF formatında bulunan geometrik bilgilerinden yararlanarak, Matlab ortamında okunan veriler geri çizildikten sonra, belli kriterlere göre robotun atacağı adımlar hesaplanmıştır. Çapak alma robotlarının çalışma kriterlerine göre, robotun iş parçasının tamamının etrafını temizleyebilmesi için takım ve iş parçasının ayrılıp yeniden geri kavuşmalarının sayısının asgariye indirilmesi gerekmektedir. Bu amaçla izlenecek yol Matlab programı kullanarak kesintisiz döngülere ayrılmış ve ters kinematik kullanarak eklemlerin açıları gezilecek yola göre belirlenmiştir.

Anahtar Kelimeler: Çapak alma, robot, yörünge, SCARA, Matlab, iş parçası

SUMMARY

The aim of this thesis is to find a solution to the difficulty of planning the trajectory of deburring industrial robots by automating the extraction of deburring trajectory from 2D AutoCAD drawings of the work-piece. In general, 2-dimensional work-piece drawings are widely available in the industrial environment. The work-piece drawing extraction is accomplished using Matlab by reading the DXF format, which includes the geometrical information of the AutoCAD drawing. After reading the AutoCAD drawing, it is redrawn by Matlab and the robot tag point is added. To deburr the work-piece, the robot prefers to follow a deburring trajectory with limited loss of connection with the edges of the work-piece. As a result, we write a program that divides the work-piece edges into loops to follow the loop tag points by robot. The loop tag points were extracted from the work-piece edge geometrical element tag-cells. Finally, the robot feeds the tag point robot angles for trajectory following, which is calculated by inverse kinematics, and by this the mission of this study is achieved.

Keywords: Deburring, robot, trajectory, SCARA, Matlab, work-piece

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ VE AMAÇ	1
2. LİTERATÜR ARAŞTIRMASI	5
2.1 Çapak Alma Problemi için Yapılan Robotik Çalışmalar	5
2.2 Çapak Alma İşlemi	12
2.3 Çapak Alma Yöntemleri	13
3. MATERYAL VE YÖNTEM	18
3.1 Giriş ve İleri Kinematik	18
3.2 Ters Kinematiğin Hesaplaması	25
3.3 Sağ ve Sol İki Eklemlili Robotun Çözümü	29
3.4 İki Eksenli Robot Kolunun Çalışma Uzayı	31
3.5 CAD Çizimleri	37
4. BULGULAR VE TARTIŞMA	42
5. SONUÇ ÖNERİLER	44
KAYNAKLAR DİZİNİ	45

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
6. EK AÇIKLAMALAR	47
Ek Açıklama-A: Daire ve Yay Çizme Programı	47
Ek Açıklama-B: Elips Çizme Programı	49
Ek Açıklama-C: Çizginin Matlab Ortamında Çizme Programı	50
Ek Açıklama-D: Matlab Ortamına Çizim Aktarma Programı	52
Ek Açıklama-E: Yayın Başlangıç ve Bitişleri Bulma Program	58
Ek Açıklama-F: Döngülerin birbirinden Ayrılması için Yazılan Program	60
Ek Açıklama-G: Etiket Noktalarını bir Hücre İçinde Toplama Programı	62
Ek Açıklama-H: Planlanan Yolun Simülasyonu için Yazılan Program	63
Ek Açıklama-I: Planlanan Yolda Robotun Açılarını Bulma Programı	64
Ek Açıklama-J: Linespace Limiti içinde Açıların Çizim Programı	65

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
3.1. Özel Tasarım SCARA Robot Kolu	18
3.2. Koordinat Sistemlerinin Robotun Üzerine Yerleştirilmesi	19
3.3. 5 Eksenli Robotun Kinematik Diyagramı	20
3.4. İleri Kinematik Programı	24
3.5. Simülasyon Ortamında Robot Gövdesi	24
3.6. Ters Kinematiğin Programı	29
3.7. İki Eklemlili Düzlemsel Robotun Sağ ve Sol Çözümü	30
3.8. Kullanılan 3 Eklemlili SCARA Robot	32
3.9. Robot Çalışma Alanının Matlab Çizim Programı	36
3.10. Üç Eklemlili Robotun Çalışma Alanı	36
3.11. Farklı Dairelerin Çizimi	38
3.12. Farklı Yön Sahibi Yayların Matlab Ortamında Çizimleri	38
3.13. Elipsin AutoCAD Çiziminin Matlab Ortamında Çizilmesi	39
3.14. AutoCAD'ta Çizilmiş Doğruların Matlab Ortamında Çizilmesi	40
3.15. 4 Eleman İçeren Resmin AutoCAD Ortamında Çizimi	41
3.16. 4 Eleman İçeren Resmin Matlab Ortamında geri Çizilmesi	41
4.1. İş Parçasının Matlab Ortamında geri Çizilmesi	43
4.2. Parça Üstündeki Etiket Noktaları	43
5.1. Linspace boyunca Açların Değişmesi	44

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
2.1. Çapak Alma İşlemine Etkileyen Parçanın Karakterleri	13
3.1. 5 Eksenli Robotun Denavit-Hartenberg Parametreleri	20
3.2. Peter Corke Kütüphanesine göre Denavit-Hartenberg Parametreleri	20



1. GİRİŞ VE AMAÇ

Madenler; ilk çağlardan beri birçok alanda farklı amaçlar için kullanılmaktadır. Fakat, madenler ham madde olarak kullanılmaz, onun elverişli bir biçime getirilebilmesi için maden işleme yöntemlerine başvurulmaktadır.

Bu araştırma kapsamında ocak üstü ızgaralarının üzerinde olan çapakların robotlar tarafından temizlenebilmesi işlemi için robotun çalışma planlaması yapılacaktır. Genelde, ızgaraların metal yapısının şekillendirmesi için döküm işlemi kullanılmaktadır. Döküm işlemi sıvı metalin çeşitli tip kalıplara doldurularak soğuyup katılaşması şeklinde yapılmaktadır. İşlem sırasında en iyi yüzey kalitesi ve en düşük boyut toleransı (en iyi boyut tutarlılığı) gerçekleştirilmesi amaçlanmaktadır.

Malzemeye son şekil verilmesi amacıyla yapılan döküm işleminin neticesinde malzemenin üzerinde çapaklar oluşabilmektedir. Malzemenin yüzey düzgünlüğünü sağlamak için çapaklarını temizlemek gerekmektedir. Çapakların temizlenmesi ya da belli bir ölçüye getirilmesi amacıyla yapılan eğeleme işlemi, talaşlı imalatta tesviyecilik konusuna girmektedir (Yasin, 2009).

Talaşlı imalat, keskin takımlar kullanarak ürün üretilmesi için iş parçasından talaş kaldırarak başlı başına bir maden işleme yöntemi olarak kullanılmaktadır. Dövme işlemi, döküm işlemi ve haddeleme işlemleri ile üretilen metallerin dörtte birinden çoğu nihai şekil ve ebada talaşlı üretim ile ulaşır (Ekrem, 2021). Talaşlı imalat sırasında kesici takım malzemeye karşı sabit bir kuvvet uygulamaktadır, fakat; bu prosesi etkileyen malzemeye bağlı madenin işlenebilirliği makineye bağlı kesme hızı (V) ve ilerleme hızı (f) gibi birden fazla faktör vardır.

(Ekrem, 2021)'e göre işlenebilirlik ne kadar iyi ise imalat o kadar kısa sürede gerçekleşir ve maliyetler da o oranda azalır, ancak işlenebilirlik ne kadar zor ise imalat süreleri ve maliyetler o denli artmaktadır. Makineye bağlı parametrelerin doğru seçilmesi halinde yüzey pürüzlülüğü asgariye indirilebilmektedir. Bu konuda önemli çalışmalar yapmış olan Thomas vd.ne (2000) göre maden işleme makinelerinde yatırım seneden

seneye artmaktadır ve ülkelerin gelişimi ve değeri bu alanda yaptıkları yatırım bakımından değerlendirilebilmektedir. Thomas vd.ne (2000) göre de iş parçası işleme toleransı (50 μm) ve bitirme toleransı (1 μm) olduğundan, makine ile gerçekleştirmesi kolay olduğu için metal işleme makine kullanarak yüksek bir hassasiyetle yapılabilir.

1970'li yıllardan itibaren CNC makineleri ortaya çıkmaya başlamıştır, 1950'li yıllarda kullanılan NC makinelere dayanarak geliştirilmiştir, üretilecek olan karmaşık ve detaylı tasarımlar sayısal kontrol verisi ve bilgisayar programı tarafından yaptırılmaktadır. CNC makinelerin hem performansı hem de sanayi ortamıyla entegre olarak çalışabilme imkânı günden güne gelişmiş, işleme takımı otomatik olarak değiştirebilmesi ve iş parçası robot kolu yardımıyla takılması mümkün olmuştur. Onstein vd.ne (2020) göre CNC makineler çapak alma işleminde gösterdiği sertlik ve tutarlıktan dolayı kullanılabilir, fakat yatırım maliyetleri yüksek olduğundan sadece kısıtlı sayıda karmaşık işlemlere uygun olmaktadır.

Robotlar doğru programlandığı durumda otomatik olarak çapak alma işlemini gerçekleştirme kabiliyetine sahiptir. Robot kolu işlem bakımından daha esnektir ve CNC makinelerin erişemediği malzemenin kenarlarına erişebilir. Robotların CNC tezgahlar gibi iş parçalarının bağlanması için kısıtlı boyut ve ağırlık taşıyabilen bir iş tablası bulunmadığından daha büyük hacimli bir iş parçası üzerine işlem yapabilmektedir. Bu nedenlerle çapak alma işleminde robot kullanılması daha uygun ve ucuz bir alternatiftir.

Sanayideki gelişmelerle bağlı olarak sanayi ortamlarında birçok sebepten dolayı robotlaşmaya duyulan ihtiyaç artmıştır. İnsanın hata payının yüksekliği, yorgunluk ve halsizliğe uğraması sonucu, kazalara ve felaketlere maruz kalabileceği gibi nedenler robotların insanların yerini almasına zemin hazırlamıştır. Bununla birlikte; robotların 7/24 çalışabilmesi ve işlemde hata toleransı düşük olduğu için endüstriyel ortamlarda çalışma oranı yükselmiştir, robotların fabrikalarda yerleri arttıkça 4. Sanayi devrimine bağlı olan akıllı fabrikaların ve entegre üretimin önemli bir unsur yerine getirilmiş olacaktır.

Robot kolu insan vücudunda el, kas ve iskelet sistemleri ile, görevinin taklidini yapması için yapılan bir alettir. Robot metalden yapılan bir gövdeye sahiptir ve bağlantıların arasında eklemler konulmaktadır. Eklemler iki tipe ayrılır: Doğrusal ve açısız. Genelde robotun hareketi eklemlerine direkt ya da dolaylı (dişliler kullanarak) bağlı olan mekanik

güç üreten (elektrik motorlar) ya da aktaran (pnömatik veya hidrolik pistonlar) aracıyla yaptırılmaktadır.

Robot kolunun matematiksel modelinin (ileri kinematiği) çıkarılması için bağlantı ilişkileri ve eklemlerin özelliklerini tablo olarak belirleyen Denavit-Hartenberg yöntemi kullanılırken ters kinematik için geometrik denklemlere başvurulmaktadır. Yakın zamanlarda yarı iletken teknolojisinin ve bilgisayarların gelişmesi ile robotların kontrol ve sürülmesinde önemli bir adım atılmıştır. Piyasaya sürülmüş olan denetim kartlarının yüksek hız, fazla giriş ve çıkış sayısı ve büyük hafızası sonucu birçok sürecin robotlaşması mümkün kılınmıştır.

Jayaweera ve Phil'e (2010) göre robotun programlaması için eski eğitim tabanlı yaklaşımlar kullanılabilir. Robotun ucuna eğitim ucu takılır, gidilecek noktalar teker teker gezdirilip kaydedilmesi ardından, o noktalara tekrar gidilmesiyle çapak alma işlemi yapılır.

Bu alanda bugünlerde CAD bilgileri kullanarak robotun hareket planlamasının çıkarılması önemli bir araştırma alanıdır ve bu çalışmanın da hedeflerinden birisidir. Bu çalışma kapsamında, SCARA robotun çapak alma işlemi yapması amacıyla yol planlanmaktadır ve robotun denetleyicisinde bulunan program vasıtasıyla robotun hareketi için gidilmesi gereken noktaların eklem açıları olarak sıralı bir şekilde beslenmesi sağlanmaktadır. Bu amaçla Matlab, AutoCAD, TrioMotion programları kullanılmaktadır.

AutoCAD programı: Bilgisayar yardımıyla tasarım yapılması için Autodesk tarafından hazırlanan bir programdır. İş parçasının geometrik çizimi AutoCAD programıyla yapılmaktadır, ardından DXF (Drawing Interchange Format) biçimiyle çeşitli kullanım amaçları için başka programlarda kullanılabilir. İş parçası temel öğeler (çizgi, yay, elips, daire vb.) kullanılıp çizildikten sonra kaydedilmektedir.

Matlab programı: Çoklu-paradigma programlama dili ve sayısal hesaplama ortamıdır, çalışmamızda birçok yazılım görevlerini üstlenmektedir. Bu görevler: DXF dosyası okuma, iş parçasını yeniden çizme, robotun gitmesi gereken noktaları hesaplama,

yörüngeyi döngülere ayırma, döngüleri sıralama ve robota uygulanması amacıyla sıralanmış pozisyonları robotun eklem açıları olarak hesaplamaktır.

Motion Perfect: TrioMotion denetleyicisi için 4 sürümünde ve yukarıda kullanılan bir programdır. Çalışmamızda robotun parametreleri ve hareket komutları kullanılarak robotun kontrolü yapılacaktır. Gidilecek noktaların pozisyonları eklemlerin açıları olarak beslenecektir.

SCARA robot: Laboratuvarımızda 3 tanesi dönel bir tanesi doğrusal olmak üzere 4 eksenli bir SCARA robot (Selective Compliance Articulated Robot Arm) bulunmaktadır. Basit bir yapıya sahip ve çeşitli amaçlar için kullanılan bir robottur. SCARA robot MC505 denetleyici ve servo motor vasıtasıyla sürülmektedir.

2. LİTERATÜR ARAŞTIRMASI

2.1 Çapak Alma Problemi için Yapılan Robotik Çalışmalar

Onstein vd.ne (2020) göre robot kullanılarak yapılan çapak alma işlemi 50 yıla yakın araştırma konusu olmuştur ve bu konuyu farklı yönlerden inceleyen çeşitli çalışmalar bulunmaktadır. Yapılan çalışmada birçok çalışmaya değinilmiş ve literatür araştırma bölümünde onlardan bir kısmı özetlenmiştir.

Murphy vd. (1998) yapmış oldukları çalışmada, iki adet robot, birisi parçanın bağlanması ve öbürü takımın taşınması için kullanılmaktadır. İşlenmesi istenilen parçaların işlenecek kenar, işleme yöntemi ve bağlanma bilgilerinin dokunmatik bir ekran kullanılarak belirlenmesi gerekmektedir. Ardından CAD geometrik bilgisi kullanarak güzergâh oluşturulmaktadır. Robot iki tur atmaktadır, birinci tur; geri besleme kullanılarak, robot kinematik hataları, takımın aşınması ve parçanın bağlama yerine tam oturmamasından kaynaklanan konum hatalarını telafi etmek için yapılmaktadır (temas gücünü kontrol ederek), ikinci tur esnasında ise çapak alma işlemi gerçekleşmektedir. Bu çalışmada parçanın iki farklı yüzey tipinden (düz ve silindirik) oluştuğu kabul edilmiş ve kenarlar iki yüzeyin kesişmesi olarak tanımlanmıştır. Çalışmaya göre bu iki yüzey bulunabilecek yüzeylerin %90'ını kapsamaktadır. Diğer yüzeyler bu iki yüzeye yaklaştırılabilmektedir. Yol planlaması şu kurallara dayanır; a) parçanın her bir tutuluşu için bir yol planı hazırlanmaktadır, b) aynı takım aletlerinin kullanılması istenen kenarlar robotun tek bir gezmesinde içerilmeyi çalışılıyor. Gezmeler döngülere bölünmekte ve her döngü takımın parçadan ayrılmaması şartıyla yapılmaktadır. Gidilecek noktalar ikiye ayrılır; işlenecek olan noktalar ve güzergâha yaklaşmak için kullanılan noktalar.

Mohammad vd. (2008) çalışmalarında çapak alma iki simülasyon üzerinde yapılmıştır ve çapak alma işleminde bazı deyimler açıklanmıştır:

Takım elde (Tool in hand): Bazı çapak alma sistemlerinde takımın yeri robotun ucudur, eğer parçayı robot taşımazsa ya da aşındırıcı takım bağlanmazsa (taşlama kaması) takım elde yönteminin kullanması zorunlu hale gelir.

Parça elde (Part in hand): Bazı parçalar küçük oldukları için robotla taşıyıp işlenmesi tercih edilir. Bu yöntemin kullanmasındaki en önemli özelliklerinden birisi, parçanın paketlenmesini yapabilmek için başka bir işçi ya da robota ihtiyaç duyulmayacaktır, takım sabit bir yere takıldığı için daha uzun taşlama kamaları ya da daha büyük taşlama diski takılabilir, dolayısıyla takımın çalışma süresi uzamaktadır.

Geliştirilmiş çapak alma takımı: Çapak alma işlemi planlanmış güzergâh üzerinde olduğu kabul edilmektedir. Kullanılan takım, parçanın tam bir şekilde temizlenebilmesi için hava piston grubu yardımıyla temizlenecek parçanın kenarlarına sabit bir temas gücüyle erişim sağlamaktadır (radyal bir şekilde 360 derecede) ve bir sensor vasıtasıyla parçanın üzerine uygulanan temas gücünü ölçümleyebilmektedir. Bu alet kullanıldığı durumda eğik ve düzensiz güzergahlar için çok detaylı işleme noktalarının hesaplamasına gerek kalmaz.

Stouffer ve Russell (1995) bu konuda yapmış oldukları çalışmada iki farklı grafik giriş ekranı; birisi mühendis rolü için ve öbürü ise eleman rolü için kullanılmaktadır. Mühendis ekranı vasıtasıyla kenarlara uygulanacak işleme derinliği belirtilmektedir. Eleman ekranı ise temizlenecek kenarın belirlenmesi ve baz parametrelerin değiştirilmesini sağlar. Hazırlanan iki yazılım paketi (PreManufacture ve PreEngineer), kenarların üzerinde etiket noktalarını belirler (etiketlemek), ardından yol planı belirlenip simülasyonu yapılır ve çarpışma olmayacağından emin olunur. Simülasyondan sonuçlanan çıktı, yol üretme aşamasına gönderilir ve onun vasıtasıyla gidilecek noktalar belirlenmektedir, ardından çıkan sonuçlar robot denetleyicisine gönderilir. Bu çalışmada robotun sabit makinelerin aksine kararlılığının az olduğundan, dalgalanmasını gidermek amacıyla geliştirilmiş çapak alma takımı kullanılmıştır.

