

**DOKUZ EYLÜL ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**SCARA ROBOT MANİPULATÖR İLE SEÇME VE  
YERLEŐTİRME UYGULAMASI**

**Mehmet ÖZKARAKOÇ**

**Haziran, 2009**  
**İZMİR**

# **SCARA ROBOT MANİPULATÖR İLE SEÇME VE YERLEŐTİRME UYGULAMASI**

**Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü**

**Yüksek Lisans Tezi**

**Mekatronik Mühendisliđi Bölümü, Mekatronik Mühendisliđi Anabilim Dalı**

**Mehmet ÖZKARAKOÇ**

**Haziran, 2009**

**İZMİR**

## YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

**MEHMET ÖZKARAKOÇ**, tarafından **PROF. DR. EROL UYAR** yönetiminde hazırlanan “**SCARA ROBOT MANİPULATÖR İLE SEÇME VE YERLEŞTİRME UYGULAMASI**” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

---

Prof. Dr. Erol UYAR

---

Yönetici

---

Yrd. Doç. Dr. Zeki KIRAL

---

Jüri Üyesi

---

Doç. Dr. M. Evren TOYGAR

---

Jüri Üyesi

---

Prof. Dr. Cahit HELVACI

Müdür

Fen Bilimleri Enstitüsü

## TEŐEKKÜR

Hazırlamıő olduđum bu alıőmada tecrübesi ve bilgisi ile bana yol gösteren danıőman hocam, Sayın Prof. Dr. Erol UYAR'a teőekkürlerimi sunarım.

Ayrıca, programlama konusunda yardımını esirgemeyen Bil. Müh. Nihat Engin TOKLU'ya teőekkür ederim.

Mehmet ÖZKARAKO

# SCARA ROBOT MANİPULATÖR İLE SEÇME VE YERLEŞTİRME UYGULAMASI

## ÖZ

Bu çalışmada, bir SCARA (Selectively Compliant Articulated Robot for Assembly) Robot Manipulatör'ün konum kontrolü üzerine çalışılmıştır.

Robotun konum kontrolünün gerçekleştirilebilmesi için, ileri ve ters kinematik problemleri çözülmüş, jakobiyen ve transformasyon matrisleri hesaplanmıştır. Pnömatik kısımların (silindir, tutucu) çalıştırılabilmesi için uygun valfler seçilmiş, bu valflerin tetiklenmesi için devreler imal edilmiştir. Motorların tahriki için sürücü devreler imal edilmiştir. Geri besleme elemanları (potansiyometreler), motor sürücü devreleri ve valf tetik devreleri ile Visual Studio ile yazılan kontrol programı arasında iletişimi sağlayacak uygun veri toplama kartı seçilmiştir.

Yazılan kontrol programında elde edilen konum ve açı değerlerini doğrulamak için SolidWorks programında robotun birebir modeli yapılmıştır. Sonuçların uyumlu olduğu gözlemlenmiştir.

**Anahtar sözcükler :** SCARA, İleri Kinematik, Ters Kinematik, Jakobiyen Matrisi, Transformasyon Matrisi, Tam Köprü.

# **PICK AND PLACE APPLICATIONS BY USING A SCARA ROBOT MANIPULATOR**

## **ABSTRACT**

In this study, a SCARA (Selectively Compliant Articulated Robot for Assembly) Robot Manipulator's position control is studied.

To perform robot's position control, forward and inverse kinematics problems were solved, jacobian and transformation matrixes were calculated. Appropriate valves were selected for operating pneumatic components (cylinder, gripper). Circuits were made to trigger the valves. To excite the DC motors, driver circuits were made. An eligible data acquisition card was chosen which provides communication among control program was written with Visual Studio and specified equipments which are feedback components (potentiometer), motor driver circuits and valve driver circuits.

To verify position and angle values which are calculated with the control programme, the robot was designed with SolidWorks programme. At the end of the verifications observed results are compatible.

**Keywords :** SCARA, Forward Kinematic, Inverse Kinematic, Jacobian Matrix, Transformation Matrix, Full Bridge.

# İÇİNDEKİLER

Sayfa

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU .....	ii
TEŞEKKÜR.....	iii
ÖZ .....	iv
ABSTRACT.....	v
<b>BÖLÜM BİR – GİRİŞ .....</b>	<b>1</b>
1.1 Robot .....	1
<b>BÖLÜM İKİ – ROBOT KİNEMATİĞİ .....</b>	<b>2</b>
2.1 Scara Robot Manipulatör Hakkında Genel Bilgiler .....	2
2.2 Scara Robot Manipulatör Çalışma Alanı.....	3
2.3 Robotların Kinematik Analizi .....	4
2.3.1 Robotların İleri Kinematığı .....	4
2.3.1.1 Denavit – Hartenberg Yöntemi .....	4
2.3.1.2 Eklemlere Koordinat Sistemi Yerleştirilmesi .....	5
2.3.1.3 Robot Kinematığının Çıkarılmasında Uygulanan Kurallar .....	5
2.3.2 Robotların Ters Kinematığı .....	6
2.3.2.1 Ters Kinematik Problemi .....	6
2.4 Scara Robot Manipulatörün İleri Kinematığı.....	7
2.5 Scara Robot Manipulatörün Ters Kinematığı .....	9
2.5.1 Scara Robot Manipulatörün Jakobiyen Matrisi.....	12
2.5.1.1 Scara Robot Manipulatörün Sayısal Yöntemle Ters Kinematığı...13	
2.5.1.2 Tekil Noktalar .....	14
<b>BÖLÜM ÜÇ – DC MOTOR KONTROL TEORİSİ .....</b>	<b>16</b>

3.1 Giriş .....	16
3.2 Motor Sürücü Devreleri.....	16
3.2.1 Alan Kontrolü ile DC Motor Hız Kontrolü .....	16
3.2.2 Endüvi Devresi Direnç Kontrolü ile DC Motor Hız Kontrolü .....	18
3.2.3 Armatür Voltajı Kontrolü ile DC Motor Hız Kontrolü .....	18
3.2.4 Ward–Leonard Sistemi ile DC Motor Hız Kontrolü .....	19
3.2.5 Yarı İletkenler Kullanılarak DC Motor Hız Kontrolü .....	20
3.2.6 Tam Köprü (H Köprüsü) ile Yön Kontrolü .....	21
<b>BÖLÜM DÖRT – KONTROL DONANIMI.....</b>	<b>23</b>
4.1 DC Motorların Sürücü Devresi .....	23
4.2 Silindir ve Tutucunun Sürücü Devresi .....	25
4.3 Veri Toplama Kartı .....	27
4.4 Konektör ve Genel Bağlantı Şeması .....	27
<b>BÖLÜM BEŞ – ROBOT KONTROLÜ .....</b>	<b>29</b>
5.1 Temel Kontrol Şekilleri.....	29
5.2 Kontrol Programı.....	30
<b>BÖLÜM ALTI – SONUÇ.....</b>	<b>33</b>
<b>KAYNAKLAR .....</b>	<b>33</b>
<b>EKLER.....</b>	<b>35</b>
EK 1.....	35
EK 2.....	37
EK 3.....	39



# BÖLÜM BİR

## GİRİŞ

### 1.1 ROBOT

Bilgisayar destekli tasarım (Computer Aided Design-CAD) ve bilgisayar destekli üretim (Computer Aided Manufacturing-CAM) gibi tekniklerin bulunmasıyla, endüstriyel otomasyon sistemleri kısa zamanda çok hızlı gelişmiştir. Bu hızlı gelişmenin sonucunda endüstriyel robotların kullanımı ivme kazanmıştır. Günümüze kadar büyük aşama kaydeden robotlar, madde taşınması (konveyör sistemler), boyacılık, kaynak endüstrisi, tıp vs. gibi bir çok sektörde kullanılmaktadır (Bingül ve Küçük, 2005, s. 1).

Robot, işçilik maliyetini azaltmak, tehlikeli ve riskli yerlerde çalışanların yerini almak, daha esnek bir üretim sağlamak, daha tutarlı bir kalite kontrol sağlamak, çıktı miktarını artırmak, vasıflı işçilik sıkıntısını karşılayabilmek, insana göre daha çabuk sonuca ulaşma kabiliyeti, vs. konularındaki üstünlüklerinden dolayı kullanımı avantajlıdır. Robotlar düşünemez, programlanmadan çalışamaz, boşa geçen bakım ve onarım zamanlarına sahip olmaları, yatırım maliyetleri vs. gibi sebeplerden dolayı ise kullanımı insan gücüne göre bir miktar dezavantajlıdır.

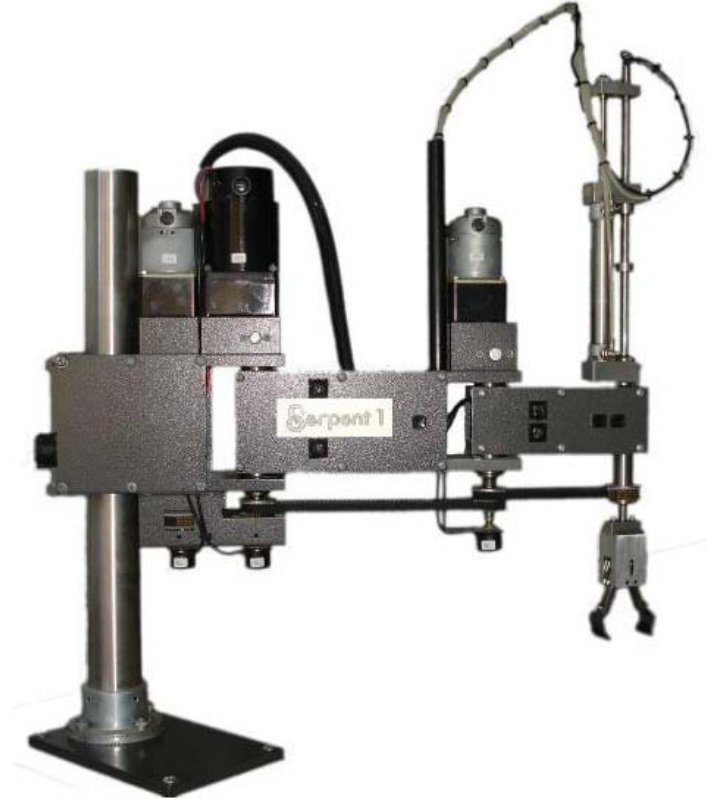
Bu çalışmada, Scara düzenlemesine sahip bir robotun sürücü devrelerinin yapımı, pnömatik yapıya uygun kontrol valflerinin seçimi ve aç-kapa mantığı ile çalışan kontrol programının yazımı üzerine çalışılmıştır.

## BÖLÜM İKİ

### ROBOT KİNEMATİĞİ

#### 2.1 Scara Robot Manipulatör Hakkında Genel Bilgiler

Projede kullanılan Scara robot dört serbestlik derecesine sahip olması nedeniyle birçok montaj ve seçme-yerleştirme işlemlerine uygun yapıdadır. Robotun asıl amacı bir cisim tutma, kaldırma ve yer değiştirmedir. Bu hareket, üç adet dc motor, silindir ve tutucu yardımı ile gerçekleştirilmektedir. Şekil 2.1’de Scara robot gösterilmiştir.



Şekil 2.1 Scara robot manipulatör.

Motorlar, gövdede 2 adet ( $M_3$ ,  $M_1$ ), ana kolda 1 adettir ( $M_2$ ). 3 nolu motor  $450^\circ$ ’lik, 2 nolu motor  $250^\circ$ ’lik, 1 nolu motor  $192^\circ$ ’lik çalışma alanı oluşturmaktadır. Pnömatik hareketler, kompresörle elde edilen basınçlı hava ile yapılmaktadır. Bu hava, tutma kolunun Z eksenini boyunca yukarı aşağı hareketi ile tutucunun açılıp kapanması için kullanılır. Tutma kolu için sadece, en yukarı ve en aşağı pozisyonu geçerlidir, ara pozisyon yoktur. Scara robotun teknik özellikleri Tablo 2.1’de verilmiştir.

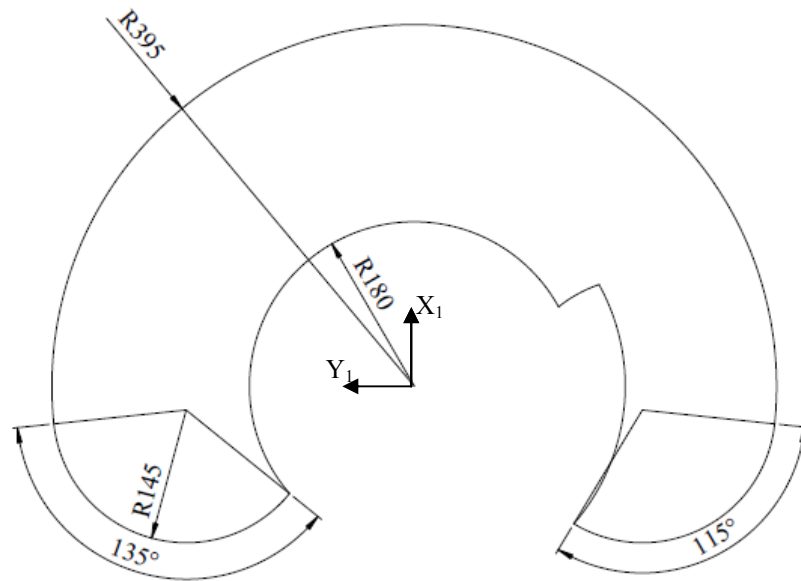
Tablo 2.1 Robot'un teknik bilgileri

Gövde Uzunluğu	170 mm
Ana Kol Uzunluğu	250 mm
Ön Kol Uzunluğu	145 mm
Ana Kol Açısı	$\pm 96^\circ$
Ön Kol Açısı	$135^\circ, -115^\circ$
Tutma Kolu Açısı	$0^\circ, 450^\circ$
Tutma Kolu Kursu	75 mm
Basınç Aralığı	5 - 8 bar

## 2.2 Scara Robot Manipulatör Çalışma Alanı

Çalışma uzayının şekli bir robot manipulatörün'ün serbestlik derecesine ve kolu oluşturan ilk üç eklem koordinatlarının yerleşimine bağlıdır. Birbirine dik veya paralel yerleştirilen ilk üç eklem genellikle prizmatik, silindirik ve küresel bir çalışma uzayı üretir. Bazı çalışma uzayları ise birkaç çalışma uzayı şeklini birden içerir.

Robotun ulaşabileceği yerleri tespit edebilmemiz için çalışma uzayı Şekil 2. 2'de gösterilmiştir.



Şekil 2.2 Scara robot manipulatör'ün çalışma alanı.

## 2.3 Robotların Kinematik Analizi

Bir sistemin yapısını ve hareketlerini inceleyen bilim dalına kinematik denir. Robot kinematiği yardımı ile robotun konum, hız ve ivme analizi yapılır. Uç işlevci (end effector) ile eklemler arasında ilişki tanımlanır.

Her bir robot eklemının konumu, bir öncekine veya bir sonrakine göre tanımlanır. Arka arkaya oluşturulan bu ilişkiye açık kinematik zincir denir. Bu ilişkiyi oluşturan ifadeler, robotun konum (position) ve yönelim (orientation) bilgisini içeren 4x4 homojen dönüşüm matrislerinden oluşur. Her bir eklem için ayrı ayrı homojen dönüşüm matrisleri oluşturulur. Oluşturulan bu dönüşüm matrislerinin sayısı, robotun serbestlik derecesine bağlıdır.

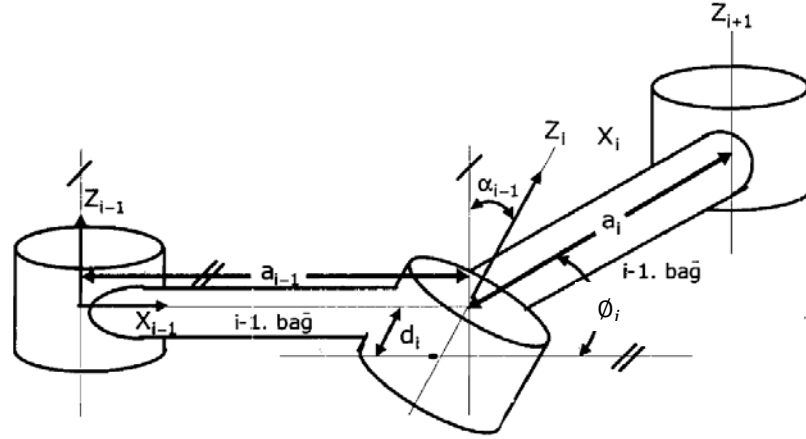
### 2.3.1 Robotların İleri Kinematiği

Robotun ileri kinematiği (forward kinematics), robot bağlarının konumları, hızları ve ivmeleri ile ilgilenir. Bir robot, ana çerçevesinden uç işlevci çerçevesine doğru birbirine prizmatik veya dönelemlerle tutturulmuş seri bağlardan oluşur. Bağlar arasındaki ilişki dönüşüm matrisleri ile açıklanır. Eklem dönüşüm matrislerinin art arda çarpılmasıyla, ana çerçeveye uç işlevci çerçevesi arasındaki ilişki tanımlanır. Bu ilişki uç işlevcinin yönelimi ve konumunu ana çerçeveye göre verir.

#### 2.3.1.1 Denavit - Hartenberg Yöntemi

Robotların eklem değişkenlerini belirlemek için bir çok kinematik yöntem geliştirilmiştir ve bunlardan en yaygın olarak kullanılanı Denavit - Hartenberg yöntemidir. Bu yöntemde dört ana değişken kullanılarak robotun kinematiği çıkarılmaktadır. Bu değişkenler, iki eksen arasındaki bağ uzunluğu  $a_{i-1}$ , iki eksen arasındaki bağ açısı  $\alpha_{i-1}$ , üst üste çakışan bağlar arasındaki eklem kaçıklığı  $d_i$  ve iki bağ arasında oluşan eklem açısı  $\theta_i$ 'dir. Bu dört değişkene D - H değişkenleri denir. Bu değişkenleri belirlemek için robotun dönme eksenleri belirlenir ve dönme eksenleri bağlardan bir fazla olacak şekilde numaralandırılır. Bu eksenlerin her birine bir koordinat sistemi yerleştirilir ve bağ dönme eksenini, koordinat sisteminin Z eksenini kabul edilir. Bağ doğrultusundaki uzunluk ise

koordinat sisteminin X eksenini olarak kabul edilir. D-H değişkenleri Şekil 2.3'de verilmiştir.



Şekil 2.3 D-H değişkenlerinin belirlenmesi (Bingül ve Küçük, 2005).

### 2.3.1.2 Eklemlere Koordinat Sistemi Yerleştirilmesi

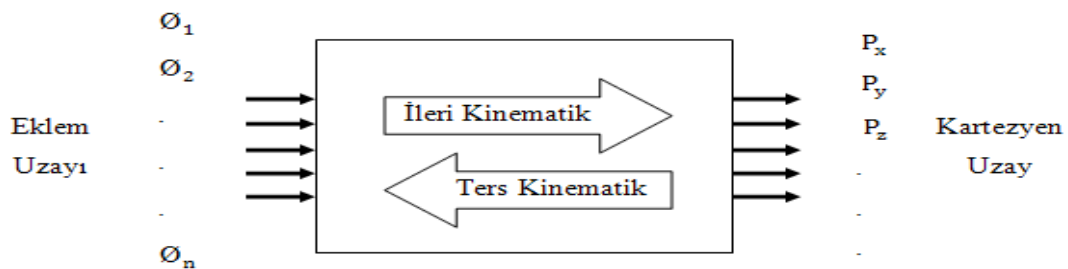
Eklemlere koordinat sistemi yerleştirilirken öncelikle eklem eksenlerinin dönme veya kayma yönleri belirlenir ve bu eksene paralel bir doğru çizilir. Döner eksenler için dönme yönü Z, prizmatik eklemler için kayma yönü Z olarak belirlenir. Z eksenine dik ve kol boyunca olan bağ uzunluğu X eksenini olarak kabul edilir. Z ve X eksenleri belirlendikten sonra son olarak sağ el kuralına göre Y eksenini belirlenir.

### 2.3.1.3 Robot Kinematığının Çıkarılmasında Uygulanan Kurallar

İlk olarak her eklem koordinat sistemi yerleştirilir. Her eklem için D-H değişkenleri belirlenerek tablo oluşturulur. Tablonun her satırında bulunan değişkenlerle bir eklem için dönüşüm matrisleri elde edilir. Dönüşüm matrisi sayısı robotun serbestlik derecesini belirler. Tabloda bulunan  $a_{i-1}$  ve  $\alpha_{i-1}$  robotun hareket etmesiyle değişmeyen sabit parametrelerdir,  $d_i$  ve  $\phi_i$  robotun hareketi ile değişen parametrelerdir. Her bir eklem için sadece bir tane değişkeni bulunur. Eklemler için ayrı ayrı elde edilen dönüşüm matrislerinin çarpılması ile ana çerçeveden uç işlevci çerçevesine doğru ileri robot kinematığı çıkarılmış olur. Dönüşüm matrislerinin çarpılması ile uç işlevcinin konumu, yönelimini içeren ve eklem değişkenlerinin bir fonksiyonu olan genel dönüşüm matrisi elde edilir.