Jayaweera ve Phil (2010) yaptıkları çalışmada bir lazer algılayıcı kullanarak parçanın bazı kenar noktalarını ölçüp, ardından matematiksel modellerin yardımıyla geometrisini tanıyıp ve ona göre bir robot hareket planı çıkarıp uygulamışlardır. Bu tip robot hareket planlama yöntemi parçanın bağlanması, boyut varyasyonu ya da robottan kaynaklanan hataları telafi edebilmektedir. Bu yöntem kullanıldığı durumda iş parçasının tutarlı bağlanması için genelde pahalı bağlama aparatlarına gerek kalmadığı için basit bir sabitleme kelepçesi kullanılabilir. Örnek titanyum iş parçası üzerinde Comau s4

robotu, PushCrop SM3002 spindle motoru ve Meta görü programı kullanılarak test edilmiştir.

Mizukawa vd. (1998) yaptıkları çalışmada belli bir göreve dönük robotlar kapsamında görevsel ve sistemsel robot tasarım yöntemini robotla çapak alma sistemlerine uygulanmıştır. Robotlaşma eylemi için genelde iki geleneksel robot tasarım yöntemi kullanılmaktadır. Her iki yöntem hareket kontrolüne dönük bir tasarım yöntemi olarak nitelendirilebilir. Tepeden aşağıya yöntemi; bu yöntem var olan bir görevin robotlaşması için hazır olan robot kalıpları kullanılmaktadır. Aşağıdan tepeye tasarım yöntemi ise; belli fiziksel süreçlerin analizlerini yaparak görevin isterlerini çıkartıp ve onları toplayarak hareket dizilişi belirlenmektedir. Bu çalışmada çapak alma robotunun tasarımı için kullanılan yöntemlerin aksine robot tasarımındaki üç unsura (robotun göreve dönük tasarım sistemi, eğitim sistemi ve görev gerçekleştirme sistemi) aynı ağırlık verilmiş, çapak alma robotu bu yöntemeye dayalı dört alt sisteme (ölçme, yol üretme, hareket denetlemesi, eğitime ve gözetim alt sistemi) ayrılıp tasarlanmıştır.

Greene (1984) yaptığı çalışmada GE P50 robot kullanılmış. Bu çalışmada daha büyük iş parçalarının çapak alma işleminin gerçekleştirilmesi için takım taşınması planlanan ve hafif yük taşıyan bir kol robotu kullanılmıştır. Yol planı için açık denetleme döngüsü kullanılmış ve pozisyonun geri beslemesi olmadığı için robotun iş parçasının dairesel kenarlarını takip etmesi için etiketleme işleminde doğrusal interpolasyon kullanılmaktadır ve robotun hareketi noktadan noktaya doğrusal bir harekettir. Bu yaklaşım kullanıldığı için yüzeyin pürüzlülüğü yeterince azaltılmamış, robot kolu yeterli sertlik göstermemiş, robot orta hızda hareket etmesine rağmen kararsızlığı ve konum aşmasına maruz kalmıştır. Bu yaklaşımı kullanarak bir parçadan çapak alma işlemi yapılması çapak almak yerine parçayı bozmaktadır. Bu yüzden robotun daire çizme kabiliyetine bağlı bir yöntemeye başvurulmuştur. Fakat robotun yunuslama açısı 180°den 135'e bir dönme yapabildiği için, bir daire ancak iki turda çizilmektedir. Diğer bir problemde dairenin merkez noktasının belirlenmesidir. Bu problemlere rağmen yeni yöntemin kullanılmasının yüzey pürüzlülüğünü iyileştirdiği bildirilmiştir.

Ryuh ve Pennock (2006) yaptıkları çalışmada robotla çapak alma süreci için yaşanan bir durumu ele almışlardır. Genelde mühendisler çapak alma işlemini

robotlaştırma kararı aldıkları zaman, yapılan uzun ve sayısı büyük referanslara ve makalelere başvurmak yerine deneme-yanılma yöntemini tercih ederler. Bu makalede deneme-yanılma sürecinin kısaltılması, takımın türü ve materyalinin belirlenmesi ve robot yazılım sürecini kolaylaştırmak için önemli detaylar sunulmuştur.

Bu çalışmaya göre, iş parçası işleme sırasında oluşan reaksiyon gücü emilmediğinden ve reaksiyon çapak boyutu değişince değiştiği için, robotun maruz kalacağı titreşimin azaltılması amacıyla uyumlu takım taşıyıcı kullanılması gerekmektedir. Bir, iki ve çok yönlü bir takım taşıyıcı vardır. Takımın da çeşitli uyum tipleri vardır ve ona göre takımın tasarımı değişmektedir. Robotun programlama aşamasında robotun uygulayacağı kuvvet uyum yönlerine (takım ve taşıyıcının) uyması gerekmektedir. Onlara göre, robotun karşılaşacağı şartlar sabit ise (çapakların en, kalınlık, maddenin özellikleri) pozisyon hareket kontrolü uygulanması yeterli olacaktır. Ancak genelde çapak alma işleminde değişken parametreler vardır:

- Çapakların şekil ve boyut değişikliği.
- Çapaklara yaklaşma yönü.
- Robotun tekrarlanabilirliği.

Yukarıdaki problemlerin çözülmesi için aşağıdaki çözümlere başvurulması tavsiye edilir:

- Yeterli uyum mekanizması sağlanması gerekmektedir.
- Aşırı temas gücünden kaçınmak için robotun hızının kontrol edilmesi gerekmektedir.
- Çapakların boyutlarının değişmesinden dolayı adaptif kontrolün uygulanması gerekmektedir.
- Robotun güç kontrolü yapılması gerekmektedir.

Genelde; yolun hatası, takımın gürültüsü ve takımın aşınması robotla çapak alma işleminde karşılaşılabilecek ciddi sorunlardır. Schmidt vd. (2021) yaptıkları çalışmada, hassas parçalar, işçiler tarafından istenen duruma gelene kadar, işleme tabi tutup kontrol edilir. Fakat işlemin kapasite ve maliyetinin sürekli iyileştirilmesi gerekmektedir, bu yüzden bu alanda otomasyon ve robotlaşma uygulanmıştır. Elle çapak alma işleminde yapıldığı gibi

yüksek kalite sağlanması için parçaların her birisi için robotun ayarları ve kontrolünün yeniden yapılması gerekmektedir. Geleneksel çapak alma robotlarının genel tasarım kriterleri; i) Robot erişebilme kabiliyeti, bütün parçanın etrafına ulaşacak şekilde yapılmaktadır, ii) Takım aletinin parametreleri ve geri besleme (feed rate) oranı, genelde çapağın kalınlığına bağlı olmayıp parçanın bütünü için sabit olarak yapılmaktadır. O zaman çapak alma işlemi esnasında, robot çapağı olmayan yerleri de gezecektir. Bu durumda hem fazladan zaman harcanır hem de parçanın toleransı azalır. Çalışmaya göre robotla çapak alma sistemlerinin karşılaşılabilecek problemlerin kaynakları:

- Takımın temas gücü ve kesmenin şartları.
- Robotun kısıtlı hareket hassasiyeti.
- İş parçasının değişimi ve konumundaki hata.

Bu çalışmanın önceki çalışmalardan farkı, çapak alma sürecini etkileyen bütün faktörlere rağmen çapak alma işleminde takım ve iş parçasının sadece sabit temasının sağlanması amaçlanmamıştır. Bu çalışma kapsamında iş parçalarından her birisi için kalite kontrolü yapılarak, çapakların alınması doğrultusunda çapakların konumlarını bulma ve özelliklerine göre, özgün ve optimal yol planlaması ve ideal parametrelerin ayarlaması için hatasız akıllı sistem hazırlanmıştır. O sistemin çalışma aşamaları:

- Nominal işlem planlaması.
- Hataların tahmini.
- Hataların önlenmesi.
- Hataların bulunması.
- Hataların onarımıdır.

Yapılan çalışma kapsamında robotla hatasız bir çapak alma işlemi yapılması için çevrimdışı (offline) çapak konumlarını ve oluşabilecek robot hatalarının tahmini yapıp, düzelttikten sonra hatalardan arındırılmış bir yol planı yapılmasının ardından, robota iletilen işlem gerçekleştirme esnasında işlenen hataların gerçek zamanda (real time) düzeltilmesi için izlenim ve geri besleme kullanılmaktadır. Yapılmış çalışmada, çapak alma robotu parça elde (part in hand) yöntemi kullanmakta ve sistem 5 bağımsız fiziksel ve programsal ayrı modüllere ayrılmaktadır:

- Parçaların beslemesi ve ayrımı.
- Parçayı taşıyıcı operatörün seçimi.
- Parçanın tutma hatasının giderilmesi.
- Çapakların tespiti.
- Çapak alma işlemi.

Onstein vd. (2020) yaptıkları çalışmada çapak alma robotlarının durumu (status) ve kullanımı (usage) hakkında rapor hazırlanmıştır. Özellikle döküm parçalarındaki robotla çapak alınması ve robotla çapak alma işlemini kolaylaştırabilecek ek sensörler üzerinde durulmuştur. Yapılmış çalışmaya göre çapak alma robotlarının kurulmasında iki aşama vardır; birincisi yol planlaması, ikincisi ise hareket icrası (execution). Yol üretmesi için üç yaklaşım vardır:1) İnsan tanıtılmasıyla yol eğitilmesi, -2) CAM/CAD programların vasıtasıyla yol üretilmesi, -3) Sensörlerin yardımıyla çapakların bulunup yol üretilmesidir. Çalışmaya göre çeşitli değişimler nedeniyle, standart iş parçasıyla örtüşmeyen işlenecek olan iş parçası ve önceki yöntemlerle planlanan yolların konum hatasını gidermek için iş parçasını tarayıp, taramadan çıkan nokta bulutu önceki yöntemlerle üretilen yol ile karşılaştırma yapıldıktan sonra yolda gereken değişiklikler yapılmaktadır. Hazırlanan makaledeki üç planlama yönteminin kısa anlatımı:

İnsan tanıtılmasıyla yol eğitilmesi (gösterimi): Robot ucuna asılı sensor takarak, gidilecek noktalara robot ucu götürülüyor, bazen temas gücü de kuvvet sensörü kullanarak ekleniyor. İş parçasının CAD çizimi kullanılarak ekrandaki noktaları tayin etmek, ya da işaret kalemleriyle parçanın üstüne çizerek, ardından görüntü sistemiyle işaretleri algılayıp çapakların pozisyonları belirtilmektedir. Belirlenmiş noktalardan yol planlama programında bir çalışma güzergahı hazırlanılır, eğer eğik bir kenar varsa; bu yöntemle bütün parça kenarlarını gezilmesi için büyük sayıda etiket noktalarının belirlenmesi gerekecektir, bu kısım çok vakit ve tecrübe istemektedir.

CAM/CAD programları vasıtasıyla yol üretilmesi: CAD programı, parçanın üretimi ve tasarımı için kullanılan bir programdır. CAM/CAD programlarından birisi işleminin planlaması özellikle takımın hareket planlaması için kullanılır. CAM programları vasıtasıyla yol planlaması, yol üretilmesi ve simülasyonu yapılabilmektedir. CAM

programının çıkışında genelde numerik kodu (C- kodu gibi) ya da robotların özel dillerinde yol planı, kod olarak verilmektedir.

Sensörler yardımıyla çapakların bulunup yol üretilmesi: Kamera ile çapaklar bulunmakta ve bazen önceki yöntemlerle beraber çalışarak yol planı hazırlanmaktadır. Çalışmaya göre parçanın kendisi, sabitlemesi ya da başka sebeplerden dolayı planlanmış yol parçanın kenarlarıyla örtüşmeyebilir. Bu hatanın telafisi için parçanın standart CAD çizimi nokta bulutu ve sensörlerden gelen parça ölçümleri bazı algoritmalar kullanarak karşılaştırılıp, yol planının parçada oluşmuş değişikliklere göre tadil güncellenmesi gerekmektedir.

Pullkinen vd. (2008) yaptıkları çalışmada robotla maden işleminin genelde yığın seri imalatta tercih edildiğini belirtmişlerdir. Kısıtlı seri imalatta robotsal maden işlemleri için robotun yazılımının yazılması uzun zaman aldığı için masraf açısından tercih edilmez. Bu çalışmada hem robotun hızlıca programlanması hem de 2D iş parçalarının CAD çizimi kullanılarak 3D parçanın üzerine işlem yapılabilmesi için bir programlama yaklaşımı sunulmuştur.

Bükme maden işleme yönteminde, genelde iş parçasının işlenmesi için sadece iki boyutlu CAD çizimi verilir. İş parçasının kesim ve delik açma işlemlerinden deformasyona maruz kalmaması için bu işlemler bükme yapıldıktan sonra uygulanır ve çapak alma işlemi son olarak icra edilmesi gerektirir. Ancak 3 boyutlu iş parçası için birçok zaman sadece 2 boyutlu bir çizim bulunduğundan, 2 boyutlu çizime (bükme, kesim, açılar ve kalınlık) bilgiler ekleyerek 3 boyutlu bir yol planlama kaynağı sağlanabilmektedir. Bu yöntemde malzemenin parçalar olarak programlanması ve parçanın kenarlarının etiket noktalarıyla eşleştirilmesi yapılmaktadır. Çapak alma yol planlaması bu çalışmada 3 aşamadan geçmektedir:

- Görevin planlaması.
- Hareket planlaması.
- Düzeltilme ve onaylama.

Yol planı ve kontrolü (erişebilme ve çarpma) çevrimdışı (offline) olarak yapılmaktadır. Yol planından çıkan etiket noktalarına (çalışma noktaları) robotun kesintisiz (parçanın yerleştirme açıları değiştirmeye gerek kalmadan) hareket edebilmesi için ve iş parçasının robot hareketi engellenmeyecek şekilde robotun açıları hesaplanmaktadır.

2.2 Çapak Alma İşlemi

Parçanın nihai şeklinin verilmesi için çeşitli maden işlerinin sonunda yapılması zorunlu olan bir iştir ve çapak alma yöntemlerinden birisini uygulayarak gerçekleştirilmektedir. Çapaklar; malzemenin işlendikten sonra yüzeyinden uzanan küçük maden parçaları olarak nitelendirirler, bitirilmiş bir malzemede bile çapak bulunmaması iddialı bir iştir, genelde maden işleme sırasında çapak oluşumunu asgariye indirmek ve takip eden imalat süreçlerine etkilememesi için oluşum yönünün denetlenmesine çalışılmaktadır.

Gillespie'ne (1980) göre çapakların tanımlanması kaynaklara göre tartışmalı bir meseledir, bazı kaynaklara göre küçük çapaklar ya da parçaların birleştirilmesini etkilemeyen çapaklar, çapak olarak sayılmaz.

Çapaklar parça üzerinde bulunduğu durumda parçanın tipini bozar ve anlaşmaya göre üretilmiş ürünün kalitesinde varyasyon aralığından çıkarak, uygunsuz bir ürün olarak nitelendirilmesine sebep olur. Tazminat, iade, onarım hizmetinin istenmesi ya da şirketin itibarının sarsılmasına bir neden olabilecektir. Bu sebeplerden dolayı, çapak alınması üretim sürecini uzatacak olmasına rağmen zorunlu bir işlemdir. Malzeme ve çapakların kimyasal ve fiziksel özellikleri, parçanın işlenmesi ve bitirme türü çapak alma işlemini etkilemektedir. Çapak alma işlemini etkileyen parçanın karakterlerini Gillespie'ne (1980) göre çizelge 1'de belirtilmektedir.

Endüstride çapak alınması için madene göre; termik, elektrokimyasal, mekanik ve kimyasal metotlara (sürecin masraflarını göze alarak) başvurulmaktadır. Bazı özel veya melez yöntemleri de (Elektrokimyasal Taşlama) vardır, fakat bu araştırmanın konusu çapak alma yöntemleri olmadığı için, bütün yöntemleri kapsamaz.

2.3 Çapak Alma Yöntemleri

Mekanik çapak alma metodu: Su jetiyle veya aşındırıcı kemerler, diskler, takımlar, fırçalar, el ile veya bir makine ucuna takılarak ya da yuvarlama taşları kullanarak yapılmaktadır.

Yuvarlama taşları: Malzemenin aşındırılması ve çapak alınması için sert taşları araç olarak kullanan bir metottur. Yuvarlama taşları malzemeye sabit bir aşındırma oranı uygulamaktadır. Taşların boyutları, şekilleri ve maddeleri sürecin özelliklerine göre belirlenmektedir.

Çizelge 2.1: Çapak Alma İşlemini Etkileyen Parçanın Karakterleri

Maddenin özellikleri	İşletme prosesi	Çapağın özellikleri	Çapak alma işlemi	Parçanın Özellikleri
Sertlik derecesi	Tipi	Sertlik derecesi	Tip	Toleransı
Dayanıklılık derecesi	Takım tasarımı	Dayanıklılık derecesi	Döngü süresi	Kenarın gereksinimleri
Termal iletkenliği	Soğutma	Enlilik	Çapağın boyutu	Boyutu
Özgül ısı	Çalışma hızı	Uzunluk	Masraf	Geometrisi
Yoğunluk	Takım aşınması	Abrazyon direnci	Tekrarlanabilirlik	İşi
Kimyasal kompozisyon	Masraf	Termal iletkenliği	Teslim süresi	
Abrazyon direnci		Kimyasal kompozisyon	Miktarı	
Metalurjik faz		Özgül ısı	Çapakların konumu	
Bükme kuvveti		Kütle		
Nihai kuvvet		tekrarlanabilirlik		
Eritme noktası				

Termal enerji metodu: Çapakların giderilmesi için malzemenin bazı karışım gazlar kullanarak iş parçası kapalı bir yerde yakılmaktadır. Çapaklar malzemenin yüzeyinde bulunduğu ve ince olduğu için kısa bir süre içinde verilen yüksek ısı ile emilip buharlaşmaktadır

Kimyasal çapak alma metodu: Aşındırıcı kimyasal çözeltiler kullanarak kimyasal reaksiyonlar neticesinde gerçekleşen bir süreçtir.

Elektrokimyasal çapak alma metodu: Anot, katot ve elektrolit kullanarak elektrokimyasal reaksiyonlar neticesinde gerçekleşen bir süreçtir. Genelde, çapaklar tekdüzelik değil, düzensiz bir şekilde bulunmaktadırlar. Gillespie'ne (1980) göre çapakların parçadan parçaya, kenardan kenara ve kenarın içinde değiştiği bilinmektedir ve işlem sırasında bazı öbür özellikler, parçayla bağlanma gücü ya da keskinliği gibi önemsenmemektedir.

Bazı eski yöntemler çapak alma işlemi için verimli ve tekrarlanabilen oldukları halde tamamlanmamış çapak alma işleminin olasılığının çok yüksek olduğu, girilmesi zor yerlerin temizlenmemesinin mümkün olduğu ve geçmiş temizleme periyotlarından kalan çapak artıklarının parçaları kirletmesinin olası olduğu belirtilmiştir.

Eski yöntemlerin kullanılması Gillespie'ne (1980) göre, parçanın bitirme ve boyutlarını etkilemektedir, parçanın biçimini bozar ve kalıntılı stresi çeşitlendirir, esnekliğin limitini azaltır, yorulma mukavemetini azaltır, rengi ve parlaklığını değiştirir, yüzeyini kirletir, paslanma direncini azaltır, özdirenci yükseltir ve malzemenin yüzeyini aşındırır.

Gillespie'ne (1980) göre de tam bir çapak alma işlemi; malzemedeki fazlalıkları kaldırarak, onun biçimini bozarak, hassas yüzeyini çizerek ya da onaylanabilecek bir aralıkta kenarını kırarak, iş parçasının boyutlarını etkilemeyen bir süreçtir. Bu yüzden malzemedeki çapak alındığında yan etkilerden kaçınmak amacıyla, özellikle bazı özel kriterler taşıyan hassas malzemeler için, en uygun çapak alma yöntemi bazı kesici aletler kullanmaktır. Bu işlem el ile yapılabilmektedir ama verimi nispeten az olur. Fakat

makinelerin işleme ile %100'e yakın bir verim yakalanabilmektedir ve tüm kenarların aynı kırma açısıyla işleme garantisi elde edilecektir.

Johannesen vd.ne (1986) göre el ile çapak alma işlemi, bir tezgâh işidir, işlem sırasında alet el ile tutulur ya da parça ele alınarak sabit bir takıma karşı işlenir. El ile kesici alet ya da motorlu alet kullanılarak yapılır. İşlem sırasında, el aletlerinin uçları özel ve bazı kaplamalı madenlerden yapılır, farklı kenar, oluk, direklerden çapak alabilmeleri için özel bir şekilde tasarlanırlar, ya da el ile çapak alması için farklı çalışma hızları sahip olan pnömatik matkaplar uçlarına çeşitli takımlar takarak kullanılmaktadır.

Birçok çalışmaya göre el ile çapak alma işlemi Endüstride en yaygın kullanılan çapak alma yöntemidir. Bu metot esnek, genişletebilen ve büyük bir yatırıma ihtiyaç duymamasına rağmen kararlı bir metot değildir. Johannesen vd.ne (1986) göre ABD 1980'li yıllarda el ile çapak alma işleminin masrafı 2 milyar dolara ulaştı.150,000 eleman çapak alma işçiliğini yapmaktadırlar, işgücünün %3'ü ve fabrikaların giderlerinden %10'unu teşkil eder.

Bu metot için yeteneklilik ve konsantrasyon gerekmektedir, uzun üretim süreci ve çalışma saatleri için ürünlerin tekdüzeliğinin sağlanması imkansızdır. İşçilerin devamsızlıkları ve iş değiştirmeleri bu süreç için kalıcı bir sorundur, yeni elemanların iyi bir eğitim verilse dahi iş kalitesi ve işlem sürecinde ciddi bir değişiklik yaşanacaktır. Her zaman yığın üretim için el ile çapak alma metodu bir dar boğazdır.

Karmaşık ve hassas parçaların (dişli gibi), küçük bölge içinde işlenecek yerlerinin sayısı çok olduğu için üretim kriterleri el ile gerçekleştirilmesi epey zordur, eğer parçanın bir bölümü çapak alma işlemi sırasında bozulmuşsa, ya da düşürülmüşse parça kullanılmaz olup tekrar üretilmesi gerekir.

Dişli gibi parçaların temizlenmesi için temizleme aletleri yapılmıştır, fakat o aletlerde önemli sıkıntılar bulunmaktadır. Bu temizleme aletleri belli tip ve boyutta iş parçalarını temizlemek için kullanılır, farklı tip ve boyuttaki parçaları temizlemek için uzun bir hazırlık süresi gerekir. Takım hareketi kısıtlı olduğundan dişli bir eksene

bağlanarak döndürülmekte, dönen iş parçası takımla yüksek temas gücüne maruz kaldığı halde, parçanın işleme esnasında oynatılmasına neden olur ve yüzey pürüzlülüğünü artırır.

Robotla çapak alma işlemi sırasında, temas gücü büyük olsa bile, iş parçasının tutulması sağlam olduğu için iş parçası yerinden kaymaz. Robot iş parçasına karşı istenen pozisyon ve yönelime gelebildiği için çapak alma işleminde robotun kullanılması tercih edilir.

Schmidt vd.ne (2021) göre robotla çapak alma işlemi oluşan küçük çapakların alınabildiği ve yanı sıra büyük çapaklar kaldırılabildiğinden dolayı tercih edilen bir çapak alma yöntemidir. Eskilerden robotun pahalılığını, yüksek yatırım maliyeti, uzman azlığı ve bunlardan daha çok Gillespie'ne (1980) göre robotun doğruluk payı yeterli olmadığı için çapak alma işleminde kullanımı yaygın değildi. Günümüzde piyasada fiyat ve tutarlılık bakımından uygun robotlar bulunduğu için kullanımı biraz daha yaygınlaşmıştır.

Ryuh ve Pennock'e (2006) göre günümüzde robotlar basit kesim ve çapak alma işlemi yapmaktadırlar. Schmidt vd.ne (2021) göre az ve orta hassasiyetli işlemler yapılabilmektedir. Robotun bu süreçleri icra edebilmesi için (özellikle robotun bileğinde) lazım olan iki önemli özelliği bulunması gerekmektedir: sertlik ve katılık (stiffness and rigidity). Piyasada bulunan endüstriyel robotlar bilhassa bileklerinde sertlikten yoksunlardır, bu yüzden bu tip işlemler için pek uygun değillerdir. Bahsedilen işlemler sırasında robot; güç ve tork reaksiyonu ve istenmeyen titreşimlere maruz kalmaktadır. Genelde endüstriyel robotlar ucuz bir maliyete tasarlanıp daha basit bir ihtiyaçları karşılamaları için yapılmışlardır.

Ryuh ve Pennock'e (2006) göre de robotun çapak alma konusunda en yaygın kullanım alanları:

- Döküm işleminden sonra çapak alma işleminde.
- Yanma motorun parçaları, krank milleri ve kam millerinden çapak alınmasında.
- Mutfak aletlerinin cilalanması ve parlatılmasında ve plastiklerin kesilme ve delik açılmasında.

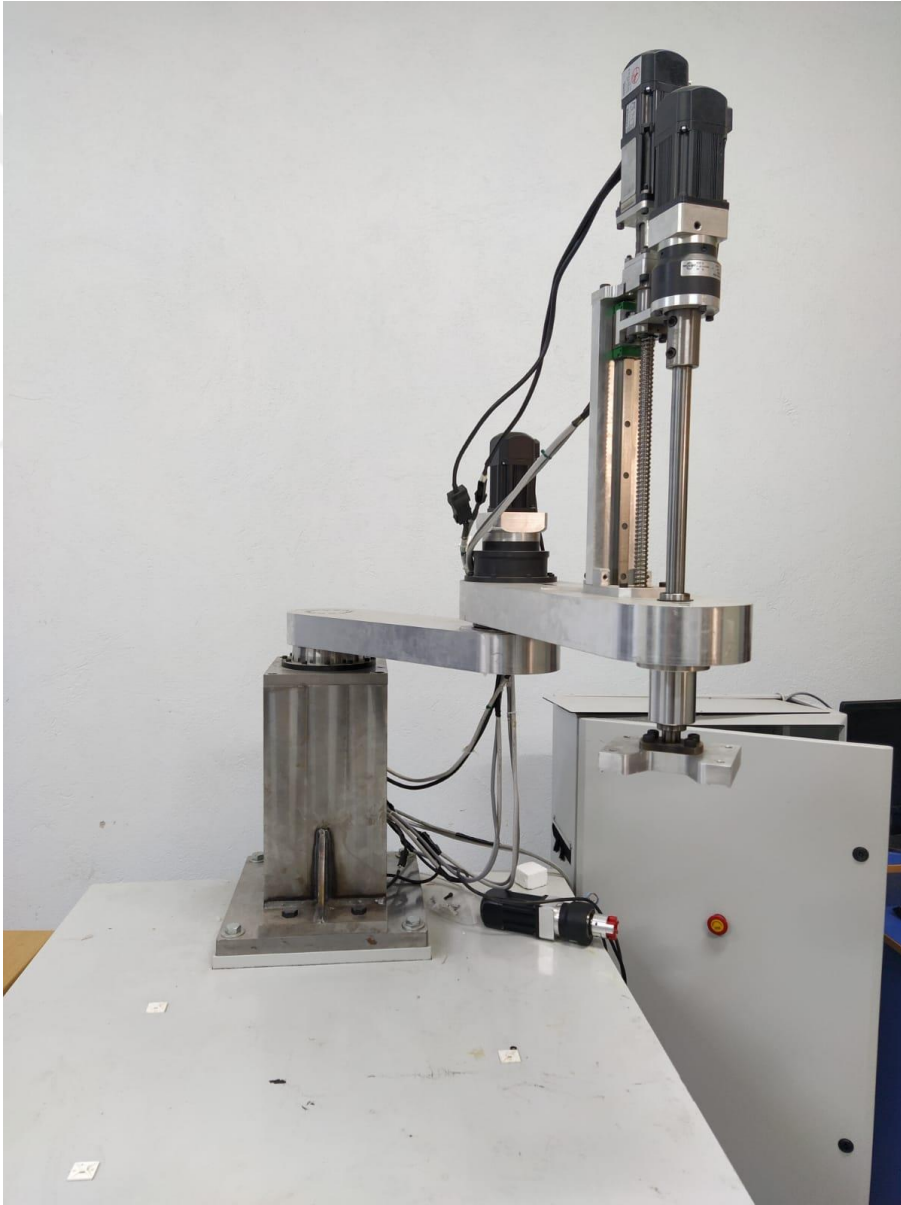
Onstein vd. (2020) göre karmaşık ve az sayıda iş parçalarındaki çapakların robot yerine el ile alınması daha karlıdır, çapak alma işleminin robotlaşmasında karşı çıkan zorluklar (challenges):

- 1- Kurma/programlama metodunun geliştirilmesi.
- 2- Etkili gücün sınırlandırılması (limit the impact force).
- 3- Sürecin parametrelerinin adaptasyon algoritması.
- 4- Kullanma ve onarımı için uzman gerektirmeyen, sıradan kullanıcıya yakın bir sistem olması.
- 5- Sürecin geliştirilmesi için çapakların yer ve boyut tahmini.
- 6- Endüstriyel ihtiyaçları karşılayabilmesi için sistemin tutarlılığının geliştirilmesi.

3. MATERYAL VE YÖNTEM

3.1 Giriş ve İleri Kinematik

Eskişehir Osmangazi üniversitesinde yapılan ve bu çalışmada kullanılacak olan SCARA robot Şekil 1’de görülmektedir.



Şekil 3.1: Özel Tasarım SCARA Robot Kolu (Eskişehir Osmangazi Üni.)

Robot ucunun pozisyonunun, eklem deęişkenlerine göre bulunabilmesi için robotun ileri kinematik denklemleri hesaplanması gerekmektedir, bunun için Denavit-Hartenberg yöntemine başvurulmuştur.

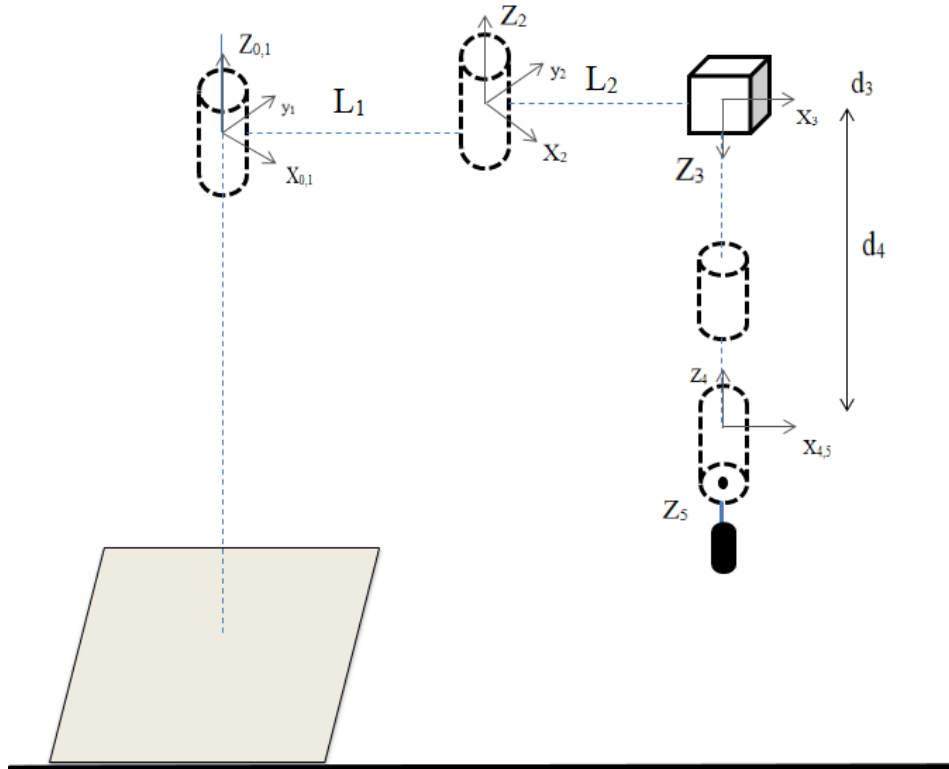
İleri kinematik dönüşüm matrisleri hesaplanması için gerekli olan parametrelerin çıkarılması amacıyla eklemlerin koordinat sistemleri ilk önce yerleştirilir (Şekil 3.2) ve (Şekil 3.3). Laboratuvarında bulunan robot 4 eksenli bir robottur, bizim çalışmamızda robotun çalışma noktalarına farklı oryantasyonla ulaşabilme kabiliyeti kazandırılması için simülasyon ortamında 5 inci eklem eklenmiştir (Çizelge 3.1).



Şekil 3.2: Koordinat Sistemlerin Robotun Üzerine Yerleştirilmesi

Çizelge 3.1: 5 Eksenli Robotun Denavit-Hartenberg Parametreleri

\dot{I}	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	180°	L_2	d_3	0
4	0	0	d_4	θ_4
5	90°	0	0	θ_5



Şekil 3.3: 5 Eksenli Robotun Kinematik Diyagramı

Robotun simülasyonu için Peter Corke tarafından geliştirilen kütüphane kullanılmıştır (Corke, 2017). Peter Corke kütüphanesine göre Denavit-Hartenberg parametrelerin yeni düzenlemesi çizelge 3.2’de gösterilmektedir.

Çizelge 3.2: Peter Corke Kütüphanesine göre Denavit-Hartenberg Parametreleri.

\dot{I}	α_i	a_i	d_i	θ_i
1	0	L_1	0	θ_1
2	180	L_2	0	θ_2
3	0	0	d_3	0
4	90	0	d_4	θ_4
5	0	0	0	θ_5

5 eksenli SCARA robotu için dönüşüm matrisleri:

$${}^0T_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$${}^1T_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & L_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$${}^2T_3 = \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$${}^3T_4 = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & 0 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$${}^4T_5 = \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$${}^0T_5 = {}^0T_1{}^1T_2{}^2T_3{}^3T_4{}^4T_5 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$r_{11} = \cos \theta_5 \cos \theta_4 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) + \cos \theta_5 \sin \theta_4 (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) \quad (3.7)$$

$$r_{11} = \cos \theta_5 \cos(\theta_1 + \theta_2) \cos \theta_4 + \cos \theta_5 \sin(\theta_1 + \theta_2) \sin \theta_4 \quad (3.8)$$

$$r_{11} = \cos(\theta_1 + \theta_2 - \theta_4) \cos \theta_5 \quad (3.9)$$

$$r_{12} = -\cos \theta_4 \sin \theta_5 (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) - \sin \theta_4 \sin \theta_5 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \quad (3.10)$$

$$r_{12} = -\cos \theta_4 \sin \theta_5 \sin(\theta_1 + \theta_2) - \sin \theta_4 \sin \theta_5 \cos(\theta_1 + \theta_2) \quad (3.11)$$

$$r_{12} = -\sin(\theta_1 + \theta_2 + \theta_4) \sin \theta_5 \quad (3.12)$$

$$r_{13} = \cos(\theta_1 + \theta_2) \cdot \sin(\theta_4) - \sin(\theta_1 + \theta_2) \cdot \cos(\theta_4) \quad (3.13)$$

$$r_{21} = \cos \theta_4 \cos \theta_5 (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) - \sin \theta_4 \cos \theta_5 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \quad (3.14)$$

$$r_{21} = \cos \theta_5 \sin(\theta_1 + \theta_2) \cos \theta_4 - \cos \theta_5 \cos(\theta_1 + \theta_2) \sin \theta_4 \quad (3.15)$$

$$r_{21} = \sin(\theta_1 + \theta_2 - \theta_4) \cos \theta_5 \quad (3.16)$$

$$r_{22} = \cos \theta_4 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) - \sin \theta_4 (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) \quad (3.17)$$

$$r_{22} = \sin \theta_5 \sin \theta_4 \cos(\theta_1 + \theta_2) - \sin \theta_5 \cos \theta_4 \sin(\theta_1 + \theta_2) \quad (3.18)$$

$$r_{22} = -\sin(\theta_1 + \theta_2 - \theta_4) \sin \theta_5 \quad (3.19)$$

$$r_{23} = \sin(\theta_1 + \theta_2) \cdot \sin \theta_4 + \cos(\theta_1 + \theta_2) \cdot \cos \theta_4 \quad (3.20)$$

$$r_{31} = -\sin \theta_4 \quad (3.21)$$

$$r_{32} = -\cos \theta_5 \quad (3.22)$$

$$r_{33} = 1 \quad (3.23)$$

$$p_x = L_1 \cos \theta_1 + L_2(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \quad (3.24)$$

$$p_x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (3.25)$$

$$p_y = L_1 \sin \theta_1 + L_2(\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) \quad (3.26)$$