### 2.3.2 Robotların Ters Kinematığı

Kartezyen uzayda, ana çerçeveye göre verilen uç işlevcinin konum ve yönelim bilgilerine göre eklem değişkenlerinin bulunmasına ters kinematik denir. Başka bir deyişle, bir robot manipulatörünün uç işlevcisinin konumu ve yönelimini Kartezyen koordinat sisteminden eklem koordinat sistemine dönüştürmeye ters kinematik denir. Şekil 2.4’de ileri ve ters kinematik şematik olarak gösterilmiştir. Robotların ters kinematığı, robot kontrolünün en önemli aşamalarından birini oluşturur. Ters kinematik probleminin çözümü; gerçek zamanlı kontrol, kaynak, boyama, yörünge planlaması gibi bir çok uygulamada son derece önemlidir.



Şekil 2.4 İleri ve ters kinematik probleminin şematik gösterimi.

#### 2.3.2.1 Ters Kinematik Problemi

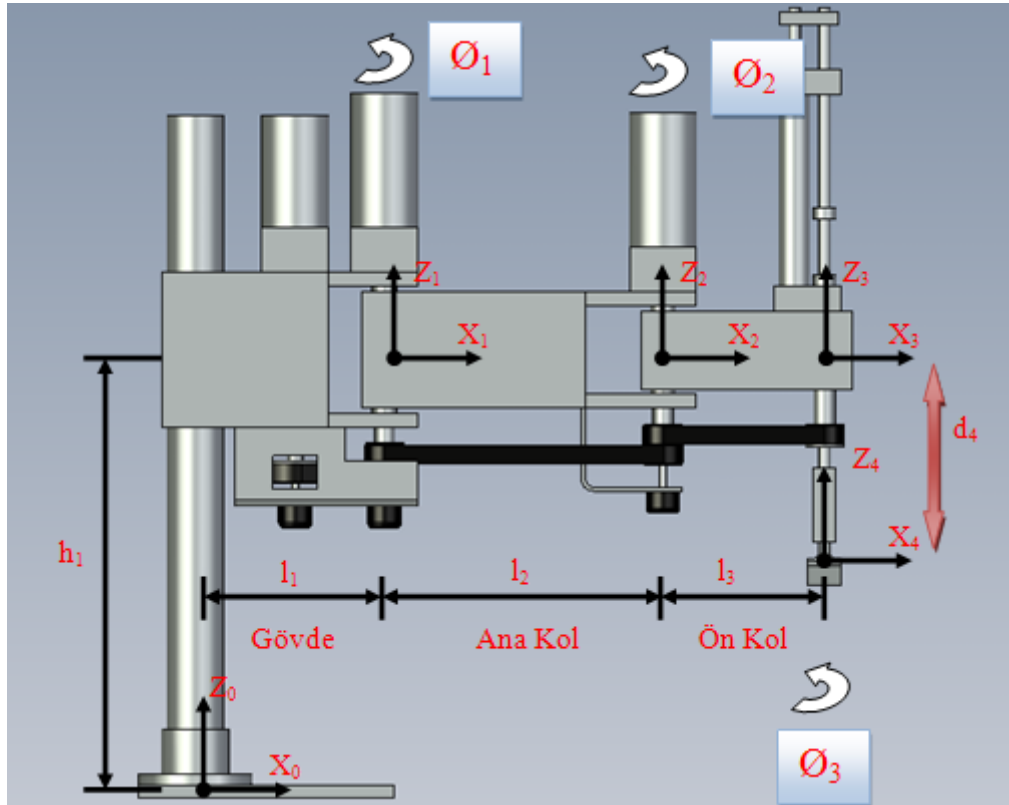
Ters kinematik problem, çözülmesi oldukça güç bir problemdir. Doğrusal olmayan denklemler içerir. Bu sebeple her zaman matematiksel çözüm fiziksel çözümü temsil etmez. Matematiksel çözümlerin yalnızca bir kısmı fiziksel çözümü temsil eder. Çözüm sayısı robotun serbestlik derecesinin yanında eklem değişkenlerine de bağlıdır. Robot çoğunlukla prizmatik eklemlerden oluşuyorsa çözüm o derece kolaylıkla elde edilirken, dönel eklem sayısı artıkça çözüm de o derece zorlaşmaktadır.

Ters kinematik problem, robot için tamamen analitik olarak çözülebileceği gibi, analitik çözümün mümkün olmadığı durumlarda sayısal yöntemler kullanılarak da çözülebilir. Tamamen kesin sonuç üreten analitik çözüme ait denklemler bilgisayar ortamında çok hızlı çözülürken, eklem açılarının iteratif olarak çözüldüğü sayısal çözüm ise bilgisayar ortamında analitik çözüme göre daha yavaş çalışır. Ayrıca, sayısal çözüm ile eklem açılarını bulmak için yazılan algoritmada belirtilen çözüm zamanı ve başlangıç koşulları son derece önemlidir.

Ters kinematik problemin sayısal yöntemlerle çözülmesinde karşılaşılan en büyük sorun, Jakobiyen matrisinin tekil olduğu noktalarda yazılan algoritmanın çözüm üretmemesidir. Ayrıca, başlangıç çözüm vektörü (eklem değişkenlerini içeren vektör) çözüme yönelik vektör elemanlarından oluşmadığı zaman, sayısal olarak kararlı bir çözüme ulaşılamaz. Sadece sayısal veya sadece analitik çözüm gerçekleştirilemediği durumlarda ise, eklem değişkenlerinden bir kaçını analitik olarak bulunup, geri kalanlar için sayısal olarak çözülebilen yarı analitik çözüm uygulanabilir.

## 2.4 Scara Robot Manipulatörün İleri Kinematığı

Bu başlık altında, incelediğimiz scara robotun ileri kinematığı çözülecektir.



Şekil 2.5 Scara robot manipulatöre koordinat sistemlerinin yerleştirilmesi.

i. ekleme yerleştirilen  $\{i\}$  koordinat sisteminin  $i-1$ . ekleme yerleştirilen  $\{i-1\}$  koordinat sistemine göre belirlenen  $a_{i-1}$  (x eksenindeki bağ uzunluğu),  $\alpha_{i-1}$  (x ekseninin rotasyon miktarı),  $d_i$  (z eksenine göre öteleme miktarı) ve  $\Theta_i$  (z eksenine göre dönme miktarı) eklem değişkenlerinden yararlanarak robotun bir eklemine ait

dönüşüm matrisi elde edilir. Bu dört değişkenin meydana getirdiği matrislerin çarpımı ile, 4 serbestlik derecesine sahip robotumuzun yalnızca bir eklemine ait dönüşüm matrisi elde edilir. Şekil 2.5’de robota koordinat sistemlerinin yerleştirilmesi gösterilmiştir.

$${}^{i-1}T_i = R_x(\alpha_{i-1})D_x(a_{i-1})R_z(\phi_i)D_z(d_i) \quad (2.1)$$

$${}^{i-1}T_i = \begin{bmatrix} \cos \phi_i & -\sin \phi_i & 0 & a_{i-1} \\ \sin \phi_i \cos \alpha_{i-1} & \cos \phi_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \phi_i \sin \alpha_{i-1} & \cos \phi_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Tablo 2.2 D - H değişkenlerinin belirlenmesi.

Eksen no	D-H Değişkenleri				Değişken
i	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\phi_i$	$d_i$ veya $\phi_i$
1	0	$l_1$	$h_1$	$\phi_1$	$\phi_1$
2	0	$l_2$	0	$\phi_2$	$\phi_2$
3	0	$l_3$	0	$\phi_3$	$\phi_3$
4	0	0	$-d_4$	0	$d_4$

Tablo 2.2’de D - H değişkenlerinin belirlenmesi için denklem 2.2’deki genel dönüşüm (transformasyon) matrisi kullanılır.

$${}^0T_1 = \begin{bmatrix} \cos \phi_1 & -\sin \phi_1 & 0 & l_1 \\ \sin \phi_1 & \cos \phi_1 & 0 & 0 \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} \cos \phi_2 & -\sin \phi_2 & 0 & l_2 \\ \sin \phi_2 & \cos \phi_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} \cos \phi_3 & -\sin \phi_3 & 0 & l_3 \\ \sin \phi_3 & \cos \phi_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$${}^3_4T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Yukarıdaki dönüşüm matrislerinin birbirleri ile çarpımıyla robotumuzun ileri yönlü kinematiği çıkar.

$${}^0_4T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T \quad (2.3)$$

$${}^0_4T = \begin{bmatrix} c(\phi_1 + \phi_2 + \phi_3) & -s(\phi_1 + \phi_2 + \phi_3) & 0 & c(\phi_1 + \phi_2)l_3 + c(\phi_1)l_2 + l_1 \\ s(\phi_1 + \phi_2 + \phi_3) & c(\phi_1 + \phi_2 + \phi_3) & 0 & s(\phi_1 + \phi_2)l_3 + s(\phi_1)l_2 \\ 0 & 0 & 1 & -d_4 + h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Transformasyon matrislerinin çarpımıyla uç işlevcinin konumunu ve yönelimini içeren, eklem değişkenlerinin birer fonksiyonu olan genel dönüşüm matrisi denklem 2.4'de elde edilmiştir. Uç işlevcinin konumu ( $P_x, P_y, P_z$ ) sabitlerin ilgili denklemlerde yerine konması ile elde edilir. Uç işlevcinin konumu;  $P_x = {}^0_4T(1,4) = \cos(\phi_1 + \phi_2)145 + \cos(\phi_1)250 + 170$ ,  $P_y = {}^0_4T(2,4) = \sin(\phi_1 + \phi_2)145 + \sin(\phi_1)250$ ,  $P_z = {}^0_4T(3,4) = -d_4 + 75$  ile hesaplanır. Scara robot manipulatörün ileri kinematik transformasyon matrisinin oluşturulması ve çözümü için yazılan Matlab dosyası Ek 1'de verilmiştir.