$$p_y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (3.27)$$

$$p_z = -d_0 - d_3 \quad (3.28)$$

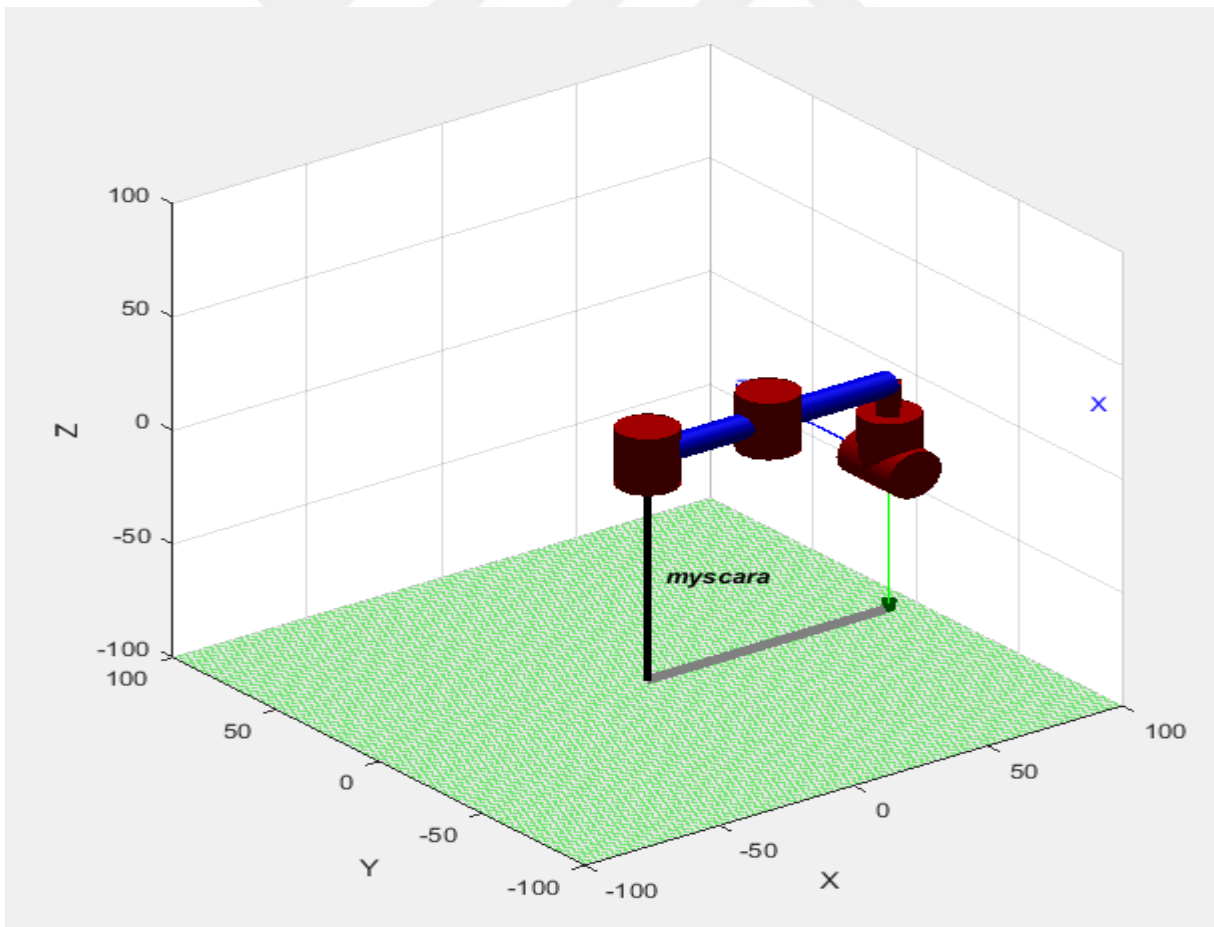
Kullanılan SCARA Robotun ölçümleri ($L_1=45\text{cm}$, $L_2=45\text{cm}$, $d_4=10\text{cm}$) olarak verilmektedir. Robotun Peter Corke kütüphanesine göre İleri kinematik programı ve simülasyon ortamında robotun gövdesi, Şekil 3.4 ve Şekil 3.5'te verilmektedirler.

```

1
2 %%5eksenli robotun ileri kinematik programı
3 % 5 eksen tanıtımı
4
5 L(1) = Link([0 0 45 0]);
6 L(2) = Link([0 0 45 pi]);
7 L(3) = Link([0 0 0 0 1]);
8 L(4) = Link([0 10 0 pi/2]);
9 L(5) = Link([0 0 0 0]);
10 L(3).qlim = [0 45];
11
12 robo=SerialLink(L);
13 robo.name='myscara';
14 robo.plot([0 0 25 0 0], 'workspace', [-100 100 -100 100 -100 100])
15

```

Şekil 3.4: İleri Kinematik Programı



Şekil 3.5: Simülasyon Ortamında Robot Gövdesi

3.2 Ters Kinematiğin Hesaplanması

Ters kinematik hesaplanması için iki yönteme başvurulmaktadır,1) Cebirsel yöntem,2) geometrik yöntem.

Aşağıdaki denklemler, ters kinematiğin cebirsel yöntem kullanarak hesaplanmasıdır

$$p_x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (3.29)$$

$$p_y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (3.30)$$

$$p_x^2 + p_y^2 = L_1^2 + L_2^2 + 2L_1L_2 \cdot \cos(\theta_2) \quad (3.31)$$

$$2L_1L_2 \cos \theta_2 = p_x^2 + p_y^2 - (L_1^2 + L_2^2) \quad (3.32)$$

$$\cos \theta_2 = \frac{p_x^2 + p_y^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (3.33)$$

$$\sin \theta_2 = \sqrt{1 - (\cos \theta_2)^2} \quad (3.34)$$

$$\tan \theta_2 = \frac{\sin \theta_2}{\cos \theta_2} \quad (3.35)$$

$$\theta_2 = \tan^{-1} \frac{\sin \theta_2}{\cos \theta_2} \quad (3.36)$$

$$p_x = L_1 \cos \theta_1 + L_2(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \quad (3.37)$$

$$p_x = \cos \theta_1 (L_2 \cos \theta_2 + L_1) - L_2(\sin \theta_1 \sin \theta_2) \quad (3.38)$$

$$p_y = L_1 \sin \theta_1 + L_2(\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) \quad (3.39)$$

$$p_y = \sin \theta_1 (L_1 + L_2 \cos \theta_2) + L_2 (\cos \theta_1 \sin \theta_2) \quad (3.40)$$

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} L_2 \cos \theta_2 + L_1 & -L_2 \sin \theta_2 \\ L_2 \sin \theta_2 & L_1 + L_2 \cos \theta_2 \end{bmatrix} \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix} \quad (3.41)$$

$$\begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix} = \begin{bmatrix} L_2 \cos \theta_2 + L_1 & -L_2 \sin \theta_2 \\ L_2 \sin \theta_2 & L_1 + L_2 \cos \theta_2 \end{bmatrix}^{-1} \begin{bmatrix} p_x \\ p_y \end{bmatrix} \quad (3.42)$$

$$A = \det \left(\begin{bmatrix} L_2 \cos \theta_2 + L_1 & -L_2 \sin \theta_2 \\ L_2 \sin \theta_2 & L_1 + L_2 \cos \theta_2 \end{bmatrix} \right) \quad (3.43)$$

$$A = (L_2 \cos \theta_2 + L_1)^2 + (L_2 \sin \theta_2)^2 \quad (3.44)$$

$$A = L_1^2 + L_2^2 + 2L_1L_2 \cos \theta_2 \quad (3.45)$$

$$\begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix} = \frac{1}{A} \begin{bmatrix} L_2 \cos \theta_2 + L_1 & L_2 \sin \theta_2 \\ -L_2 \sin \theta_2 & L_1 + L_2 \cos \theta_2 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} \quad (3.46)$$

$$\begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix} = \frac{1}{A} \begin{bmatrix} (L_2 \cos \theta_2 + L_1)p_x + L_2 \sin \theta_2 \cdot p_y \\ -L_2 \sin \theta_2 \cdot p_x + (L_1 + L_2 \cos \theta_2)p_y \end{bmatrix} \quad (3.47)$$

$$\cos \theta_1 = \frac{1}{A} \left((L_2 \cos \theta_2 + L_1)p_x + L_2 \sin \theta_2 \cdot p_y \right) \quad (3.48)$$

$$\sin \theta_1 = \frac{1}{A} \left(-L_2 \sin \theta_2 \cdot p_x + (L_1 + L_2 \cos \theta_2)p_y \right) \quad (3.49)$$

$$\tan \theta_1 = \frac{-L_2 \sin \theta_2 \cdot p_x + (L_1 + L_2 \cos \theta_2)p_y}{(L_2 \cos \theta_2 + L_1)p_x + L_2 \sin \theta_2 \cdot p_y} \quad (3.50)$$

$$\theta_1 = \tan^{-1} \frac{-L_2 \sin \theta_2 \cdot p_x + (L_1 + L_2 \cos \theta_2)p_y}{(L_2 \cos \theta_2 + L_1)p_x + L_2 \sin \theta_2 \cdot p_y} \quad (3.51)$$

$$r_{13} = \cos(\theta_1 + \theta_2) \cdot \sin(\theta_4) - \sin(\theta_1 + \theta_2) \cdot \cos(\theta_4) \quad (3.52)$$

$$r_{23} = \sin(\theta_1 + \theta_2) \cdot \sin \theta_4 + \cos(\theta_1 + \theta_2) \cdot \cos \theta_4 \quad (3.53)$$

$$\begin{bmatrix} r_{13} \\ r_{23} \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \cdot \begin{bmatrix} \sin \theta_4 \\ \cos \theta_4 \end{bmatrix} \quad (3.54)$$

$$\begin{bmatrix} \sin \theta_4 \\ \cos \theta_4 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix}^{-1} \cdot \begin{bmatrix} r_{13} \\ r_{23} \end{bmatrix} \quad (3.55)$$

$$\det \left(\begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \right) = 1 \quad (3.56)$$

$$\begin{bmatrix} \sin \theta_4 \\ \cos \theta_4 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) \\ -\sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \cdot \begin{bmatrix} r_{13} \\ r_{23} \end{bmatrix} \quad (3.57)$$

$$\begin{bmatrix} \sin \theta_4 \\ \cos \theta_4 \end{bmatrix} = \begin{bmatrix} r_{13} \cdot \cos(\theta_1 + \theta_2) + r_{23} \cdot \sin(\theta_1 + \theta_2) \\ -r_{13} \cdot \sin(\theta_1 + \theta_2) + r_{23} \cdot \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (3.58)$$

$$\sin \theta_4 = r_{13} \cdot \cos(\theta_1 + \theta_2) + r_{23} \cdot \sin(\theta_1 + \theta_2) \quad (3.59)$$

$$\cos \theta_4 = -r_{13} \cdot \sin(\theta_1 + \theta_2) + r_{23} \cdot \cos(\theta_1 + \theta_2) \quad (3.60)$$

$$\frac{\sin \theta_4}{\cos \theta_4} = \frac{r_{13} \cdot \cos(\theta_1 + \theta_2) + r_{23} \cdot \sin(\theta_1 + \theta_2)}{-r_{13} \cdot \sin(\theta_1 + \theta_2) + r_{23} \cdot \cos(\theta_1 + \theta_2)} \quad (3.61)$$

$$\tan \theta_4 = \frac{r_{13} \cdot \cos(\theta_1 + \theta_2) + r_{23} \cdot \sin(\theta_1 + \theta_2)}{-r_{13} \cdot \sin(\theta_1 + \theta_2) + r_{23} \cdot \cos(\theta_1 + \theta_2)} \quad (3.62)$$

$\cos(\theta_1 + \theta_2)$ i bölerek

$$\tan \theta_4 = \frac{r_{13} + r_{23} \cdot \tan(\theta_1 + \theta_2)}{-r_{13} \cdot \tan(\theta_1 + \theta_2) + r_{23}} \quad (3.63)$$

$$\tan \theta_4 = \frac{r_{13} + r_{23} \cdot \tan(\theta_1 + \theta_2)}{r_{23} - r_{13} \tan(\theta_1 + \theta_2)} \quad (3.64)$$

$$\tan \theta_4 = \frac{\frac{r_{13}}{r_{23}} + \tan(\theta_1 + \theta_2)}{1 - \frac{r_{13}}{r_{23}} \tan(\theta_1 + \theta_2)} \quad (3.65)$$

$$A = \frac{r_{13}}{r_{23}} \quad (3.66)$$

$$B = \tan(\theta_1 + \theta_2) \quad (3.67)$$

$$\tan \theta_4 = \frac{A + B}{1 - A \cdot B} \quad (3.68)$$

$$\theta_4 = \tan^{-1} \left(\frac{A + B}{1 - A \cdot B} \right) = \tan^{-1} A + \tan^{-1} B \quad (3.69)$$

$$\theta_4 = \tan^{-1} \left(\frac{r_{13}}{r_{23}} \right) + \tan^{-1}(\tan(\theta_1 + \theta_2)) \quad (3.70)$$

$$\theta_4 = \tan^{-1} \left(\frac{r_{13}}{r_{23}} \right) + \theta_1 + \theta_2 \quad (3.71)$$

(3.21) ve (3.22) denklemlere dayanarak

$$\frac{r_{31}}{r_{32}} = \frac{-\sin \theta_4}{-\cos \theta_4} = \tan \theta_5 \quad (3.72)$$

$$\tan \theta_5 = \frac{-r_{31}}{-r_{32}} \quad (3.73)$$

$$\tan \theta_5 = \frac{-r_{31}}{-r_{32}} \quad (3.74)$$

d_3 değerinin bulunması için (3.28)'tan faydalanarak

$$d_3 = -d_0 - p_z \quad (3.75)$$

Ters kinematiğin programı Şekil 3.6'da resimde gösterilmektedir

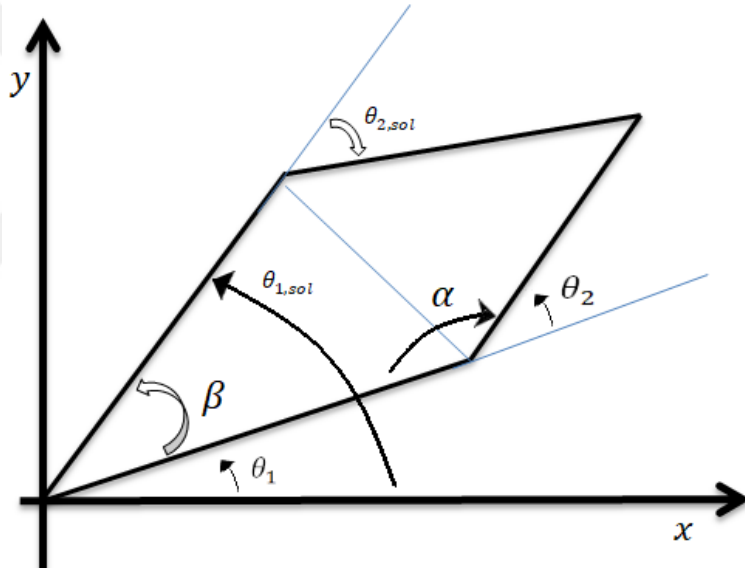
```
terskinematik5ksenlirobot.m x +
1 %5 eksenli scara robotun ters kinematiği hesaplaması için yapılan proje
2 - L1=45;
3 - L2=45;
4 - L(1) = Link([0 0 L1 0]);
5 - L(2) = Link([0 0 L2 pi]);
6 - L(3) = Link([0 0 0 0 1]);
7 - L(4) = Link([0 10 0 pi/2]);
8 - L(5) = Link([0 0 0 0]);
9 - L(3).qlim = [0 45];
10 - scararobot=SerialLink(L);
11 - scararobot.name='5ksenli scararobot';
12 - t=scararobot.fkine([pi/6 pi/3 20 pi/4 pi/6]);
13 - n=t(:,1);
14 - o=t(:,2);
15 - d=t(:,3);
16 - r=t(:,4);
17 - costh2=((r(1).^2)+(r(2).^2)-(L1.^2)-(L2.^2))/(2*L1*L2);
18 - costh2q= (costh2).^2;
19 - sinth2=sqrt(1-costh2q);
20 - th2=atan2(sinth2,costh2);
21 - sicth1=(L1+L2*cos(th2))*r(2)-(L2*sin(th2))*r(1);
22 - coth1=(L1+L2*cos(th2))*r(1)+(L2*sin(th2))*r(2);
23 - th1=atan2(sicth1,coth1);
24 - th4=atan2(d(1),d(2))+th1+th2;
25 - th5=atan2(-n(3),-o(3));
26
```

Şekil 3.6: Ters Kinematiğin Programı

3.3 Sağ ve Sol İki Eklemlili Robotun Çözümü

Bizim çalışmamız iki boyutlu bir düzlemde gerçekleştirilmiştir, robotun sadece iki dönel eklem bağlantısı kullanılmaktadır. İki eklemlili robotta, bir noktaya gidilmesi için robotun matematiksel çözümlerinden bulunan açılar sağ el ve sol el robotun çözümleri olarak iki gruba ayrılmaktadır (Şekil 3.7). Laboratuvarında kullanılan robot için Ters kinematik çözümü sağ el robotu için yapılmıştır. Laboratuvarında kullanılan robot programı başlangıcında sağ ya da sol robot seçenekleri sunmaktadır, sağ ya da sol kol robotu kullanıldığında sağ ya da sol çalışma bölgesi seçilmesi anlamına gelir. Robotla çapak alma

alanında bulunan çalışmalar, yol planlama aşaması bittiğinde simülasyon yapılır, eğer robotun iş parçasıyla bir çarpışma olasılığı varsa robotun aynı çalışma noktasına gitmesi için farklı çözümlere başvurulması gereklidir. Bu doğrultuda bir adım atmak için, çalışma başlangıcında seçilen ve değiştirilmesi imkânsız olan sağ el robotun konfigürasyonuna ilave olarak gerektiğinde uygulanmak üzere geometrik yöntemi kullanarak sol el ters kinematik çözümü de bulunmuştur. Yapılan çalışmamıza göre, sağ el çalışma bölgesi kullanıldığında sol el çözümünde bazı çalışma noktaları için eklemin limitleri aşılmaktadır. Ancak çapak alma yol planlamasında yapılan çalışmalara göre kullanılan çözümleri onaylamadan önce eklemlerin limitlerinin aşıp aşılmadığı simülasyon ortamında test edilmektedir. Sol el çözümü kullanılması durumunda eklemlerin limitleri test edilecektir.