## 2.5 Scara Robot Manipulatörün Ters Kinematiği

Bu başlık altında, incelediğimiz scara robotun ters kinematiği çözülecektir. Ters kinematik problem analitik olarak karmaşık ve doğrusal olmayan denklemler içerdiğinden çözülmesi oldukça güçtür. Gerçek zamanlı kontrol uygulamaları gerçekleştirilirken herhangi bir robota ait ters kinematik problemin kesin çözümü tercih edilir. Kesin çözüm, problemin analitik olarak çözülmesi ile elde edilir. Analitik çözüm yaparken bazı trigonometrik eşitliklerden faydalanılır. Bu eşitlikler aşağıda verilmiştir.

$$\cos \phi = a \text{ ise } \phi = \arctan(\pm\sqrt{1 - a^2}, a) \quad (2.5)$$

$$b \sin \phi + c \cos \phi = d \text{ ise } \phi = \arctan(b, c) + \arctan(\pm\sqrt{b^2 + c^2 - d^2}, d) \quad (2.6)$$

4 serbestlik derecesine sahip robotumuzun ileri kinematiği denklem 2.3'deki gibi yazılabilir.  ${}^0T_4$  ileri yön kinematik matrisi, konum ve yönelim bilgilerini içeren matris elemanları cinsinden denklem 2.7'deki gibi yazılıp denklem 2.8'deki gibi eşitlik sağlanır.

$${}^0T_4 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$[{}^0T_1]^{-1}{}^0T_4 = [{}^0T_1]^{-1}{}^1T_2{}^2T_3{}^3T_4 \quad (2.8)$$

Denklemden  $[{}^0T_1]^{-1}{}^1T_2 = I$  olduğundan, denklem 2.8 aşağıdaki gibi daha basit bir şekilde ifade edilebilir.

$$[{}^0T_1]^{-1}{}^0T_4 = {}^1T_2{}^2T_3{}^3T_4 \quad (2.9)$$

Denklem 2.9'un çözümü yapılarak denklem 2.10'daki eşitlik elde edilir.

$$\begin{bmatrix} \cos(\phi_1) r_{11} + \sin(\phi_1) r_{21} & \cos(\phi_1) r_{12} + \sin(\phi_1) r_{22} \\ -\sin(\phi_1) r_{11} + \cos(\phi_1) r_{21} & -\sin(\phi_1) r_{12} + \cos(\phi_1) r_{22} \\ r_{31} & r_{32} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \cos(\phi_1) r_{13} + \sin(\phi_1) r_{23} & \cos(\phi_1) P_x + \sin(\phi_1) P_y - \cos(\phi_1) l_1 \\ -\sin(\phi_1) r_{13} + \cos(\phi_1) r_{23} & -\sin(\phi_1) P_x + \cos(\phi_1) P_y + \sin(\phi_1) l_1 \\ r_{33} & P_z - h_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\phi_2 + \phi_3) & -\sin(\phi_2 + \phi_3) & 0 & \cos(\phi_2)l_3 + l_2 \\ \sin(\phi_2 + \phi_3) & \cos(\phi_2 + \phi_3) & 0 & \sin(\phi_2)l_3 \\ 0 & 0 & 1 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Denklem 2.10'daki eşitliğin sol tarafına "e", sağ tarafına "f" dersek, matris eşitliğinden de faydalanarak, denklem 2.11 ve denklem 2.12'yi elde ederiz.

$$e(1,4) = f(1,4) \text{ ise } \cos(\phi_1) P_x + \sin(\phi_1) P_y - \cos(\phi_1) l_1 = \cos(\phi_2)l_3 + l_2 \quad (2.11)$$

$$e(2,4) = f(2,4) \text{ ise } -\sin(\phi_1) P_x + \cos(\phi_1) P_y + \sin(\phi_1) l_1 = \sin(\phi_2)l_3 \quad (2.12)$$

Denklem 2.11 ile denklem 2.12’de eşitliğin her iki tarafının karelerini alıp alt alta topladığımızda denklem 2.13’ü elde ederiz.

$$P_x^2 - 2P_x l_1 + P_y^2 + l_1^2 = 2 \cos \phi_2 l_3 l_2 + l_2^2 + l_3^2 \quad (2.13)$$

Denklem 2.13, denklem 2.5’e benzetilerek 2 adet  $\phi_2$  açısı bulunur.

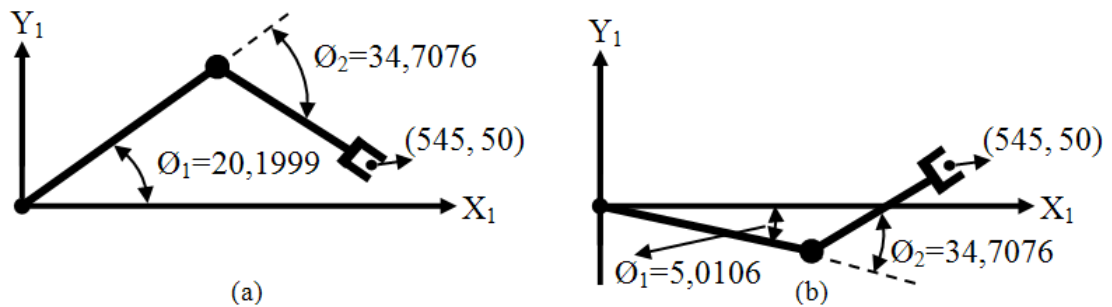
$$a = \cos \phi_2 = \frac{P_x^2 - 2P_x l_1 + P_y^2 + l_1^2 - l_2^2 - l_3^2}{2l_3 l_2} \text{ ise } \phi_2 = \tan^{-1} \left( \pm \frac{\sqrt{1 - a^2}}{a} \right) \quad (2.14)$$

Denklem 2.11, denklem 2.6’ya benzetilerek 2 adet  $\phi_1$  açısı bulunur.

$$\phi_1 = \tan^{-1} \left( \frac{P_y}{P_x - l_1} \right) + \tan^{-1} \left( \pm \frac{\sqrt{P_y^2 + (P_x - l_1)^2 - (\cos \phi_1 l_3 + l_2)^2}}{\cos \phi_1 l_3 + l_2} \right) \quad (2.15)$$

Toplam dört adet çözüm kümesi bulmuş oluruz. Bunlardan sadece iki tanesi fiziksel çözümdür ve hedef konuma ulaşmamızı sağlar. Diğer iki çözüm ise matematiksel çözümdür, hedef konuma ulaşmamızı sağlayamaz. Çözümlerden hangilerinin gerçek çözüm olduğunu anlayabilmek için elde ettiğimiz değerleri ileri kinematik denklemlerinde gerekli yerlere koyarak değerlerin fiziksel çözüm olup olmadığı hakkında sağlama yapılır.

Tek bir konuma ulaşabilmek için elde edilen iki adet çözüm tek bir çözüme indirilebilir. Bunun için hedef konuma yaklaşma şeklini belirlememiz gerekir. Yaklaşma şekilleri, sol kol (left arm configuration) ve sağ kol (right arm configuration) düzenlemleridir. Scara robot’un hedefe yaklaşma şekilleri Şekil 2.6’da gösterilmiştir.

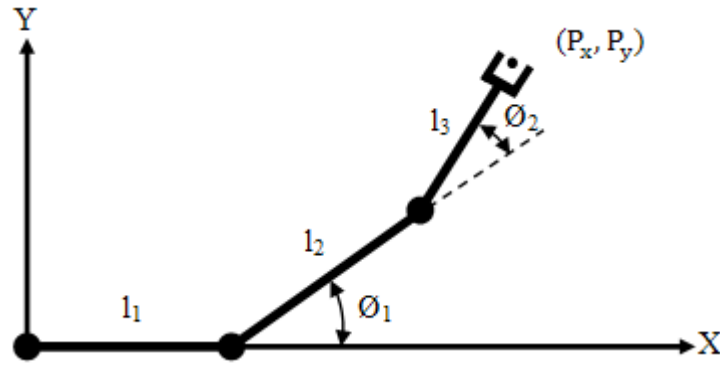


Şekil 2.6 Hedefe yaklaşma şekilleri (a) Sol kol düzenleşimi, (b) Sağ kol düzenleşimi.

Ters kinematik problemin çözümünde  $\theta_3$  açısı hesaplanmamıştır. Bu durumun sebebi,  $\theta_3$  açısının robotun uç işlevcisinin XY düzlemindeki konumunu etkilememesidir. Scara robot manipulatörün ters kinematik çözümü için yazılan Matlab dosyası Ek 2’de verilmiştir.

### 2.5.1 Scara Robot Manipulatörün Jakobiyen Matrisi

Ters kinematik problemler, analitik yöntemler dışında jakobiyen matris kullanarak sayısal yöntemlerle de çözülebilir. Jakobiyen matris, eklemlerdeki ufak değişimlerin uç işlevciye nasıl yansıtacağını gösterir. Şekil 2.7’de Scara robot manipulatörün düzlemsel zinciri gösterilmiştir.



Şekil 2.7 Scara robot manipulatörün düzlemsel zinciri

$$l_1 + l_2 \cos \theta_1 + l_3 \cos(\theta_1 + \theta_2) = P_x(\theta_1, \theta_2)$$

$$l_2 \sin \theta_1 + l_3 \sin(\theta_1 + \theta_2) = P_y(\theta_1, \theta_2)$$

Eklemlerdeki küçük yer değiştirmelerin uç işlevcinin konumuna etkisini çıkarabilmemiz için yukarıdaki kinematik eşitliklerin türevini alırız.

$$dP_x = \frac{\partial P_x(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial P_x(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2$$

$$dP_y = \frac{\partial P_y(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial P_y(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2$$

Yukarıdaki ifadeler vektör forma dönüştürülerek J matrisi olarak da adlandırılan Jakobiyen matrisi robotumuz için bulunur.

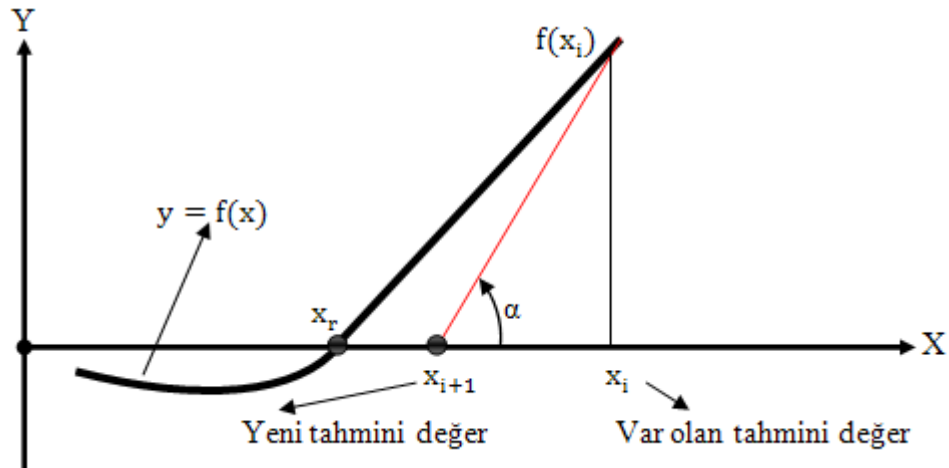
$$dp = \begin{pmatrix} dP_x \\ dP_y \end{pmatrix}, dq = \begin{pmatrix} d\theta_1 \\ d\theta_2 \end{pmatrix} \text{ ise } dp = Jdq$$

$$J = \begin{bmatrix} \frac{\partial P_x}{\partial \theta_1} & \frac{\partial P_x}{\partial \theta_2} \\ \frac{\partial P_y}{\partial \theta_1} & \frac{\partial P_y}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -l_2 \sin \theta_1 & -l_3 \sin(\theta_1 + \theta_2) \\ l_2 \cos \theta_1 & l_3 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (2.16)$$

### 2.5.1.1 Scara Robot Manipulatörün Sayısal Yöntemle Ters Kinematiği

Ters kinematik, analitik yöntem dışında robotun jakobiyen matrisi kullanılarak sayısal yöntemle de çözülebilir. Ters kinematik çözüm, bütün eklem değişkenlerinin bilgisayar ortamında yinelemeli işlemler gerçekleştirilerek Newton-Raphson yönteminde olduğu gibi sayısal olarak çözülebilir. Sayısal çözüm için kinematik eşitliklerin farklarının toplamını alan tahmin edici ve düzeltici algoritmalar kullanılır.