Şekil 3.7: İki Eklemlilik Düzlemsel Robotun Sağ ve Sol Çözümü

Birinci çözümden faydalanarak ikinci çözüm çıkarılması için geometrik hesaplama yöntemi:

$$\alpha = \pi - \theta_2 \quad (3.77)$$

$$\beta = \pi - \frac{\pi - \theta_2}{2} - \frac{\pi - \theta_2}{2} \quad (3.78)$$

$$\beta = \pi - \pi + \theta_2 \quad (3.79)$$

$$\theta_{1,sol} = \beta + \theta_1 = \theta_1 + \theta_2 \quad (3.80)$$

$$\theta_{2,sol} = \theta_2 \quad (3.81)$$

3.4 İki Eksenli Robot Kolunun Çalışma Uzayı

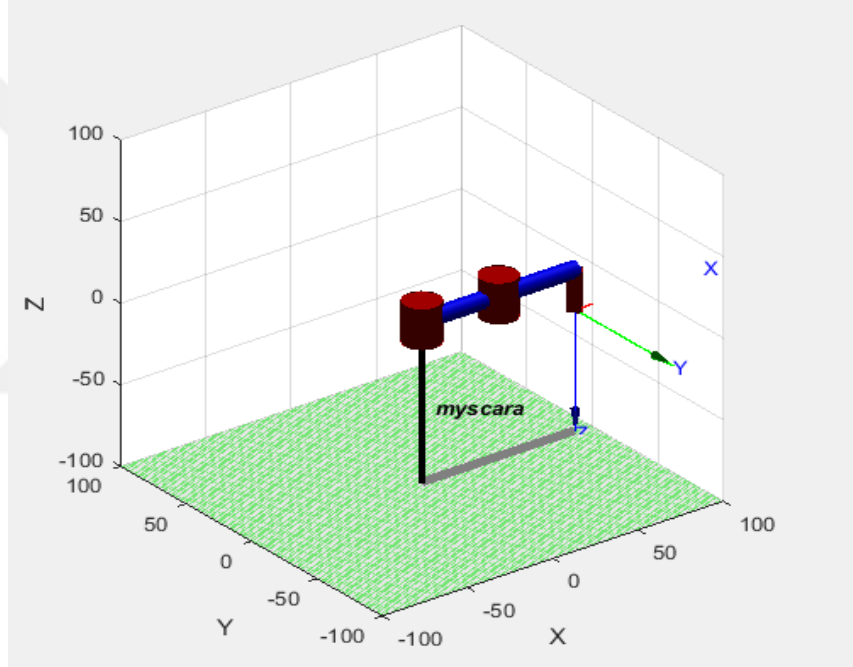
İnsanlar, kolları kullanım açısından, yapabildiği ve ulaşabildiği şeylere, öğrenme süreci bittiğinde kavrayabilmektedir ve onun kol hareket mekanizması edindiği bilgilere göre çalışmaktadır. Robot kolunun insan gibi doğru hareket planlaması için kolun imkanları ve kabiliyetini tanımak amacıyla çalışma uzayının çıkarılması gereklidir. Çalışma uzayı robotun ulaşabileceği bütün noktaların toplamı olarak nitelendirilir. Çalışma uzayının en önemli karakterleri şekli, boyutu ve bünyesidir.

Yang vd. (2014) göre çalışma uzayı çıkarılması için birçok analitik ve sayısal yöntemler vardır. Bilgisayarların performansı gelişmeden önce araştırmacılar çalışma uzayını belirlemek için yaratıcı yollara başvurdular, kullandıkları yöntemler karmaşık ve genel metotlar değildi. Genelde çalışma uzaylarının birçok tipi vardır, bizim çalışmamızda çıkardığımız çalışma uzayın adı ulaşılabilir (reachable) çalışma uzayıdır; robotun ucu en az bir oryantasyonla ulaşabildiği noktalar olarak belirtilmektedir. Bizim robotumuz sağ el olarak çalıştığı için, noktalara tek bir yönle gidilmektedir.

Çalışma alanını bulmak için kullanılan yöntem bütün eklem uzayı için ileri kinematik kullanarak; robot ucunun grafiksel olarak, konum ve oryantasyonun gösterilmesidir.

Çalışma uzayı grafiksel bir şekilde çıkarıldığında tekillik (singularity) noktaların dikkate alınması gerekmektedir. Robotun tekillik noktalarında Jacobian matrisinin determinanı sıfıra ulaştığı için robotun hızı sonsuza doğru yükselip ve kontrolden çıkmaktadır. Genelde iki tip tekillik noktaları vardır, çalışma alanının uç noktalarında olan tekillik noktaları ve çalışma alanının içinde olan tekillik noktaları. Çalışma alanının içinde

olan tekillik noktaları 3 veya daha fazla eklemlili düzlemsel robotun eklemlerinden birisi sıfıra ulaştığında ortaya çıkmaktadır. Bizim çalışmamızda iki eklemlili bir robot içerdiği için sadece çalışma alanının uç noktalarında olan tekillik noktaları bulunmaktadır. Bu noktaların bulunması ve çalışma alanına dahil edilmemesi için Jacobian matrisi elde edildiği ve sıfıra eşitleyerek tekilliğe sebep olan açılar, eklemlerin limitlerinin dışında bırakılmıştır. (3.82), (3.83), (3.84) ve (3.85) denklemlerde robotun açısal ve doğrusal eklemlerin hızların genel denklemleri gösterilmektedir. Çalışmamızda kullanılan 3 eklemlili SCARA robot Şekil 3.8’de görülmektedir



Şekil 3.8: Kullanılan 3 Eklemlili SCARA Robot

Dönel eklemler için (3.82) ve Prizmatik eklemler için (3.84)

$${}^{i+1}_{i+1}\omega = {}^{i+1}_i R {}^i \omega + \dot{\theta}_{i+1} {}^{i+1}_i \hat{Z} \quad (3.82)$$

$${}^{i+1}_{i+1}v = {}^{i+1}_i R ({}^i v + {}^i \omega \times {}^{i+1}_i P) \quad (3.83)$$

$${}^{i+1}\omega = {}^{i+1}R_i {}^i\omega \quad (3.84)$$

$${}^{i+1}v = {}^{i+1}R_i ({}^i v + {}^i \omega \times {}_{i+1}P) + \dot{d}_{i+1} \cdot {}^{i+1}\hat{Z} \quad (3.85)$$

$${}^1\omega = {}^1R \cdot {}^0\omega + \dot{\theta}_1 \cdot \hat{Z}_1 \quad (3.86)$$

$${}^1\omega = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \dot{\theta}_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \quad (3.87)$$

$${}^1v = {}^1R ({}^0v + {}^0\omega \times {}^0P) \quad (3.89)$$

$${}^1v = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.90)$$

$${}^2\omega = {}^2R \cdot {}^1\omega + \dot{\theta}_2 \cdot \hat{Z}_2 \quad (3.91)$$

$${}^2\omega = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \dot{\theta}_2 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.92)$$

$${}^2\omega = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \quad (3.93)$$

$${}^2v = {}^2R ({}^1v + {}^1\omega \times {}^1P) \quad (3.94)$$

$${}^2v = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \times \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \right) \quad (3.95)$$

$${}^2v = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \dot{\theta}_1 \cdot L_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 \cdot L_1 \cdot \sin \theta_2 \\ \dot{\theta}_1 \cdot L_1 \cdot \cos \theta_2 \\ 0 \end{bmatrix} \quad (3.96)$$

$${}^3\omega = {}^3R \cdot {}^2\omega + \dot{\theta}_3 \cdot \hat{Z}_3 \quad (3.97)$$

$${}^3_3\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\dot{\theta}_1 - \dot{\theta}_2 \end{bmatrix} \quad (3.98)$$

$${}^3_3\omega = -{}^2_2\omega \quad (3.99)$$

$${}^3_3v = {}^3_2R({}^2_2v + {}^2_2\omega \times {}^2_3P) + \dot{d}_3 \cdot {}^3_3\hat{Z} \quad (3.100)$$

$${}^3_3v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \left(\begin{bmatrix} \dot{\theta}_1 \cdot L_1 \cdot \sin \theta_2 \\ \dot{\theta}_1 \cdot L_1 \cdot \cos \theta_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \times \begin{bmatrix} L_2 \\ 0 \\ -d_3 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ -\dot{d}_{i+1} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.101)$$

$${}^3_3v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot \left(\begin{bmatrix} \dot{\theta}_1 \cdot L_1 \cdot \sin \theta_2 \\ \dot{\theta}_1 \cdot L_1 \cdot \cos \theta_2 + L_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} \right) \quad (3.102)$$

$${}^3_3v = \begin{bmatrix} \dot{\theta}_1 \cdot L_1 \cdot \sin \theta_2 \\ -\dot{\theta}_1 \cdot L_1 \cdot \cos \theta_2 - L_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} \quad (3.103)$$

$${}^0_3R = {}^0_1R \cdot {}^1_2R \cdot {}^2_3R \quad (3.104)$$

$${}^0_3R = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.105)$$

$${}^0_3R = \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) & 0 \\ \sin(\theta_1 + \theta_2) & -\cos(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.106)$$

$${}^0_3v = {}^0_3R \cdot {}^3_3v \quad (3.107)$$

$${}^0_3v = \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) & 0 \\ \sin(\theta_1 + \theta_2) & -\cos(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \cdot L_1 \cdot \sin \theta_2 \\ -\dot{\theta}_1 \cdot L_1 \cdot \cos \theta_2 - L_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} \quad (3.108)$$

$${}^0_3v = \begin{bmatrix} -\dot{\theta}_1 \cdot L_1 \cdot \sin \theta_1 - L_2 \cdot \sin(\theta_1 + \theta_2) \cdot (\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{\theta}_1 \cdot L_1 \cdot \cos \theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2) \cdot (\dot{\theta}_1 + \dot{\theta}_2) \end{bmatrix} \quad (3.109)$$

$$v^0 = J^0(\Theta) \cdot \dot{\Theta} \quad (3.110)$$

$$J^0(\Theta) = \begin{bmatrix} -L_1 \cdot \sin \theta_1 - L_2 \cdot \sin(\theta_1 + \theta_2) & -L_2 \cdot \sin(\theta_1 + \theta_2) \\ L_1 \cdot \cos \theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2) & L_2 \cdot \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (3.111)$$

$$\det[J^0(\Theta)] = \begin{vmatrix} -L_1 \cdot \sin \theta_1 - L_2 \cdot \sin(\theta_1 + \theta_2) & -L_2 \cdot \sin(\theta_1 + \theta_2) \\ L_1 \cdot \cos \theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2) & L_2 \cdot \cos(\theta_1 + \theta_2) \end{vmatrix} \quad (3.112)$$

$$\det[J^0(\Theta)] = L_1 \cdot L_2 \cdot \sin \theta_2 \quad (3.113)$$

$$\det[J^0(\Theta)] = 0 \Rightarrow \sin \theta_2 = 0 \Rightarrow \theta_2 = 0 \text{ ya da } \theta_2 = 180^\circ \quad (3.114)$$

Tekillik durumu ikinci eklemin sadece sifira ulařtıęında gerekleřmektedir, ünkü ikinci eklem alıřma limiti $[0, 120^\circ]$ arasındadır. Robotun alıřma alanı, robotun lümleri ve eklemlerin limitlerine göre Matlab programı kullanarak grafiksel bir řekilde ıkarılmıřtır. řekil 3.9 robot alıřma alanının Matlab izim programı gsterilmektedir ve řekil 3.10 üç eklemlı robotun alıřma alanı gstermektedir.

Robotun lümleri:

Eklemlerin uzunlukları: $L_1 = L_2 = 45\text{cm}$

Eklemlerin limitleri: $\theta_1 = [0, 180]$ $\theta_2 = [5, 120^\circ]$

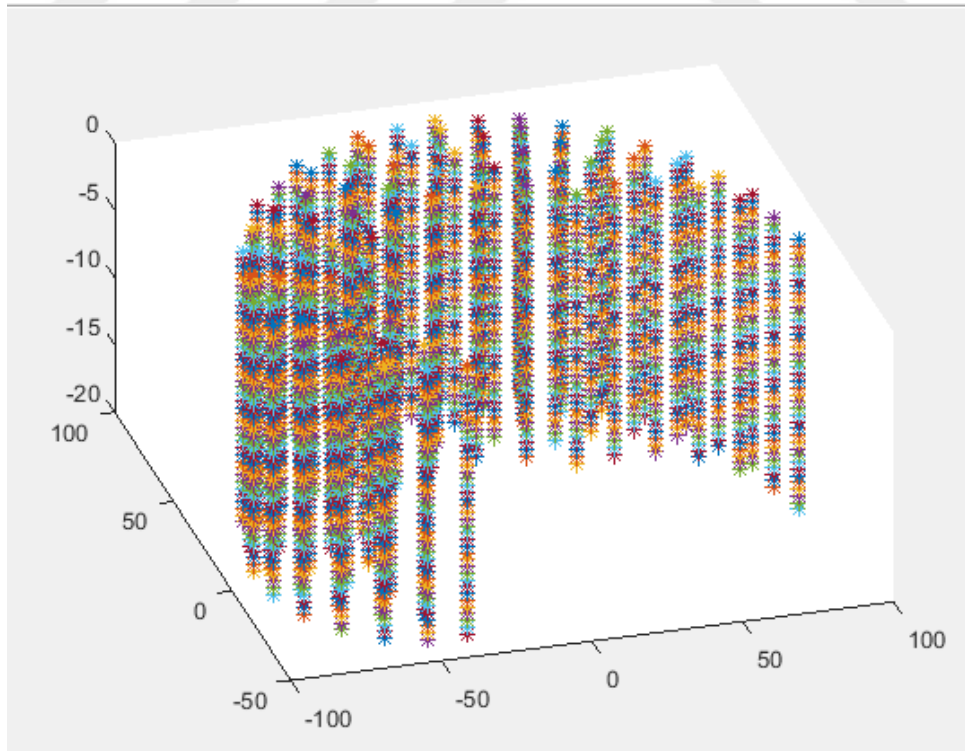
Prizmatik eklem $d_3 = [0, 20]$

```

calismauzayim x +
1  %üç eklemlı robotun çalışma uzayıdır
2  clc;
3  clear;
4
5  syms th1 th2 th3
6
7  L(1)=Link([0 0 45 0]);
8  L(2)=Link([0 0 45 pi]);
9  L(3)=Link([0 0 0 0 1]);
10 L(3).qlim = [0 20];
11 robv=SerialLink(L);
12 % açı [5,120]
13 for th2=0.09:0.3 :2.09
14     for th1=0:0.3:pi
15         for th3=0:0.5:20
16
17             e= robv.fkine([th1 th2 th3]);
18             r= e(:,4);
19
20             plot3(r(1),r(2),r(3),'*')
21             hold on
22                 end
23                 end
24         end

```

Şekil 3.9: Robot Çalışma Alanının Matlab ile Çizim Programı.



Şekil 3.10: Üç Eklemlı Robotun Çalışma Alanı

3.5 CAD Çizimleri

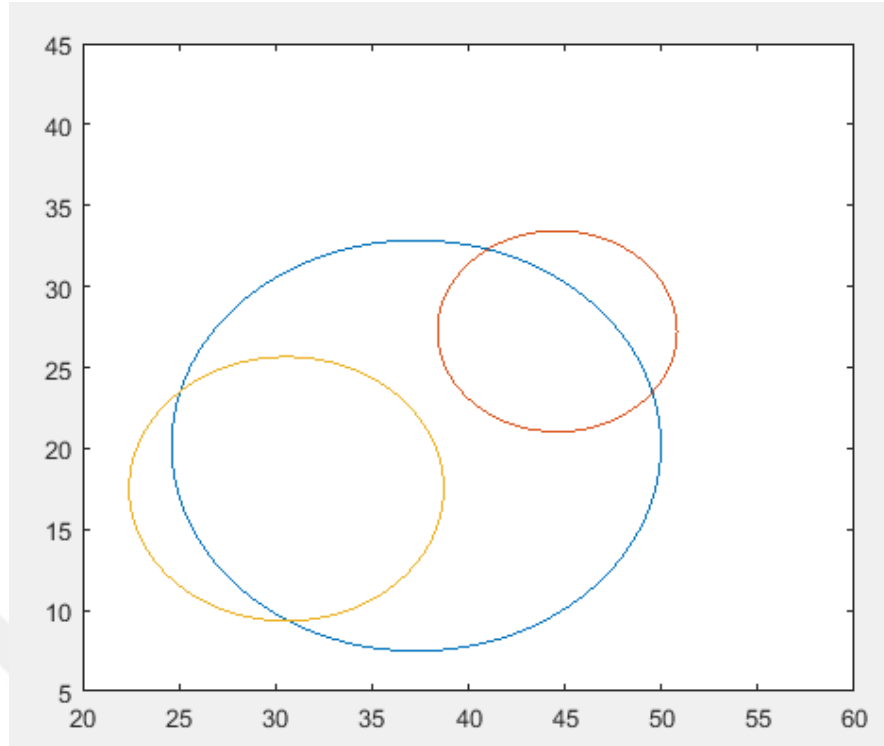
Bu bölüm kapsamında, iş parçasının kenar geometrisini AutoCAD DXF dosyasından çıkarılıp Matlab ortamına aktarılacaktır, CAD çizimi Matlaba aktarma amacı, CAD sistemi robotla CAM sistemiyle bağlamaktadır.

Genelde, AutoCAD yardımıyla tasarlanan mekanik parçaları, DXF formatı ya da STL formatı kullanarak başka bir programa aktarılmalarını ve içeriklerinin okunmasına izin vermektedir, bu çalışmada DXF formatı kullanılmıştır. DXF formatı parçanın 4 geometrik bileşene (daire, yay, doğru, elips) ayrılıp metin olarak kaydetmektedir.

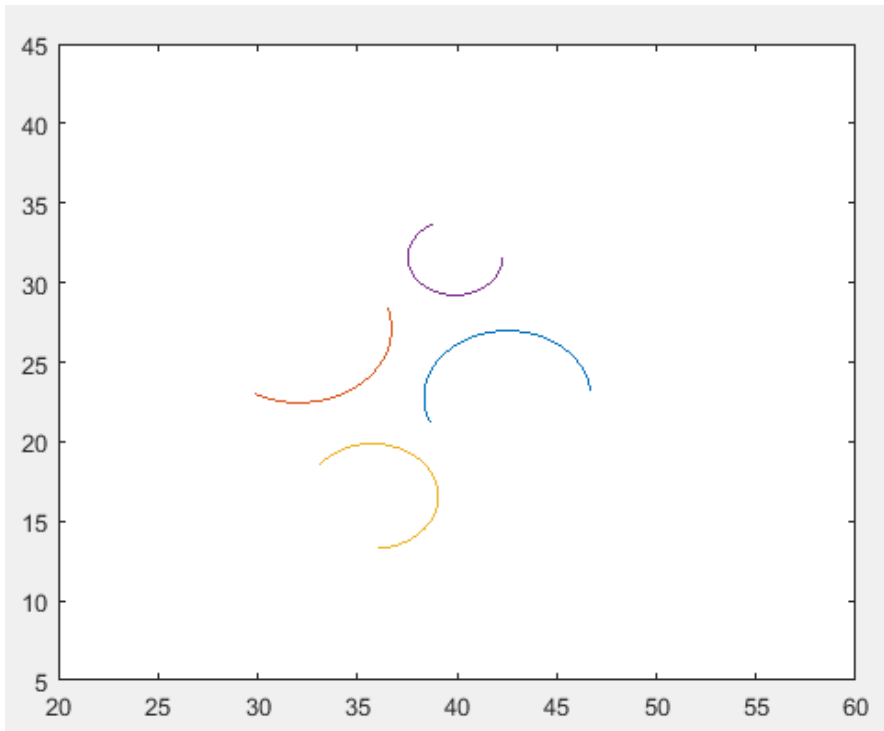
Geometrik bilgilerin bulunması ve çıkarılması için metin belli bir yer işaretleri ihtiva eder, her bileşen için özgün bir tanımlama ismi mevcuttur. Tanımlama ismi metin tarama komutasıyla bulduktan sonra, geometrik bilgilerinin metindeki yerleri tanımlama isminden standart bir mesafede oldukları, için buldukları satırların numaraları belli olur. Bu satırlar okunduktan sonra (str2double) komutuyla formatlarını rakamsal olarak değiştirilmektedir. Her bileşenin geometrik bilgileri ve koordinat merkezi elde edildikten sonra matematiksel denklemleri kullanarak, yeterli bir hassasiyet göze alındığında hesaplanan noktaları plot komutası yardımıyla çizilmektedir.

Daire ve yay çizimi: Daire ve yay DXF formatında aynı tanıma ismine sahip oldukları için text tarama komutu tarafından bulduktan sonra yazılan program yardımıyla ayrılmaktadırlar. Dairenin Matlab ortamına aktarılması için DXF metninden dairenin yarıçapı, koordinat merkezi elde edilip, dairenin denklemi kullanılarak Matlab ortamında çizilmektedir.

Yay ögesi DXF formatından okunduktan sonra daireden ayrılmaktadır. Matlab'ta çizilmesi için yayın merkezi ve yarıçapı, yayın açısız başlangıcı ve bitişi ve yayın yönü DXF programından elde edilmesi gerekmektedir. Daire ve yay çizme programı Ek Açıklama-A'da gösterilmektedir. Farklı dairelerin geri çizimi Şekil (3.11)'de ve farklı yön sahibi yaylar Şekil 3.12'de Matlab ortamında çizimleri gösterilmektedir

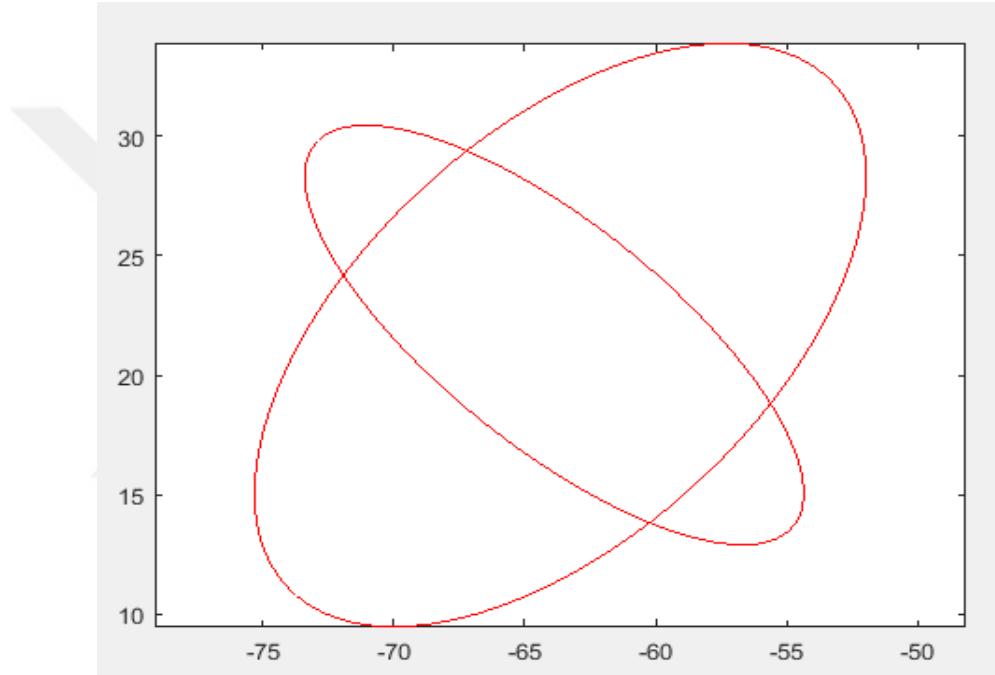


Şekil 3.11: Farklı Dairelerin Çizimi



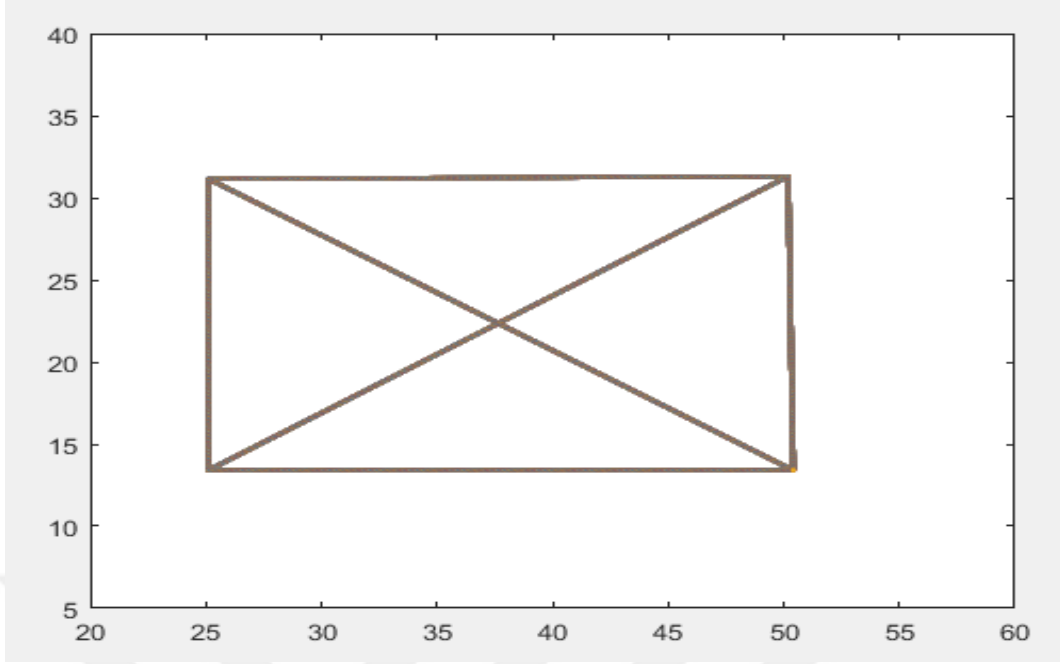
Şekil 3.12: Farklı Yön Sahibi Yayların Matlab Ortamında Çizimleri

Elips çizimi: Elips AutoCAD DXF dosyasından, metin tarama komutu tarafından tanındıktan sonra, Matlab'ta çizilmesi için geometrik bilgileri elde edilip, ardından onun denklemine dayanarak noktalarının geometrik yerlerini belirtilmektedir. Elips aktarılması için geometrik bilgileri (elipsin merkezi, asal eksenin bitiş noktası, asal ve yedek eksenin oranları) olarak AutoCAD çiziminin DXF formatı tarafından verilmektedir. Ek Açıklama-B'de elips çizme programı ve Şekil 3.13'te elipsin AutoCAD çizimi, Matlab ortamında çizilmesini göstermektedirler



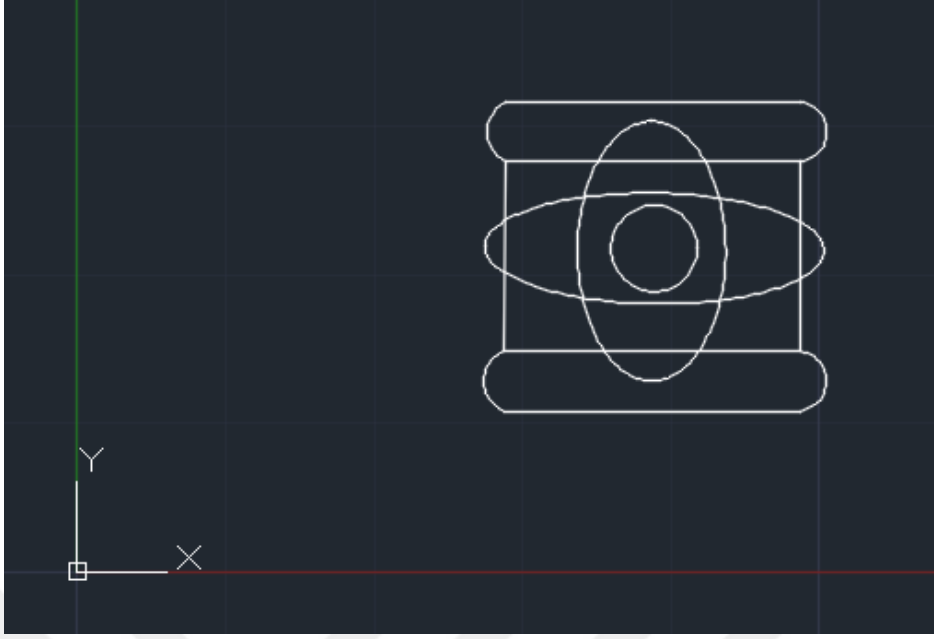
Şekil 3.13: Elipsin AutoCAD Çiziminin Matlab Ortamında Çizilmesi

Doğru çizimi: AutoCAD'ın içinde bulunan doğruların başka ortamlara aktarılabilmesi için, her doğrunun başlangıç ve bitiş noktaları DXF metninde verilmiştir. Doğrunun Matlab'ta çizilmesi amacıyla başlangıç ve bitiş noktaları arasında küçük adımlarla gidilmesi amaçlanmıştır. Adım sayısının bulunması için, resmin eksenlere izdüşümlerinin uzunluklarına başvurulmaktadır. Doğrunun düzlemde hangi eksene daha büyük izdüşümü varsa, belirlenmiş bir adım uzunluğuna göre doğrunun adım noktalarının sayısı elde edilmektedir. Şekil 3.14 AutoCAD'ta çizilmiş doğruların çıkarılması ve Matlab ortamında çizilmesi gösterilmektedir, Ek Açıklama-C ise AutoCAD ortamında çizilen çizimin, Matlab ortamında geri çizme programı göstermektedir

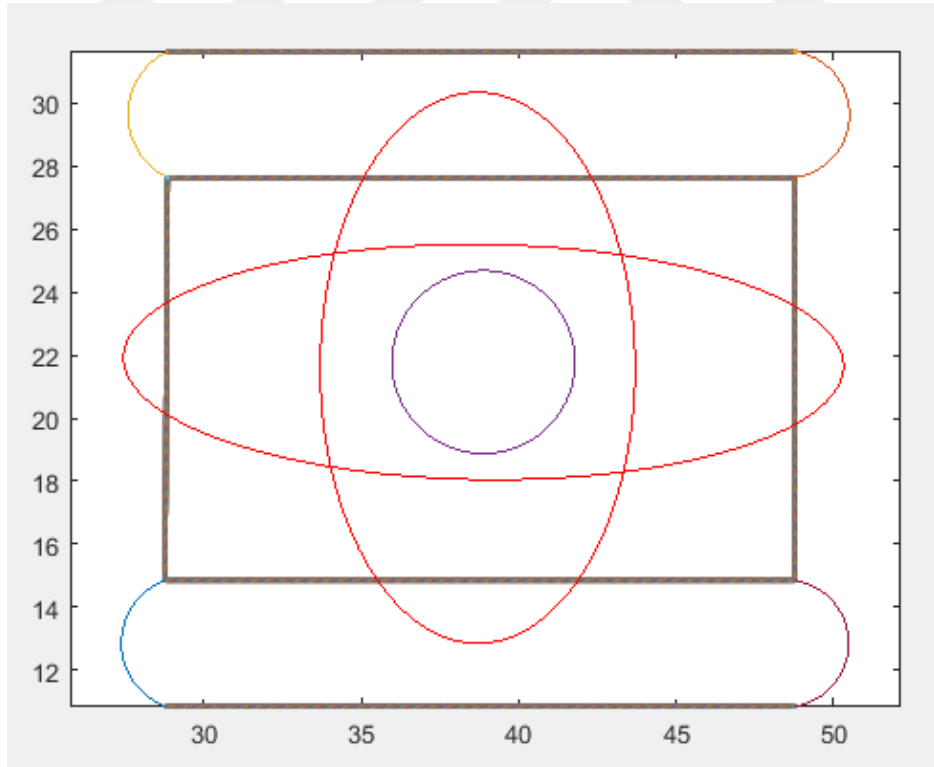


Şekil 3.14: AutoCAD'ta Çizilmiş Doğruların Matlab Ortamında Çizilmesi

AutoCAD'tan bütün resim aktarımı: Resmin öğeleri DXF formatlı dosyasından Matlab'a teker teker aktarabildikten sonra, geliştirilen yazılımlar tek bir programda toplanmıştır. Yay, elips, doğru ve daire içeren AutoCAD resmi, Matlab programı tarafından okunup geri çizilmiştir. (Şekil 3.16) 4 öge resmin AutoCAD ortamında çizimi ve (Şekil 3.15) Matlab ortamında 4 öge resmin geri çizilmesi göstermektedirler. Geliştirilen yazılım Ek Açıklama-D içinde gösterilmektedir.



Şekil 3.15: 4 Eleman İçeren Resmin AutoCAD Ortamında Çizimi



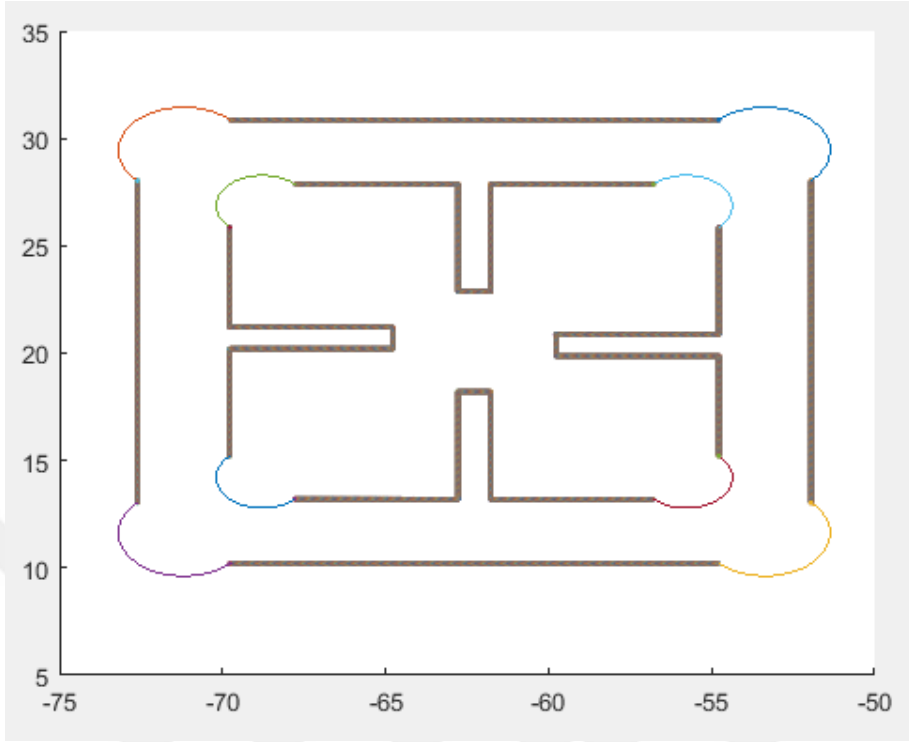
Şekil 3.16: 4 Eleman İçeren Resmin Matlab Ortamında geri Çizilmesi

4. BULGULAR VE TARTIŞMA

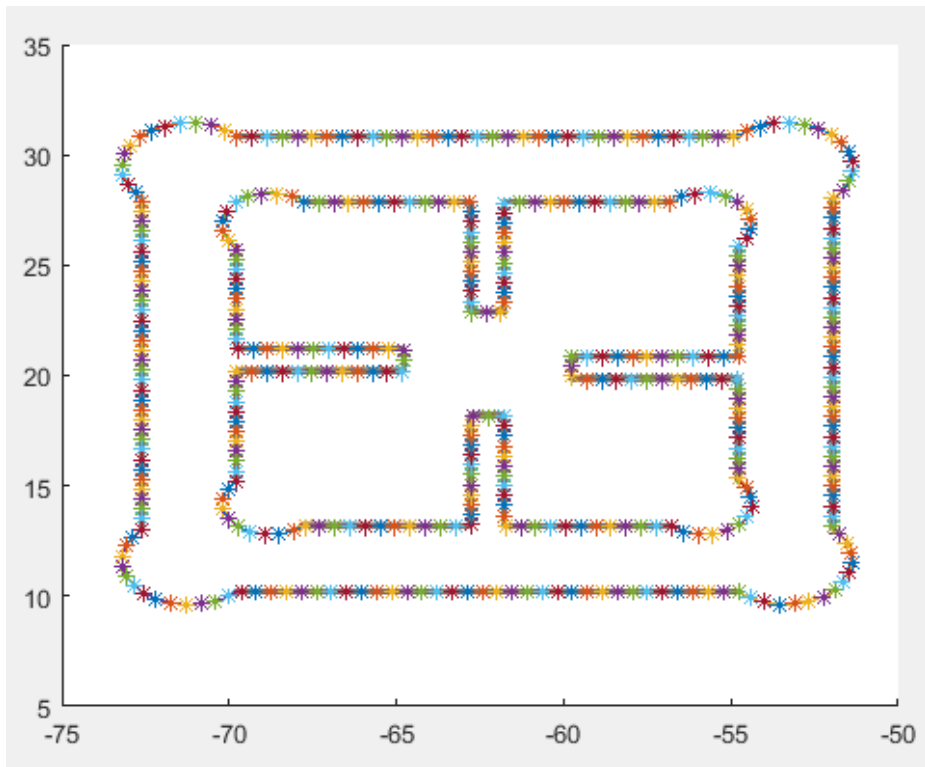
AutoCAD ile çizilen 2 boyutlu iş parçasını temsil eden resmin Matlab ortamına aktarıldıktan sonra, yol planı yapılır. Yol planı döngüler olarak tasarlanmıştır. Bu çalışmada döngüler yaylar ve çizgilerden oluşmaktadır. Döngüleri oluşturabilmek için, yaylar ve çizgilerin başlangıç ve bitiş noktalarına dayanarak, bitişik olanlar aralarındaki mesafeden hangi öğeye ait oldukları tespit edilmekte ve öğeler sıralanarak döngüler oluşturulmaktadır. Parçanın kenarlarını gezebilmek için, öğelerin etiket noktaları, döngülerin bulunma sırasına göre dizilerek, yol planı bir Matlab hücresi içinde toplanır. Hücrelerin her elemanı bir döngünün etiket noktalarını ihtiva eder. Etiket noktalarını laboratuvardaki robota besleyebilmesi için, ters kinematik kullanarak, birinci ve ikinci dönел robot eklemlerinin açıları hesaplanmaktadır.