Newton-Raphson metodu olarak da bilinen Newton metodu, sürekli eşitlikleri eş zamanlı olarak çözer. Çözüm hızı ve basitliği sebebiyle sık kullanılan bir yöntemdir.



Şekil 2.8 Newton-Raphson metodu.

$$\tan \alpha = f'(x_i) = \frac{f(x_i)}{x_i - x_{i+1}} \text{ ise } x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (2.17)$$

Denklem 2.17’de metodun genel denklemi gösterilmiştir. Bu denklemi scara robot manipulatöre uyarlarken, robotun uç işlevcisinin düzlemdeki konumuna iki adet eklem ( $\theta_1, \theta_2$ ) değişkeninin etkidiğini görürüz.

$$F_1 = l_1 + l_2 \cos \theta_1 + l_3 \cos(\theta_1 + \theta_2) - P_x = 0 \quad (2.18)$$

$$F_2 = l_2 \sin \theta_1 + l_3 \sin(\theta_1 + \theta_2) - P_y = 0 \quad (2.19)$$

$$\begin{Bmatrix} \phi_1 \\ \phi_2 \end{Bmatrix}_{i+1} = \begin{Bmatrix} \phi_1 \\ \phi_2 \end{Bmatrix}_i - [J]^{-1} \begin{Bmatrix} F_{1i} \\ F_{2i} \end{Bmatrix} \quad (2.20)$$

Denklem 2.20, scara robot manipulatör'ün uç işlevcisinin XY düzlemindeki konumuna bağlı olarak eklem değişkenlerinin değerlerinin hesaplanmasını sağlar. Eklem değişkenlerinin hesaplanması için yazılan program Ek 3'de verilmiştir. Bu döngünün hesaba başlayabilmesi için  $i$  vektörünün elemanlarına tahmini değerler verilmiştir ve  $i+1$  vektörünün elemanları hesaplanmıştır.  $i+1$  vektörünün elemanları bir sonraki adımda  $i$  vektörünün elemanları olur ve yeniden  $i+1$  vektörünün elemanları hesaplanır.  $i+1$  vektörünün elemanlarının değerleri ile  $i$  vektörünün elemanlarının değerleri arasındaki fark istediğimiz değere düşünceye kadar hesaplama işlemi devam eder. Hata istediğimiz değere ulaştığında elde edilen  $i+1$  vektöründe bulunan eklem değişkenleri hedef koordinatımıza ulaşabilmemiz için gereken değerler olarak tespit edilmiş olur.

Programlamada  $i$  vektörüne ait değişkenlere atanan ilk tahmini değerler, kabul edilebilir hata değeri ve döngü sayısı gibi parametreler hesaplamamıza uygun seçilmelidir. Uygun seçilmeyen değerler sebebi ile hedef noktaya ulaşabilmemiz için gerekli olan eklem değişkenleri hesaplanamaz. Ayrıca program, jakobiyen matrisin tekil olduğu noktalarda çözüm üretemez.

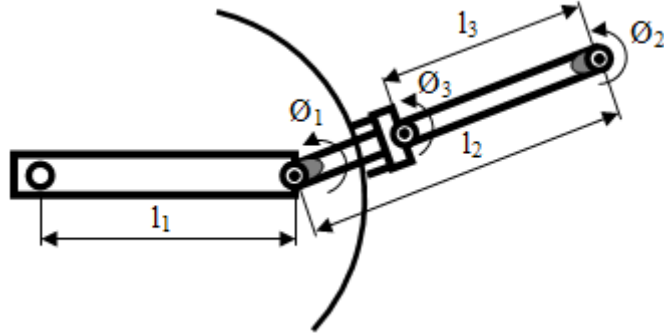
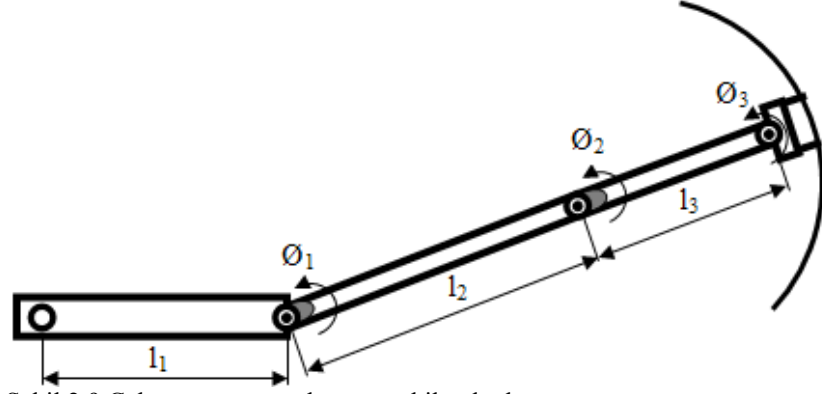
### 2.5.1.2 Tekil Noktalar

Bir matrisin determinanı sıfıra eşit ise o matrisin tersi alınamaz. Determinanı sıfır olan jakobiyen matris tekildir denir. Jakobiyen matris, eklem değişkenlerinin bazı değerlerinde tekildir. Jakobiyen matrisin tekil olduğu noktalarda robot serbestlik derecesini kaybeder ve kaybettiği bu noktalarda matris robot kontrolünde kullanılamaz.

Çoğu robot manipulatör jakobiyen matrisini tekil yapan eklem açılarına sahiptir. Bu açıların meydana getirdiği kartezyen noktalara robotun tekil noktaları denir. Tekil noktaları iki kısımda inceleyebiliriz.

- Şekil 2.9'da görüldüğü gibi robotun uzanabileceği en uzak noktalar çalışma uzayının dış sınır tekil noktalarını belirler.

- Şekil 2.10’da görüldüğü gibi robotun merkezine en yakın noktalar çalışma uzayının iç sınır tekil noktalarını belirler.



Scara robot manipulatör fiziksel kısıtlarından dolayı iç sınır tekil noktalara sahip değildir.

## BÖLÜM ÜÇ

### DC MOTOR KONTROL TEORİSİ

#### 3.1 Giriş

Veri toplama kartlarının analog çıkışları  $\pm 10V$  veya daha küçük bir çalışma aralığına sahiptir. Bu nedenle veri toplama kartlarının analog çıkışları kullanılarak direk bir DC motor kontrol edilemez. Direk kontrolün kart tarafından sağlanamamasının bir diğer sebebi de motorların sahip oldukları yüksek çalışma ve kalkınma akımlarıdır. Bu sorunun üstesinden gelmek için motor ve veri toplama kartı arasında uygun bir sürücü devresi kullanılır.

#### 3.2 Motor Sürücü Devreleri

Sürücü devreleri, motorun hızının ve dönüş yönünün kontrolü esaslarına göre farklı yöntemlerle tasarlanabilirler. DC motor hızı genellikle aşağıdaki beş adet yöntemle kontrol edilir:

- Alan kontrolü,
- Endüvi devresi direnç kontrolü,
- Armatür voltajı kontrolü,
- Ward-Leonard sistemi,
- Yarı iletkenler.

DC motor yön kontrolünde en çok tam köprü (H köprüsü) ile kullanılır.

##### 3.2.1 Alan Kontrolü ile DC Motor Hız Kontrolü

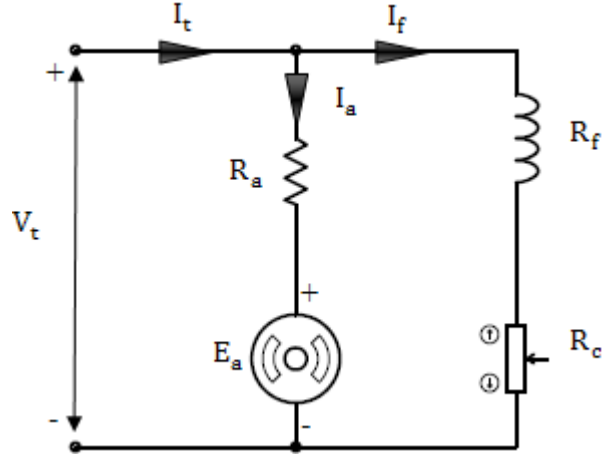
DC motorun rotorunun hızı ( $\omega_m$ ) ikaz alanı ( $\Phi$ ) ile denklem 3.1'de gösterildiği gibi değişir.

$$\omega_m = \frac{V_t}{K_a \Phi} - \frac{R_a}{(K_a \Phi)^2} T \quad (3.1)$$



Bu sebeple, DC motor rotorun hızı, alanın zayıflaması sonucunda artacaktır. Rotor hızının artmasında durumunda ise torkun bir kısmı harcanacaktır. Çünkü motorun torku direk olarak alan ile orantılıdır.

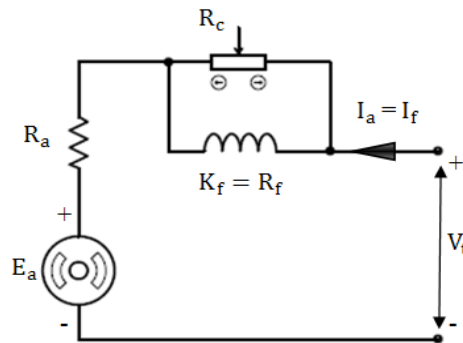
Bir şönt motorda alan sargılarına değişken bir direncin seri bağlanması sonucunda alan zayıflatılır. Şekil 3.1’de şönt motorda alan kontrolü devresi gösterilmiştir.



Şekil 3.1 Şönt motorda alan kontrolü

Alan kontrolünü değişken bir direnç ile sağlamak oldukça ucuza gerçekleştirilir. Bunun sebebi ise kontrolün düşük güç kullanılan kısımda (alan devresinde) yapılmasıdır. Ancak, alan devresindeki büyük indüktans nedeniyle direnç değişimine cevap yavaş olarak gerçekleşir.

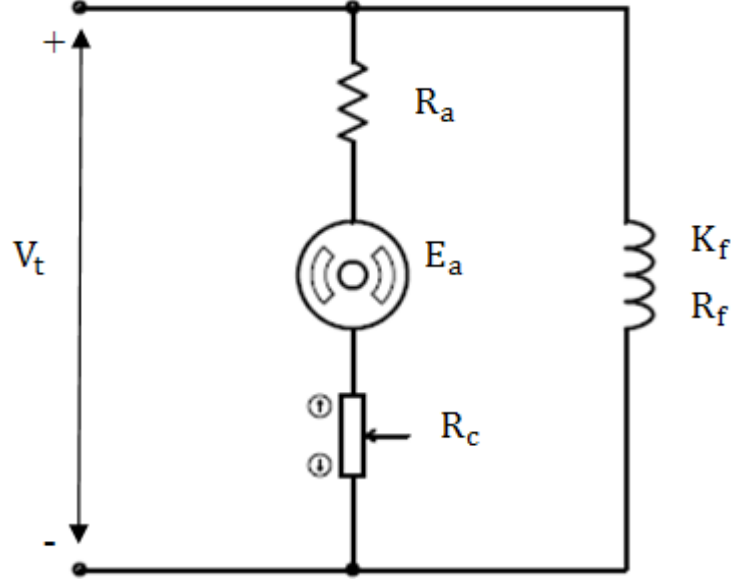
Seri motorlarda, alan akımının bir kısmının alan sargılarından bölünmesi ile alan zayıflatma etkisi gözlemlenir. Bu durumda kontrol aygıtına alan saptırıcısı denir. Şekil 3.2’de seri motorlarda alan kontrolü devresi gösterilmiştir.



Şekil 3.2 Seri motorlarda alan kontrolü

### 3.2.2 Endüvi Devresi Direnç Kontrolü ile DC Motor Hız Kontrolü

Endüvi devresine deęişken dirençler seri olarak bağlanarak motor hızı kolayca kontrol edilebilir. Şekil 3.3’de endüvi devresi direnci ile hız kontrolü devresi gösterilmiştir.



Şekil 3.3 Endüvi direnci ile hız kontrolü

Bu yöntem daha çok DC motorlara kademeli olarak yol vermek için kullanılır. Uygulamada kullanılan direnç sürekli olarak kullanımda kalırsa direnç üzerinde önemli ölçüde güç kaybı meydana gelecektir. Bunun sebebi endüvi devresinde yüksek akım çekilmesi ve direncin deęeri ile orantılı enerjinin ısıya dönüşmesidir.

DC motor rotorunun hızı denklem 3.1’de gösterilmiştir. Denklemde gösterilen endüvi direnci ( $R_a$ ) deęeri seri bağlanan deęişken direnç ( $R_c$ ) ile artar ve rotorun hızı denklem 3.2’de gösterildięi gibi deęişir.

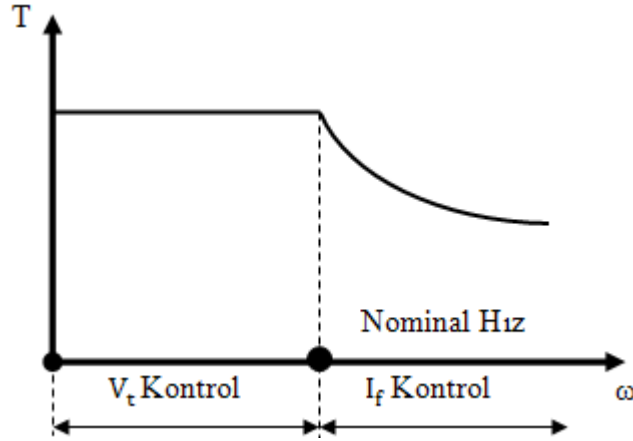
$$\omega_m = \frac{V_t}{K_a \Phi} - \frac{(R_a + R_c)}{(K_a \Phi)^2} T \quad (3.2)$$

### 3.2.3 Armatür Voltajı Kontrolü ile DC Motor Hız Kontrolü

Hız kontrolünde en sık kullanılan yöntem armatür voltajının direk kontrol edilmesi yöntemidir.

$$\omega_m = \frac{V_t}{K_a \Phi} - \frac{R_a}{(K_a \Phi)^2} T \quad (3.3)$$

Armatür voltajı ( $V_t$ ) 0 ile nominal değeri arasında değişirken, rotorun hızı da 0 ile nominal hız değeri arasında değişir. Armatür voltajı kontrolü ile alan kontrolü (alan akımı kontrolü) beraber kullanılarak, motorun hızı 0 ile nominal değerinin birkaç katı aralığında kontrol edilebilir.

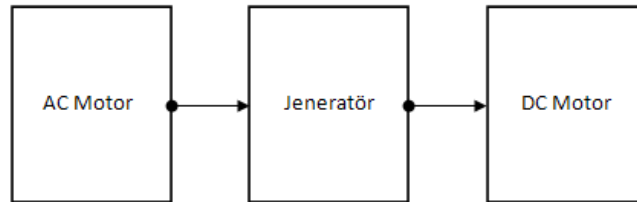


Şekil 3.4 Armatür voltaj kontrolü, alan kontrolü ile motor hız kontrolü.

Bu yöntem, yüksek akım çekmeleri sebebi ile güçlü DC motorlarda kullanılması elverişsiz olduğundan kullanımı küçük motorlar ile sınırlı kalmıştır. Aynı etki, bir diğer yöntem olan Ward-Leonard motor kontrol sistemi ile daha büyük motorlarda elde edilir.

#### 3.2.4 Ward-Leonard Sistemi ile DC Motor Hız Kontrolü

Ward-Leonard motor kontrol sistemi 1891 yılında takdim edilmiştir. DC motor hızının geniş sınırlar arasında ayarlanmasını sağlamak için kullanılır. Şekil 3.5'de sistemin işleyiş sırası şematik olarak gösterilmiştir.



Şekil 3.5 Ward-Leonard Sistemi

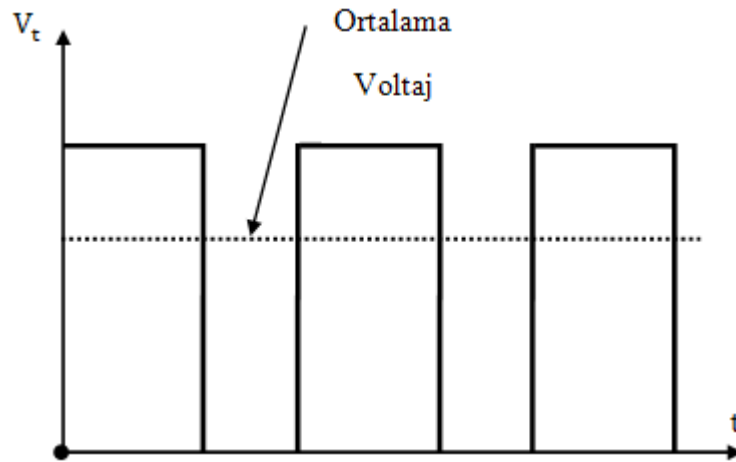
Bir AC motor jeneratörü tahrik eder, jeneratör de DC motorun armatür voltajını üretir. Jeneratörün alan akımının değiştirilmesi ile armatür voltajı ayarlanabilir.

Ward-Leonard sistemi verim kaybı yaşanmaksızın çok düzgün ve güvenilir hız kontrolü sağlar. Fakat oldukça karmaşık sistemlerdir. Bu nedenle sadece yüksek kalitenin gerekli olduğu maliyeti yüksek uygulamalarda kullanılmaları uygundur.

### 3.2.5 Yarı İletkenler Kullanılarak DC Motor Hız Kontrolü

Güçlü silikon diyotlar ve tristörlerin geliştirilmesi ile alternatif akımın doğrultularak DC motor kontrol devreleri için yeni bir kontrol metodu ortaya çıkmıştır. Bu aygıtlar gitgide motor jeneratör takımlarından meydana gelen Ward Leonard sisteminin yerini almaktadır.

Kontrollü doğrultmaçlar Silicon Controlled Rectifiers veya SCRs olarak bilinirler. Bu doğrultmaçlar, besleme gerilimini kıyarak darbe zinciri şeklinde armatür voltajı üretirler. Şekil 3.6'da SCRs tarafından üretilen darbe zinciri gösterilmiştir.



Şekil 3.6 Darbe Zinciri

Darbe zincirinin frekansı değiştirilerek, armatür voltajının ortalama değeri ayarlanır. Darbelerin şekli her zaman dörtgen değildir. Örneğin, bir DC motor için gerekli olan doğrultulmuş uyarım, SCR tarafından doğranmış AC sinyali (yarım sinüs dalgası biçimde) ile sağlanır.

Standart gerilim-hız-trok ilişkisi ortalama değerler cinsinden denklem 3.4'de gösterilmiştir.

$$V_t = E_a + I_a R_a$$

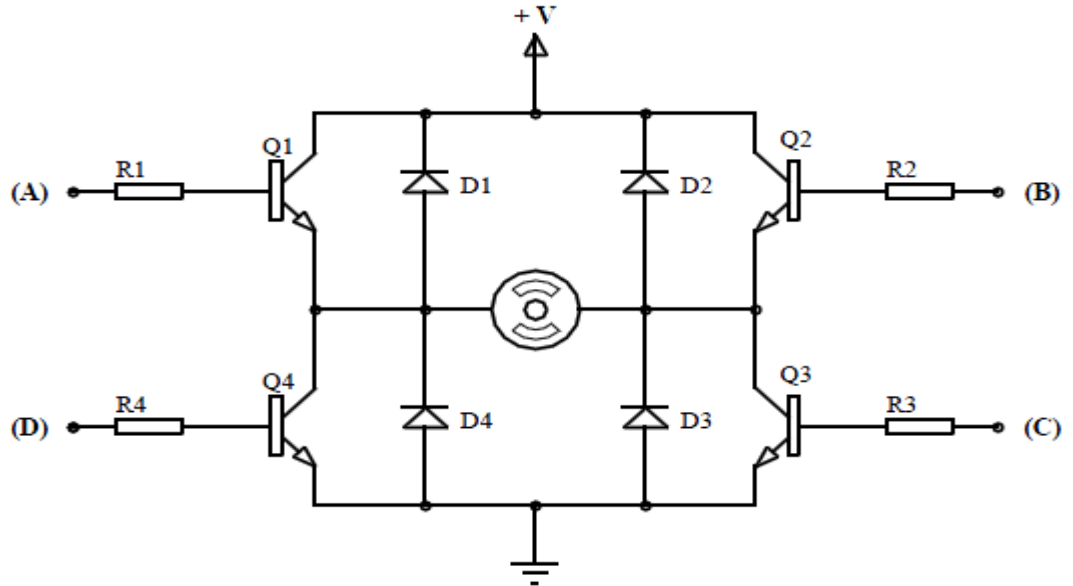
$$E_a = K_a \Phi \omega_m$$

$$T = K_a \Phi I_a$$

(3.4)

### 3.2.6 Tam Köprü (H Köprüsü) ile Yön Kontrolü

DC motorların yön kontrolünün sağlanabilmesi için Tam Köprü denilen bir yöntem geliştirilmiştir. Tam Köprü, genellikle 4 adet transistör ve diyotlar ile gerçekleştirilen ve motorun iki yönlü dönebilmesini sağlayan bir yöntemdir. Şekle 3.7'de bir tam köprü gösterilmiştir.