DXF dosyasında, çizgilerin başlangıç ve bitiş noktaları belirtilmektedirler. Yayların başlangıç ve bitiş noktalarını belirtmek için, çizgilerin başlangıcına bitişik olan yayın tarafları yayın bitişleri olarak tanıtılmışlardır. Yazılan programın çalışabilmesi için iş parçasının AutoCAD çizimi aşamasında bazı noktaların dikkate alınması gerekmektedir. Döngü çizme esnasında çizgilerin döngü boyunca tek bir yönle çizilmesi, en az bir yayı ihtiva etmesi ve iki yayın bitişik olmaması gerekmektedir, bu çizme şartları program geliştirdikçe ortadan kaldırılacaktır.

Yayın başlangıç ve bitiş noktalarının bulunması için yazılan program Ek Açıklama-E içinde ve döngülerin birbirinden ayrılması için yazılan program Ek Açıklama-F içinde bulunmaktadır. Çıkarılan döngülerin etiket noktalarını bir hücre unsurunda toplaması için yazılan program Ek Açıklama-G içinde ve planlanan yolun simülasyon ile kontrol etmek için yazılan program Ek Açıklama-H içinde bulunmaktadır. Planlanan yol boyunca etiket noktalarının ters kinematik kullanarak robotun açıları bulması için yazılan program Ek Açıklama-I içinde ve belli bir linspace limiti için açıların çizilmesi için yazılan program Ek Açıklama-J içinde bulunmaktadır. Program denemesi için, iş parçası AutoCAD'ın DXF dosyasından okunup geri çizilmesi Şekil 4.1, Yol planının etiket noktalarını sırayla çizildikten sonra Şekil 4.2'de gösterilmektedir.



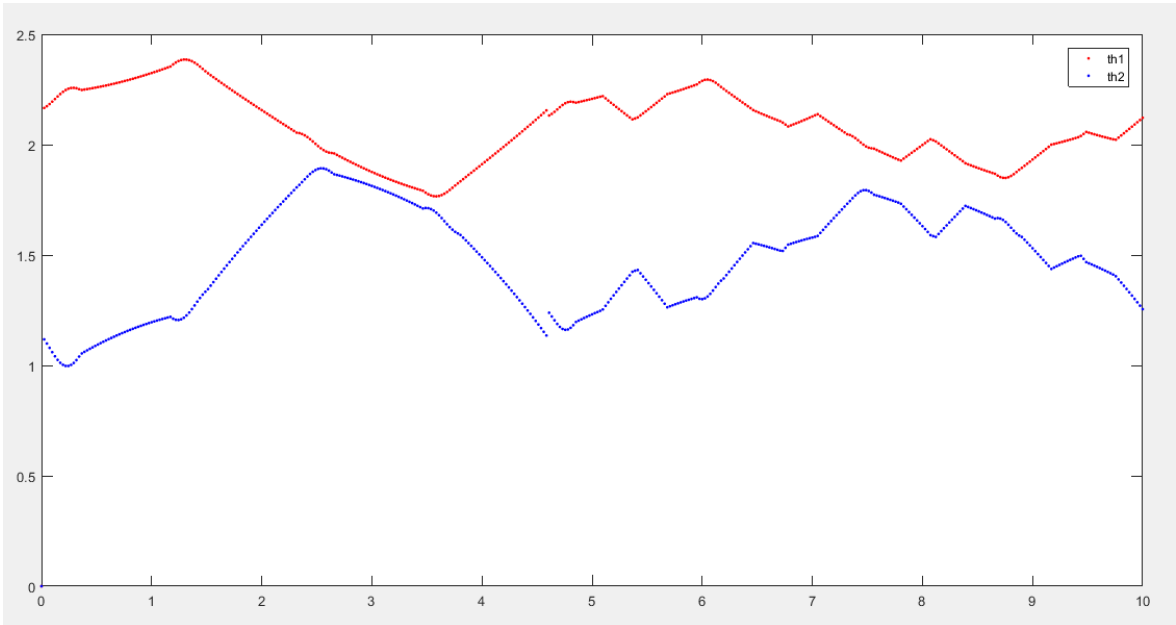
Şekil 4.1: İş Parçasının Matlab Ortamında geri Çizilmesi



Şekil 4.2: Parça Üstündeki Etiket Noktaları

5. SONUÇ VE ÖNERİLER

Çizimden etiket noktaları çıkardıktan sonra SCARA robotun açıları çizilmektedir, linspace boyunca açıların değişimi Şekil 4.3'te gösterilmektedir. Çalışmanın sonucu, çapak alma robotların yörüngesi otomatik üretilmesi ele alınıp, çapak alma robotları için DXF dosyasından yörünge üretilebilmiştir. Ancak çapak alma robotunun sanayi ortamında çalışabilmesi için robot kinematik hataları, parçanın sabitleme ve tolerans hatası ve başka kaynaklardan çıkan hataların telafi edilmesi gerekmektedir. Çapak alma probleminin robotla yapılması gerçekleştiği durumda, zamanın ve birçok beşeri ve sanayi kaynakların tasarruf edilmesi sağlanacaktır. Gelecek çalışmalarda; hataların telafisi, robotun kontrolü, çapakların yerlerine ve özelliklerine göre 3 boyutlu iş parçalarında yörünge planlaması, takımın aşınması ve robotla çapak alma sürecini tamamlayan görevlerin araştırılması ve uygulanması planlanmaktadır.



Şekil 5.1: Linspace boyunca Açların Değişmesi

KAYNAKLAR DİZİNİ

- Aydın, H., 5 Eksenli Çapak Alma Robotunun Kontrolü, Eskişehir Osmangazi Üniversitesi.
- Corke, P., 2017, Robotics Toolbox, <https://petercorke.com/toolboxes/robotics-toolbox/>, Erişim: 12:8:2021.
- Çakır, H., 2019, Genel Amaçlı Hareket Kontrol Kartıyla 4 Eksenli Robotun Kontrolü, Eskişehir Osmangazi Üniversitesi.
- Ekrem, A., 2021, Çeliklerin Talaşlı İmalat Prosesinde İşlenebilirlik Performansı ile İşletme Maliyeti Optimizasyonu, Uşak Üniversitesi.
- Gillespie, L., 1980, Precision Deburring Using NC and Robot Equipment, United States Department of Energy.
- Gillespie, L., 1980, Deburring: Technical Capabilities and Cost-Effective Approaches Lessons 5 and 6, United States Department of Energy.
- Greene, S., 1984, Path Control for Robotic Deburring, Massachusetts Institute of Technology.
- Jayaweera, N., Phil, W., 2010, Measurement Assisted Robotic Edge Deburring of Aero engine Components, WSEAS Transactions on Systems and Control, 5, 3, p.1-10, <https://www.wseas.us/e-library/conferences/2010/Cambridge/ISPRA/ISPRA-22.pdf>, erişim tarihi: 30.12.2021.
- Johannesen, R., Gillespie, L., McMillen, J., 1986, Hand deburring: A Necessity That can Be Improved, agency of united States Government.
- Mizukawa, M., Mitsuya, E., Ohara, S., Iwaki, S., Matsuo, S., Shakunaga, T., Okada T., 1994, Robotic Task and System Design for Task Directed Robotic Case Study: Deburring Task, IEEE, 949-956.
- Mohammad, M., Babriya, V., Sobh, T., 2008, Modeling a Deburring Process Using DELMIA V5, Journal of Engineering and Applied Sciences, 835-847.
- Murphy, K., Norcross, R., Proctor, F., 1998, CAD Directed Robotic Deburring, ResearchGate, 1-9.
- Onstein, I., Semeniuta, O., Bjerkg, M., 2020, Deburring Using Robot Manipulators: A Review, IEEE, 1-7.
- Pullkinen, T., Heikkila, T., Sallinen, M., Kivikunnas, S., Salmi, T., 2008, 2 CAD Based Robot Programming for Processing Metal Profiles in short Series Manufacturing, International Conference on Control, Automation and Systems, 156-162.

KAYNAKLAR DİZİNİ (devam)

- Ryuh, B., Pennock, G., 2006, Robot Automation Systems for Deburring, ResearchGate, 609-622.
- Schmidt, J., Grandi, F., Peruzzini M., Raffaelli, R., pelliccirari, M., 2021, Novel Robotic Cell Architecture for Zero Defect Intelligent Deburring, ScienceDirect, 140-147.
- Stouffer, K., Russell R., 1995, ADACS: An Advanced Deburring and Chamfering System, National Institute of Standards and Technology, 1-6.
- Thomas, C., Katsuhiro, M., Toshiyuki, O., Yasuo Y., 2000, Metal Machining Theory and Application, Aronld a member of the Hodder Headline Group, p.8-10.
- Yang, X., Lou, Y., Li, Z., Chen, H., 2014, Survey of Robotic Manipulator Workspace, IEEE, 8530-8538.
- Yasin, E., 2009, Sanayinde Talaşlı Üretimde Kullanılacak Web Esaslı Veri Tabanın Geliştirmesi, Selçuk Üniversitesi.

EK AÇIKLAMALAR

Ek Açıklama-A: Daire ve Yay Çizme Programı

11.12.2021 17:28 E:\matlab2...\yayvedaireprojeprogrami.m 1 of 2

```
%yay ve daire çizme programı
fileID = fopen('yayve45.txt');
values = textscan(fileID,'%s %f');
fclose(fileID);
groupcode=values{1};

dairelersayi=0;
yaylarsayi=0;

indexcircle= strmatch('AcDbCircle',groupcode,'exact');
sayidaire=length(indexcircle);

for looptekrari=1:sayidaire
    cizilandaire=indexcircle(looptekrari);
    daireveyayay=groupcode(cizilandaire+10);

    h=char(daireveyayay);
    h1='AcDbArc';
    h2=strcmp(h,h1);
    h2;
    if h2
        yaylarsayi=[yaylarsayi;looptekrari];

        xmerkez= groupcode(cizilandaire+2);
        ymerkez= groupcode(cizilandaire+4);
        circleradius= groupcode(cizilandaire+8);
        basaci=groupcode(cizilandaire+12);
        bitis=groupcode(cizilandaire+14);

        basradian=(str2double(basaci)*pi)/180;
        bitisradian=(str2double(bitis)*pi)/180;

        if basradian>bitisradian
            b=basradian:0.01:6.2832;
            bb= 0:0.1:bitisradian;
            a=[b bb];
        else
            a=basradian:0.01:bitisradian;
        end
        radius=str2double(circleradius);

        xm=str2double(xmerkez);
        ym=str2double(ymerkez);

        x1=xm +radius*cos(a);
        y1=ym +radius*sin(a);

        plot(x1,y1)
        hold on;

    else
        dairelersayi=[dairelersayi;looptekrari];
        cizilandaire=indexcircle(looptekrari);
    end
end
```

11.12.2021 17:28 E:\matlab2...\yayvedaireprojeprogrami.m 2 of 2

```
xmerkez= groupcode(cizilandaire+2);
ymerkez= groupcode(cizilandaire+4);
circleradius= groupcode(cizilandaire+8);

radius=str2double(circleradius);
a=0:0.01:2*pi;

xmerkeze=str2double(xmerkez);
ymerkezke=str2double(ymerkez);
xg= xmerkeze +radius*cos(a);
yg= ymerkezke +radius*sin(a);

plot(xg,yg)

    hold on;

    end
end

xlim([20 60])
ylim([5 45])
```



Ek Açıklama-B: Elips Çizme Programı

12.12.2021 10:04 E:\matlab2016be...\elipsprojeprogrami.m 1 of 1

```
%elips çizme programı

fileID = fopen('elips45.txt');
values = textscan(fileID,'%s %f');
fclose(fileID);
groupcode=values{1};

noktalare=[0 0];
ellipslersayi=0

indexellips= strmatch('AcDbEllipse',groupcode,'exact');
sayiellipse=length(indexellips);

for looptekrarie=1:sayiellipse
    ellipslersayi=[ellipslersayi;looptekrarie];
    cizilanellipse=indexellips(looptekrarie);
xmerkez= groupcode(cizilanellipse+2) ;
ymerkez= groupcode(cizilanellipse+4) ;

majoraxisx=groupcode(cizilanellipse+8);
majoraxisy=groupcode(cizilanellipse+10);
xs=str2double(majoraxisx);
ys=str2double(majoraxisy);
majoraxisnorm=[xs ys];
majoraxis= norm(majoraxisnorm);
manoraxis=groupcode(cizilanellipse+20);
acibaslangics=groupcode(cizilanellipse+22);
acibitiss=groupcode(cizilanellipse+24);

acibaslangic=str2double(acibaslangics);
acibitis=str2double(acibitiss);
t = acibaslangic:0.01: acibitis ;
a=majoraxis;
b = a*str2double(manoraxis);
xel = a*cos(t);
yel = b*sin(t);

theta=atan2(ys,xs);

R = [cos(theta), -sin(theta);
     sin(theta), cos(theta)];
xy =R*[xel ; yel];

xelipsmerkez=str2double(xmerkez);
yelipsmerkez=str2double(ymerkez);
x1=xy(1,:)+xelipsmerkez;
y1=xy(2,:) +yelipsmerkez;
plot(x1,y1,'r')
hold on

axis equal
end
```

Ek Açıklama-C: Çizginin Matlab Ortamında Çizme Programı

12.12.2021 18:32 E:\matlab2016be...\dogrucizmeprogrami.m 1 of 2

```
% doğru çizme programı

fileID = fopen('dogruve45.txt');
values = textscan(fileID,'%s %f');
fclose(fileID);
groupcode=values{1};

adim=0.01;

cizgilersayi=0;

baslagicxler=0;
baslagicyeler=0;
bitisxler=0;
bitisyler=0;

indexLine= strmatch('AcDbLine',groupcode,'exact');
sayiciz=length(indexLine);

for looptekrari=1:sayiciz
    cizgilersayi=[cizgilersayi;looptekrari];
    cizilanline=indexLine(looptekrari);

    xbaslangic= groupcode(cizilanline+2);
    ybaslangic= groupcode(cizilanline+4);
    xbaslangicn=str2double(xbaslangic);
    ybaslangicn=str2double(ybaslangic);

    xbitis= groupcode(cizilanline+8);
    ybitis= groupcode(cizilanline+10);
    xbitisn=str2double(xbitis);
    ybitisn=str2double(ybitis);

    baslagicxler=[baslagicxler;xbaslangicn];
    baslagicyeler=[baslagicyeler;ybaslangicn];
    bitisxler=[bitisxler;xbitisn];
    bitisyler=[bitisyler; ybitisn];

    xtemesafe=abs(xbitisn- xbaslangicn);
    ytemesafe=abs(ybitisn- ybaslangicn);

    xaron=sign(xbitisn- xbaslangicn);
    yaron=sign(ybitisn- ybaslangicn);

    xsayi=xtemesafe/adim;
    ysayi=ytemesafe/adim;

    yoln=0;
    if xsayi>ysayi
        xadim=adim;
        dongetekrari=xtemesafe/xadim;
        yadim=ytemesafe/dongetekrari;
    elseif xsayi<ysayi
        yadim=adim;
        dongetekrari=ytemesafe/yadim;
```

12.12.2021 18:32 E:\matlab2016be...\dogrucizmeprogrami.m 2 of 2

```
xadim=xtemesafe/dongetekrari;
else
    yadim=adim;
    xadim=adim;
    dongetekrari=xtemesafe/xadim;
end

tgx=xbaslangicn;
tgy=ybaslangicn;
r=[tgx tgy];
for vv=1:1:dongetekrari
    vv;
    r=[tgx tgy];

    tgx=tgx+(xadim*xaryon);
    tgy=tgy+(yadim*yaryon);
    tgx1=tgx+(xadim*xaryon);
    plot(tgx1,tgy,'.')
    hold on
    yoln=yoln+1;

end

end

xlim([20,60])
ylim([5,40])
```



Ek Açıklama-D: Matlab Ortamına Çizim Aktarma Programı

09.01.2022 14:18 E:\matlab2...\autocadtanokumaprogrami.m 1 of 6

```
%AutoCAD'tan okuma programi(1)
clear;
clc;
fileID = fopen('gercekocakg.txt');
values = textscan(fileID,'%s %f');
groupcode=values{1};

adim=0.01;
kaydirmax=100;
takiminyaricapi=0.45;

tgizleme=[0 0];
nktalarc=[0 0];
nktalard=[0 0];
nktalary=[0 0];
nktalare=[0 0];

cizgilersayi=0;
dairelersayi=0;
yaylarsayi=0;
ellipslersayi=0;

baslagiccizxlerk=0;
baslagiccizylerk=0;
bitiscizxlerk=0;
bitiscizylerk=0;

baslagicyayxler=0;
baslagicyayyler=0;
bitisyayxler=0;
bitisyayyler=0;

indexLine= strmatch('AcDbLine',groupcode,'exact');
sayiciz=length(indexLine);

indexcircle= strmatch('AcDbCircle',groupcode,'exact');
sayidaire=length(indexcircle);

indexellips= strmatch('AcDbEllipse',groupcode,'exact');
sayiellipse=length(indexellips);

for looptekrari=1:sayiciz
    cizgilersayi=[cizgilersayi;looptekrari];
    cizilanline=indexLine(looptekrari);

    xbaslangic= groupcode(cizilanline+2);
    ybaslangic= groupcode(cizilanline+4);
    xbaslangicn=str2double(xbaslangic);
    ybaslangicn=str2double(ybaslangic);

    xbitis= groupcode(cizilanline+8);
    ybitis= groupcode(cizilanline+10);
    xbitisn=str2double(xbitis);
    ybitisn=str2double(ybitis);
```