Şekil 3.7 Tam Köprü devresi

Şekil 3.7'deki devrede, A=1, B=0, C=1, D=0 yapıldığında motor saat ibresi yönünde dönecektir. Tersi durumda, A=0, B=1, C=0, D=1 yapıldığında motor saat ibresinin tersi yönünde dönecektir. A=1, B=1, C=0, D=0 ve A=0, B=0, C=1, D=1 durumlarında ise motor fren yapacaktır. A=1, B=0, C=0, D=1 ve A=0, B=1, C=1, D=0 olduğu durumlarda ise kısa devre gerçekleşecektir. Tablo 3.1'de tam köprün fonksiyonu verilmiştir.

Motorda açma ve kapama anlarında indüklenerek gerilim çoğu zaman besleme geriliminin bile üzerine çıkar. Böyle bir durumda transistörlerin üzerinden ters akım geçecek ve transistörler yanacaktır. Açma ve kapama anlarında motorda üretilecek

gerilimden transistörleri korumak için Şekil 3.7’de gösterildiği gibi devreye diyotlar bağlanır.

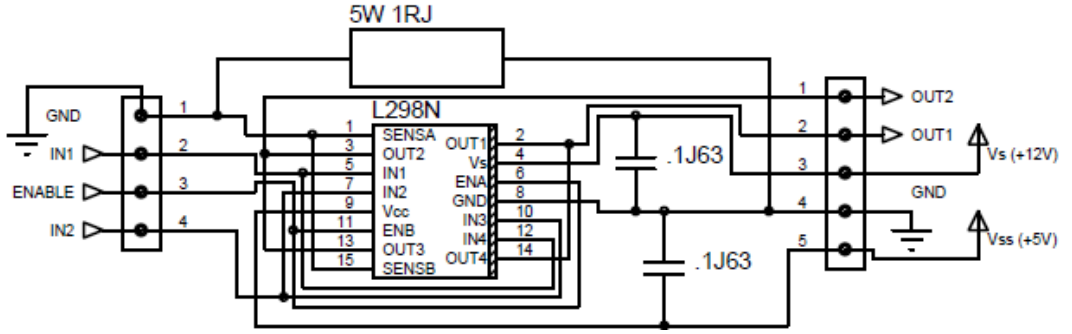
Tablo 3.1 Tam Köprü fonksiyonu

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>FONKSİYON</b>
1	0	1	0	Saat ibresi
0	1	0	1	Saat ibresi tersi
1	1	0	0	Fren
0	0	1	1	Fren
1	0	0	1	Kısa devre
0	1	1	0	Kısa devre

## BÖLÜM DÖRT KONTROL DONANIMI

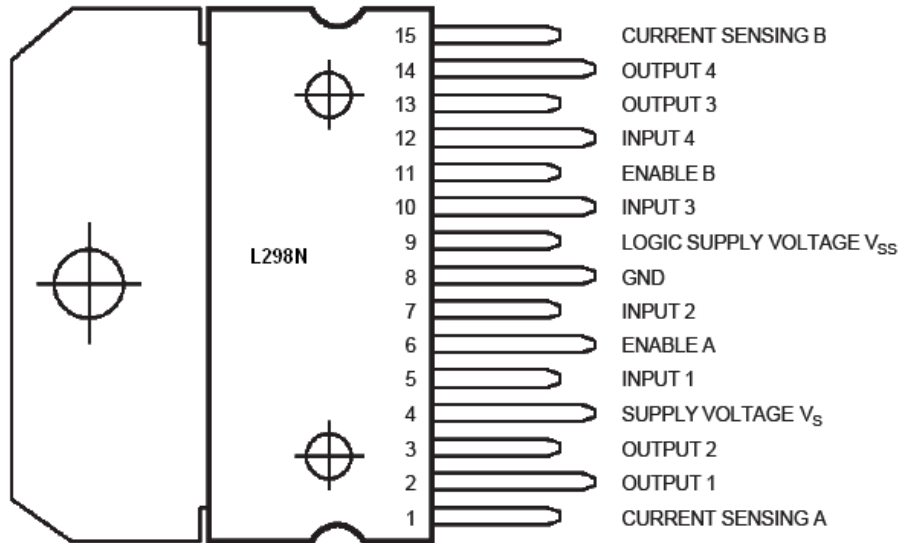
### 4.1 DC Motorların Sürücü Devresi

Robot kollarımızın konumunu veri toplama kartı yardımı ile kontrol edebilmemiz için DC motor sürücü devresi tasarlanmıştır. Tasarlanan devre şekil 4.1'de gösterilmektedir.



Şekil 4.1 DC motor sürücü devresi

Devre tasarımında, bir adet 5W 1Ω direnç, iki adet 100nF kondansatör ve bir adet L298N entegresi kullanılmıştır. L298N entegresi, röleler, selenoidler, DC ve step motorlar gibi indüktif yükleri sürmek için tasarlanmış çift tam köprü içeren tek parça bir entegredir. Bu entegre, iki adet enable girdisi sayesinde diğer girdi sinyallerinden bağımsız olarak etkin veya kullanım dışı kılınabilir.



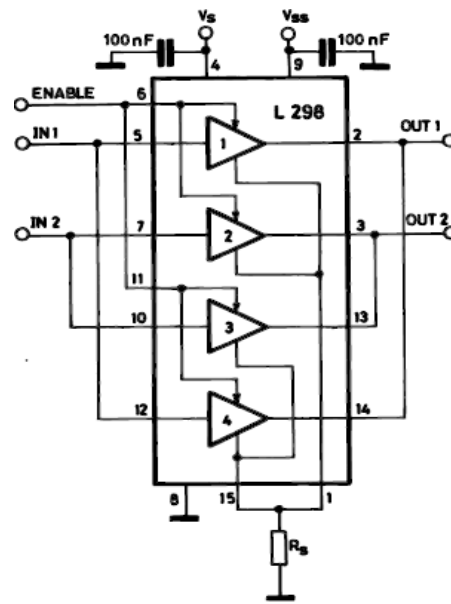
Şekil 4.2 L298N

Devreye besleme ( $V_s$ ) ve lojik besleme ( $V_{ss}$ ) gerilimi uygulanması ve iki köprünün herhangi birinin çıkış pinlerine motorun bağlanması ile sürücü devremiz kullanıma hazır duruma gelir. Lojik sinyallerden, input1 ve enable yüksek, input2 düşük ise gerilim output1'den output2'ye akar ve motor saat ibresinin tersi yönünde döner. Eğer input2 ve enable yüksek, input1 düşük ise gerilim output2'den output1'e doğru akar ve motor saat ibresi yönünde döner. Input1, input2 ve enable yüksek ise motor frenler. Input1, input2 ve enable düşük ise motor serbest (boşta) kalır. Tablo 4.1'de sürücü devre fonksiyonu verilmiştir.

Tablo 4.1 Sürücü devre fonksiyonu

INPUT1	ENABLE	INPUT2	FONKSİYON
1	1	0	Saat ibresi tersi
0	1	1	Saat ibresi
1	1	1	Fren
0	0	0	Serbest

L298N entegresindeki tam köprülerin her biri en fazla 2 amper (DC) değerindeki yükleri taşıyabilmektedir. Bu nedenle motorların kalkınma akımlarının yüksekliği dikkate alınarak entegre içindeki köprüler paralel şekilde birleştirilmiştir. Böylece entegre daha yüksek akımları taşıyabilir duruma gelmiştir. Şekil 4.3'de L298N paralel düzenleşimi gösterilmiştir.



Şekil 4.3 L298N paralel düzenleşim

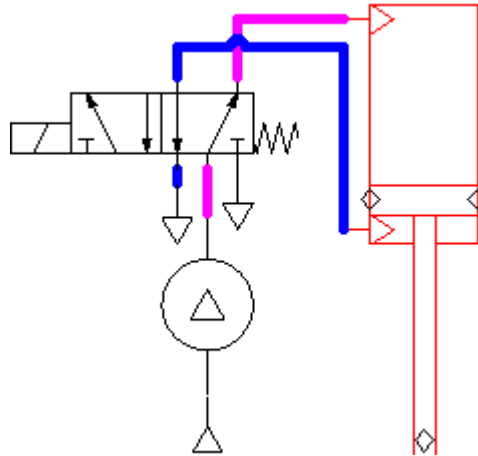


Entegrenin paralel düzenleşimi için, input1 ve input4 pinleri birleştirilerek yeni input1, input2 ve input3 pinleri birleştirilerek yeni input2, output1 ve output4 pinleri birleştirilerek yeni output1, output2 ve output3 pinleri birleştirilerek yeni output2, enable A ve enable B pinleri birleştirilerek yeni enable pinleri elde edilir.

#### 4.2 Silindir ve Tutucunun Sürücü Devresi

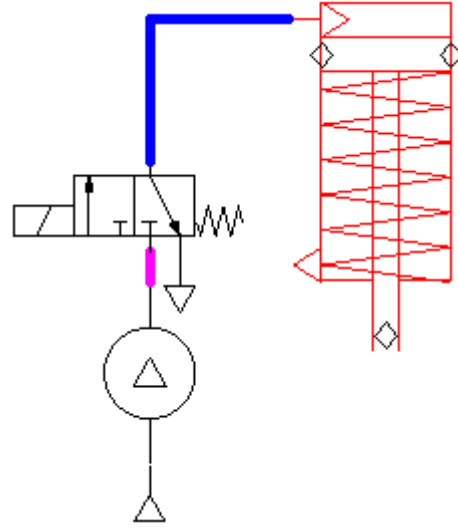
Bu başlık altında, silindir ve tutucunun hareketini sağlayan yön kontrol valfleri ve valfler ile veri toplama kartı arasındaki iletişimi sağlayan sürücü devresine değinilmiştir.

Scara robot manipulatörün Z eksenindeki hareketini sağlamak için çift etkili bir silindir kullanılmıştır. Bu silindiri yönlendirmek için 5/2 selenoid tetikli yay geri dönüşlü normalde kapalı bir pnömatrik valf kullanılmıştır. Valf (DC) 12V'luk bir selenoid ile tetiklenmektedir. Şekil 4.4'de silindirin pnömatrik devresi verilmiştir.



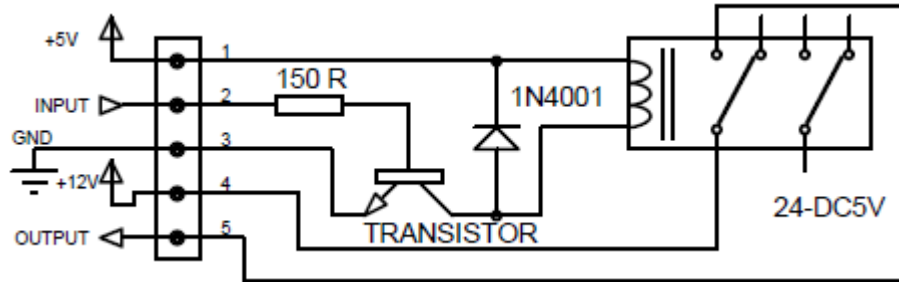
Şekil 4.4 Silindirin pnömatrik devresi

Scara robot manipulatörün tutma işlemini gerçekleştiren tutucu basınçlı hava verildiğinde açılmakta, basınçlı hava gönderilmediğinde ise kapanmaktadır. Bu özelliği ile tek etkili yay geri dönüşlü silindir ile eşdeğer çalışma şekline sahiptir. Tutucunun açma – kapama işlemini gerçekleştirebilmesi için 3/2 selenoid tetikli yay geri dönüşlü normalde kapalı bir yön kontrol valfi kullanılmıştır. Valf, (DC) 12V'luk bir selenoid ile tetiklenmektedir. Şekil 4.5'de tutucu eş değer çalışma şekline sahip tek etkili yay geri dönüşlü silindir ile gösterilmiştir.