09.01.2022 14:18 E:\matlab2...\autocadtanokumaprogrami.m 2 of 6

```

xbaslangicnnd=xbaslangicn-kaydirmax;
xbitlangicnnd=xbitisn-kaydirmax;
%plot(xbaslangicnnd,ybaslangicn, '*')

baslagiccizxlerk=[baslagiccizxlerk;xbaslangicnnd];>>>>
baslagiccizylerk=[baslagiccizylerk;ybaslangicn];

bitiscizxlerk=[bitiscizxlerk;xbitlangicnnd];
bitiscizylerk=[bitiscizylerk;ybitisn];
% plot(xbitlangicnnd,ybitisn, '*')
hold on

xtemesafe=abs(xbitisn- xbaslangicn);
ytemesafe=abs(ybitisn- ybaslangicn);

xaryon=sign(xbitisn- xbaslangicn);
yaryon=sign(ybitisn- ybaslangicn);

    xsayi=xtemesafe/adim;
    ysayi=ytemesafe/adim;

    yoln=0;
    if xsayi>ysayi
        xadim=adim;
        dongetekrari=xtemesafe/xadim;
        yadim=ytemesafe/dongetekrari;
    elseif xsayi<ysayi
        yadim=adim;
        dongetekrari=ytemesafe/yadim;
        xadim=xtemesafe/dongetekrari;
    else
        yadim=adim;
        xadim=adim;
        dongetekrari=xtemesafe/xadim;
    end

    tgx=xbaslangicn;
    tgy=ybaslangicn;
    r=[tgx tgy];
    noktalarc=[0 0];
    for vv=1:1:dongetekrari
        r=[tgx tgy];
    tgizleme=[tgizleme;r];
    tgx=tx+(xadim*xaryon);
    tgy=ty+(yadim*yaryon);
    tgx1=tx+(xadim*xaryon)-kaydirmax;
        plot(tgx1,tgy, '.')
        hold on
        yoln=yoln+1;
        takiminyaricapid=takiminyaricapid*100;
        if yoln==takiminyaricapid
            gc=[tgx1 tgy];
        noktalarc=[noktalarc;gc];

        yoln=0;

```

09.01.2022 14:18 E:\matlab2...\autocadtanokumaprogrami.m 3 of 6

```
end

end

noktalarcuzun=length(noktalarc);
noktalarcx=noktalarc(2:noktalarcuzun,1);
noktalarcy=noktalarc(2:noktalarcuzun,2);
noktalarcdes=[noktalarcx noktalarcy];
noktalarcflip=flip(noktalarcdes);
A{looptekrari}=noktalarcflip;
end
basbitiscizuzu=length(baslagiccizxlerk);

baslagiccizxler=baslagiccizxlerk(2:basbitiscizuzu);
baslagiccizyler=baslagiccizylerk(2:basbitiscizuzu);
bitiscizxler=bitiscizxlerk(2:basbitiscizuzu);
bitiscizyler=bitiscizylerk(2:basbitiscizuzu);

bitiscizler=[bitiscizxler bitiscizyler];

for looptekrari=1:sayidaire
    cizilandaire=indexcircle(looptekrari);
    daireveyayay=groupcode(cizilandaire+10);

    h=char(daireveyayay);
    h1='AcDbArc';
    h2=strcmp(h,h1);
    h2;
    if h2
        yaylarsayi=[yaylarsayi;looptekrari];

        xmerkez= groupcode(cizilandaire+2);
        ymerkez= groupcode(cizilandaire+4);
        circleradius= groupcode(cizilandaire+8);
        basaci=groupcode(cizilandaire+12);
        bitis=groupcode(cizilandaire+14);

        basradian=(str2double(basaci)*pi)/180;
        bitisradian=(str2double(bitis)*pi)/180;

        if basradian>bitisradian
            b=basradian:0.01:6.2832;
            bb= 0:0.01:bitisradian;
            a=[b bb];

        else
            a=basradian:0.01:bitisradian;
        end
        radius=str2double(circleradius);
        xtekayma=100;
        xm=str2double(xmerkez)-xtekayma;
        ym=str2double(ymerkez);
```

09.01.2022 14:18 E:\matlab2...\autocadtanokumaprogrami.m 4 of 6

```

x1=xm +radius*cos(a);
y1=ym +radius*sin(a);

baslagicyayxlern=xm +radius*cos(a(1));
baslagicyayylern=ym +radius*sin(a(1));
%plot(baslagicyayxlern,baslagicyayylern,'*')
baslagicyayxler=[baslagicyayxler;baslagicyayxlern];
baslagicyayyler=[baslagicyayyler;baslagicyayylern];

abitissayi=length(a);
bitisyayxlern=xm +radius*cos(a(abitissayi));
bitisyayylern=ym +radius*sin(a(abitissayi));
%plot(bitisyayxlern,bitisyayylern,'*')
bitisyayxler=[bitisyayxler;bitisyayxlern];
bitisyayyler=[bitisyayyler;bitisyayylern];

plot(x1,y1)
hold on;
ld=length(a);
dadimyas=(takiminyaricapi/radius);%%
dadimyaac=asin(dadimyas);
dado=dadimyaac/0.01;
dad=round(dado);
noktalary=[0 0];
for sd=1:dad:ld

    xd=x1(sd);
    yd=y1(sd);
    gy=[xd yd];
    noktalary=[noktalary;gy];
    nly=length(noktalary);
end
noktalaryuzun=length(noktalary);
noktalaryx=noktalary(2:noktalaryuzun,1);
noktalaryy=noktalary(2:noktalaryuzun,2);
noktalarydes=[noktalaryx noktalaryy];
B{looptekrari}= noktalarydes;

else
    dairelersayi=[dairelersayi;looptekrari];
    cizilandaire=indexcircle(looptekrari);

    xmerkez= groupcode(cizilandaire+2);
    ymerkez= groupcode(cizilandaire+4);
    circleradius= groupcode(cizilandaire+8);

    radius=str2double(circleradius);
    a=0:0.01:2*pi;

    xmerkeze=str2double(xmerkez)-kaydirmax;
    ymerkezke=str2double(ymerkez);
    xg= xmerkeze +radius*cos(a);
    yg= ymerkezke +radius*sin(a);

```

09.01.2022 14:18 E:\matlab2...\autocadtanokumaprogrami.m 5 of 6

```

plot(xg,yg)

    hold on;
    ld=length(a);
    dadimyas=(takiminyaricapi/radius)%%
dadimyaac=asin(dadimyas);
dado=dadimyaac/0.01;
dad=round(dado);
for sd=1:dad:ld

    xd=xg(sd);
    yd=yg(sd);
    gd=[xd yd];
    noktalar=[noktalar;gd];

end
C{looptekrari}=noktalar;
end
end
babitayuzu=length(bitisyayxler);
bitisyayxlerdes=bitisyayxler(2:babitayuzu);
bitisyayylerdes=bitisyayyler(2:babitayuzu);
baslagicayxlerdes=baslagicayxler(2:babitayuzu);
baslagicayylerdes=baslagicayyler(2:babitayuzu);

bityayxvey=[bitisyayxlerdes bitisyayylerdes];
basyayxvey=[baslagicayxlerdes baslagicayylerdes];
YBB{1}=basyayxvey;
YBB{2}=bityayxvey;

for looptekrarie=1:sayiellipse
    ellipslersayi=[ellipslersayi;looptekrarie];
    cizilanellipse=indexellips(looptekrarie);
xmerkez= groupcode(cizilanellipse+2) ;
ymerkez= groupcode(cizilanellipse+4) ;

majoraxisx=groupcode(cizilanellipse+8);
majoraxisy=groupcode(cizilanellipse+10);
xs=str2double(majoraxisx);
ys=str2double(majoraxisy);
majoraxisnorm=[xs ys];
majoraxis= norm(majoraxisnorm);
manoraxis=groupcode(cizilanellipse+20);
acibaslangics=groupcode(cizilanellipse+22);
acibitiss=groupcode(cizilanellipse+24);

acibaslangic=str2double(acibaslangics);
acibitiss=str2double(acibitiss);
t = acibaslangic:0.01: acibitiss ;
a=majoraxis;
b = a*str2double(manoraxis);
xel = a*cos(t);

```

09.01.2022 14:18 E:\matlab2...\autocadtanokumaprogrami.m 6 of 6

```
yel = b*sin(t);

theta=atan2(ys,xs);

R = [cos(theta), -sin(theta);
     sin(theta), cos(theta)];
xy =R*[xel ; yel];

xelipsmerkez=str2double(xmerkez)-kaydirmax;
yelipsmerkez=str2double(ymerkez);
x1=xy(1,:)+xelipsmerkez;
y1=xy(2,:) +yelipsmerkez;
plot(x1,y1,'r')
hold on
ld=length(t);
dadimyas=(takiminyaricapi/a)%%

dadimyaac=asin(dadimyas);
dado=dadimyaac/0.01;
dad=round(dado);
for sd=1:dad:ld

    xd=x1(sd);
    yd=y1(sd);

    ge=[xd yd];
    noktalare=[noktalare;ge];

end
D{looptekrarie}= noktalare;
hold on
axis equal
end
```

Ek Açıklama-E: Yayın Başlangıç ve Bitişleri Bulma Programı

09.01.2022 16:00 E:\matlab2016benim\bin\yaybasprogrami.m 1 of 2

```
%yayın bslangicleri bulma programi
%(2) olarak calistiriyoruz
yeniyyaybas=[0 0];
yeniyyaybit=[0 0];
yeniyyaybittop=[0 0];
yayinsira=0;

basbitsay=length(YBB);
for basbitsa=1:1:basbitsay %yayın baslangicle ve bitislerle bitisler karis-
    babitince=YBB(basbitsa);
    babitinceuzun=length(babitince);
    for babitgez=1:1:babitinceuzun %yay baslangic ve bitisleri gezme
        babitkaris=babitince(babitgez,:);
        bitisuzunlugu=length(bitiscizler);
    for bitlerintara=1:1:bitisuzunlugu % cizgi bitis gezme

        bitiskaris=bitiscizler(bitlerintara,:);
        cykarsilasrirma=[bitiskaris;babitkaris];
        kakatsayi=pdfdist(cykarsilasrirma);
        kakatsayira=round(kakatsayi);

        if kakatsayira==0
            yeniyyaybas=[yeniyyaybas;babitkaris];

            basbitsa;
            bitlerintara;
            yayinsira=yayinsira+1;
            if basbitsa==1

                yeniyyaybit=bityayxvey(babitgez,:);
                yeniyyaybittop=[yeniyyaybittop;yeniyyaybit];
                YB{yayinsira}=B(babitgez);

            end
            if basbitsa==2
                yeniyyaybit2=basyayxvey(babitgez,:);
                yeniyyaybittop=[yeniyyaybittop;yeniyyaybit2];
                YB{yayinsira}=B(babitgez);
            end

        end

        bitlerintara=bitlerintara+1;
    end
    babitgez=babitgez+1;
    end
    basbitsa=basbitsa+1;
end

yeniyyaybasxdo=yeniyyaybas(:,1);
yeniyyaybasydo=yeniyyaybas(:,2);
yeniyyaybittopxdo=yeniyyaybittop(:,1);
yeniyyaybittopydo=yeniyyaybittop(:,2);
```

09.01.2022 16:00 E:\matlab2016benim\bin\yaybasprogrami.m 2 of 2

```
yeniyaybasxuzun=length(yeniyaybasxdo);
yeniyaybasyuzun=length(yeniyaybasydo);
yeniyaybittopxuzun=length(yeniyaybittopxdo);
yeniyaybittopyuzn=length(yeniyaybittopydo);

yeniyaybasx=yeniyaybasxdo(2:yeniyaybasxuzun);
yeniyaybasy=yeniyaybasydo(2:yeniyaybasyuzun);
yeniyaybittopx=yeniyaybittopxdo(2:yeniyaybittopxuzun);
yeniyaybittopy=yeniyaybittopydo(2:yeniyaybittopyuzn);
```

Ek Açıklama-F: Döngülerin birbirinden Ayrılması için Yazılan Program

09.01.2022 16:10 E:\matlab2016benim...\farkliyollarson.m 1 of 2

```
% farklı döngülere ayırma programı
% (3) olarak çalıştırıyoruz>>> ilk önce birinciden parametreler geliyor
% ve ikinciden yay başlanıc ve bitisleri dahil olur

baslagicxler=[baslagiccizxler;yeniyaybasx];
baslagicyeler=[baslagiccizyeler;yeniyaybasy];
bitisxler=[bitiscizxler;yeniyaybittopx];
bitisyler=[bitiscizyeler;yeniyaybittopy];

cbaslari=[baslagicxler baslagicyeler];
cbitlari=[bitisxler bitisyler];
lengu=length(cbaslari);
ciznoktasa=length(cbitlari);%
taramasira=1;%

    masurpar1=1;
    masurpar2=1;
    bititarama=2;
    taramaisareti=2;
    tabit=1;
    iza=1;
    while tabit
        masurpar2=1;%bir baslangic bulmak için başlama isareti
        masurpar1=1;%bir yol bulmak için başlama isareti
        while masurpar2 %baslangic bulmak ve program bitirmek
            m12=1;

            refbax=baslagicxler(taramasira);
            refbay=baslagicyeler(taramasira);
            refba=[refbax refbay];
            notaiz=cbitlari(taramasira,1)%sifirlari aramak için
            if notaiz~=0
                masurpar2=0; %bir baslangic buldugun zaman cik
            end
            if taramasira==ciznoktasa %eger butun bitisler sifir ise
                masurpar2=0 ;%baslangic aramasi bitir
                tabit=0 ; %butun program bitir
                masurpar1=0 ; %yol dongsu bitir
            end
            taramasira=taramasira+1;%baslangic arama artirmasi
        end
        cbittop=[0 0];
        siratop=0;
        while masurpar1
            m13=1;
            masurpar1=0;

for bahi=1:1:lengu

        bitisirala=cbitlari(bahi,:);
        mekiyama=[bitisirala;refba];
        refbasuz= pdist(mekiyama);
        refbasuzround=round(refbasuz);
```

09.01.2022 16:10 E:\matlab2016benim...\farkliyollarson.m 2 of 2

```
if refbasuzround==0
    bassira=bahi;
    masurparl=1;% bir buldugu zaman dongu tamamlamasi gerek
    cbittopy=cbitlari(bahi,:);
    cbittop=[cbittop;cbittopy];%bir cellde toplamak amaciyla
    siratopy=bahi;
    siratop=[siratop;siratopy];
    cbitlari(bahi,:)= [0 0];

end

end

    refbax=baslagicxler(bassira);
    refbay=baslagicyeler(bassira) %onemli
    refba=[refbax refbay];

end

M{iza}=cbittop;
S{iza}=siratop; %sira toplamasi
iza=iza+1;

end
```

Ek Açıklama-G: Etiket Noktalarını bir Hücre İçinde Toplama Programı

09.01.2022 16:39 E:\matlab2016benim\bin\gericizim.m 1 of 1

```
% geri cizim programi
% bir yol halinde gitirme programi
% farkliyollarson dan gelen yol icin (4) olarak calistiriyoruz
YCT=[A,YYB];
yolnoktop=[0 0];
yolsayi=length(S)-1; %bir tanesi fazla oldugu icin
for dongusayi=1:1:yolsayi
    gh1=S{dongusayi};
    donguuzun=length(S{dongusayi})-1;
    yolnoktop=[0 0];
    for dongugezi=1:1:donguuzun
        dongugezi;
        gL=length(gh1);
        gE=gh1(2:gL);
        gh2=gE(dongugezi);
        gh3=YCT{gh2};
        cizgiz=length(gh3);
        for ciznoktasayi=1:1:cizgiz
            ciznoktagos=gh3(ciznoktasayi,:);
            yolnoktopn=ciznoktagos;
            yolnoktop=[yolnoktop;yolnoktopn];
            ciznoktasayi=ciznoktasayi+1;
        end
        dongugezi=dongugezi+1;
        J{dongusayi}=yolnoktop;
    end
    dongusayi=dongusayi+1;
end
```

Ek Açıklama-H: Planlanan Yolun Simülasyonu için Yazılan Program

09.01.2022 16:52 E:\matlab2016be...\similasyonprogrami.m 1 of 1

```
%(5)
% similasyon calismasi oncekiler calismasi gerek

juzunlugu= length(J);
for jgizimi=1:1:juzunlugu
    jgizimi;
    yoliceri=J(jgizimi);
    yoluzunlugu=length(yoliceri);
    for yolgizimi=2:1:yoluzunlugu
        yolcizme= yoliceri(yolgizimi,:);
        yolcizmex=yolcizme(1);
        yolcizmey=yolcizme(2);
        plot(yolcizmex,yolcizmey,'*')
        hold on
        pause(0.1)
    end
    jgizimi=jgizimi+1;
end
```

Ek Açıklama-I: Planlanan Yolda Robotun Açılarının Bulma Programı

09.01.2022 18:47 E:\matl...\similasyonterskinematikpro.m 1 of 1

```

% simülasyon ters kinematik programı
% (5) olarak çalışmaktadır

L1=45;
L2=45;
L(1) = Link([0 0 L1 0]);
L(2) = Link([0 0 L2 pi]);

scararobot=SerialLink(L);
scararobot.name='5eksenlisscararobot';

th1top=0;
th2top=0;

juzunlugu= length(J);
for jgizimi=1:1:juzunlugu

    yoliceri=J{jgizimi};
    yoluzunlugu=length(yoliceri);
    for yolgizimi=2:1:yoluzunlugu
        yolcizme= yoliceri(yolgizimi,:);
        yolcizmex=yolcizme(1);
        yolcizmey=yolcizme(2);
        % plot(yolcizmex,yolcizmey,'*')

        %%%%%%%%% ters kinematik
        xe=yolcizmex;
        ye=yolcizmey;

        costh2=((xe.^2)+(ye.^2)-(L1.^2)-(L2.^2))/(2*L1*L2);
        costh2q= (costh2).^2;
        sinh2=sqrt(1-costh2q);
        th2=atan2(sinh2,costh2);
        sicth1=(L1+L2*cos(th2))*ye-(L2*sin(th2))*xe;
        coth1=(L1+L2*cos(th2))*xe+(L2*sin(th2))*ye;
        th1=atan2(sicth1,coth1);
        t=scararobot.fkine([th1 th2]);
        r=t(:,4);
        plot(r(1),r(2),'*')
        hold on
        pause(0.1)

        %%%%%%%%%
        th1top=[th1top;th1];
        th2top=[th2top;th2];
    end
    jgizimi=jgizimi+1;
end

```

Ek Açıklama-J: Linspace Limiti içinde Açıkların Çizilme Programı

10.01.2022 21:58 E:\matlab2...\acilarcizmeprolinespace.m 1 of 1

```
% (6) linspace kullanarak acilar çizme programi

acilaruzun=length(th1top);
cizmedilimi=linspace(0,10,acilaruzun);
for thlerciz=1:1:acilaruzun
    th1cikar=th1top(thlerciz);
    th2cikar=th2top(thlerciz);
    cizmedilimicikar=cizmedilimi(thlerciz);

    plot(cizmedilimicikar,th1cikar,'.r',cizmedilimicikar,th2cikar,'.b')

    hold on
    legend('th1','th2')
    thlerciz=thlerciz+1;
end
```