Şekil 4.5 Tutucunun pnömatik devresi

Silindiri ve tutucuyu yönlendiren yön kontrol valfleri ile veri toplama kartı arasında iletişimi sağlayacak sürücü devreler tasarlanmıştır. Bunun sebebi yön kontrol valflerinin tetiklenmesi için gerekli olan gerilim değerinin veri toplama kartı tarafından sağlanamamasıdır.



Şekil 4.6 Silindir ve Tutucunun tetik devresi

Veri toplama kartının +5V çıkış gerilimine sahip olması ve yön kontrol valflerinin +12V'luk selenoidler ile tetiklenmesi nedeniyle bobin gerilimi +5V olan röleye ihtiyaç duyulmuştur. Ancak bağlantı kablosundaki kayıplar nedeni ile veri toplama kartının çıkış gerilimi rölenin tetiklenmesi için yetersiz kalmaktadır. Bu sorunun üstesinden gelmek için transistör kullanılmıştır. Şekil 4.6'daki INPUT veri toplama kartına gelen girdiyi, OUTPUT ise yön kontrol valfini tetiklemek için gerekli çıktıyı temsil etmektedir.

Devre tasarımında, bir adet 150Ω direnç, bir adet diyot, bir adet transistör ve bir adet röle kullanılmıştır.

### 4.3 Veri Toplama Kartı

National Instruments tarafından üretilen NI USB-6008, sekiz adet analog giriş, iki adet analog çıkış, oniki adet dijital giriş/çıkış kanalına sahiptir. USB arabirimli veri toplama ve kontrol aracıdır. Şekil 4.7’de NI USB-6008 gösterilmiştir.

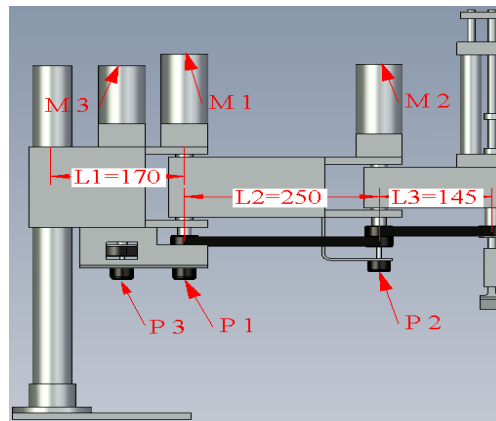
DC motor sürücü devrelerini tetiklemek için dokuz adet dijital giriş/çıkış kanalı, silindirik ve tutucu sürücü devrelerini tetiklemek için iki adet analog çıkış kanalı, DC motorların konum bilgilerinin tespiti için kullanılan potansiyometreler için üç adet analog giriş kanalı kullanılmıştır.



Şekil 4.7 NI USB-6008

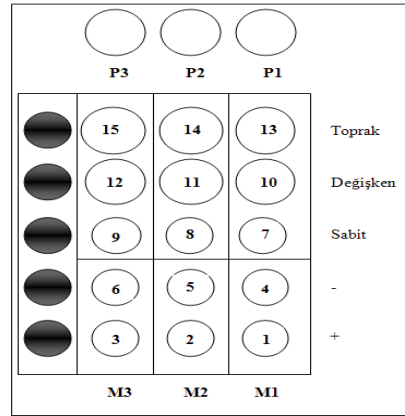
### 4.4 Konektör ve Genel Bağlantı Şeması

Scara robot manipulatör üzerindeki üç adet DC motor ve bu motorlar ile bağlantılı çalışan potansiyometrelerin robot üzerindeki yerleşimi Şekil 4.8’de gösterilmiştir.



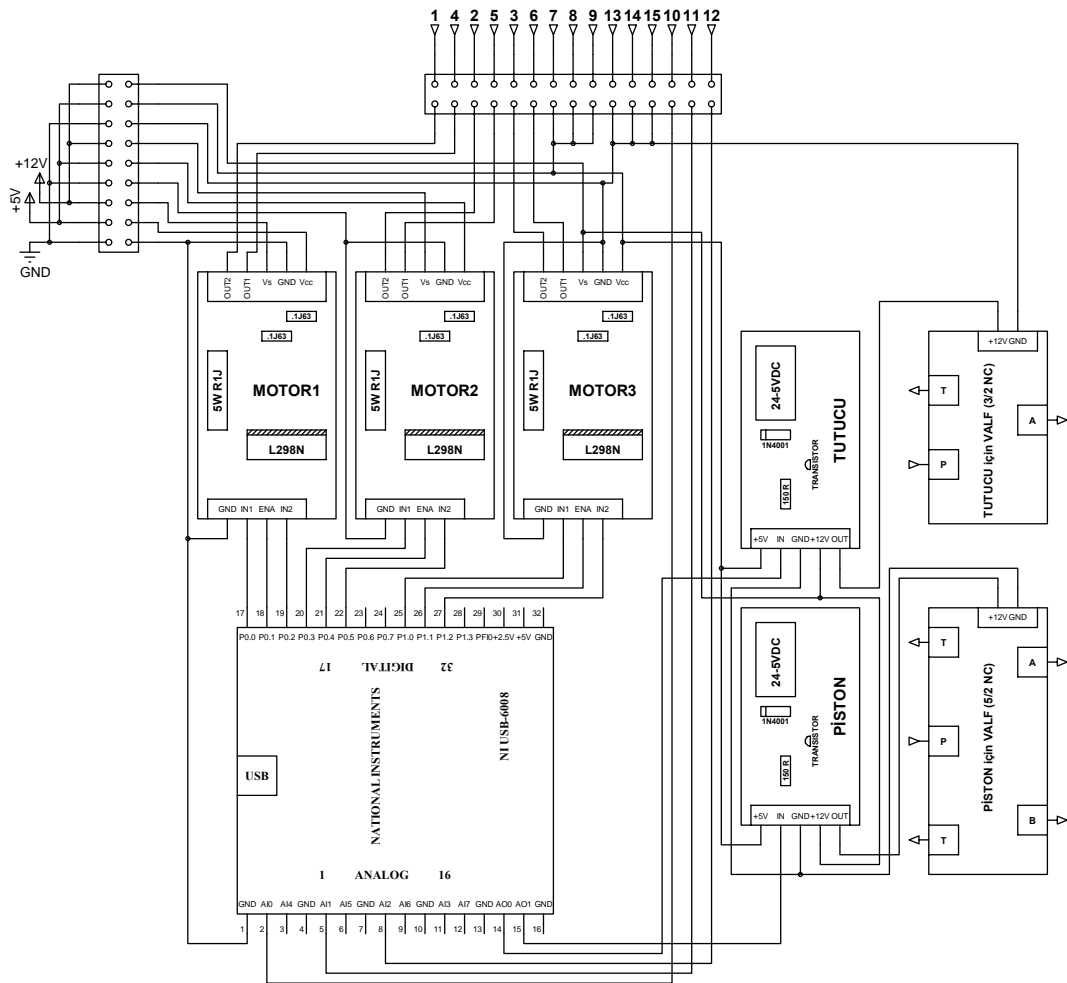
Şekil 4.8 DC motorlar ve potansiyometreler

DC motorların ve potansiyometrelerin bağlantı uçlarının konektör üzerindeki yerleşimi Şekil 4.9’da gösterilmiştir.



Şekil 4.9 Konektör

Sürücü devreleri, konektör ve veri toplama kartı arasındaki genel bağlantı şeması Şekil 4.10'da gösterilmiştir.



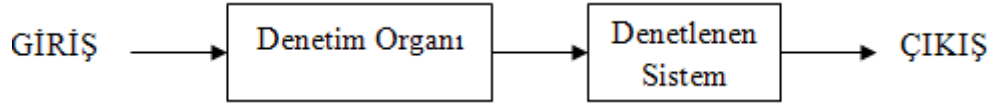
Şekil 4.10 Genel bağlantı şeması

## BÖLÜM BEŞ ROBOT KONTROLÜ

### 5.1 Temel Kontrol Şekilleri

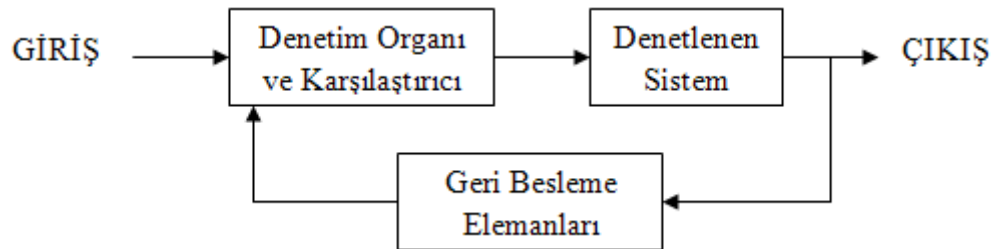
Kontrolün amacı, işlemdeki değişkenlerin hedeflenen şekilde davranması için giriş sinyali üretmektir. Başka bir ifade ile sisteme uygun bir giriş uygulayarak, arzu edilen çıkışı sağlamaktır. Kontrol sistemleri, açık ve kapalı çevrimli kontrol sistemleri olmak üzere ikiye ayrılır.

Kontrol çevriminde kumanda, sistemin çıkışlarına bağlı olarak belirlenmiyor ise bu kontrol sistemine açık çevrimli kontrol sistemi denir. Diğer bir deyişle, çıkışlar kumandayı etkilememektedir. Bu tür kontrol genellikle sistem yapısının ve sistem girişlerinin çok iyi bilindiği uygulamalarda kullanılır. Şekil 5.1'de açık çevrimli kontrol diyagramı gösterilmiştir.



Şekil 5.1 Açık çevrimli kontrol

Sistemin referansı ile gerçek çıkış arasındaki farka göre giriş uygulanır ise bu kontrol çevrimine kapalı çevrimli kontrol denir. Sistem çıkışındaki değişmeler, sisteme uygulanacak kumandanın belirlenmesi için geri gönderilir. Bu sebeple kapalı çevrimli kontrole, geri beslemeli kontrol de denir. Şekil 5.2'de kapalı çevrimli kontrol diyagramı gösterilmiştir.

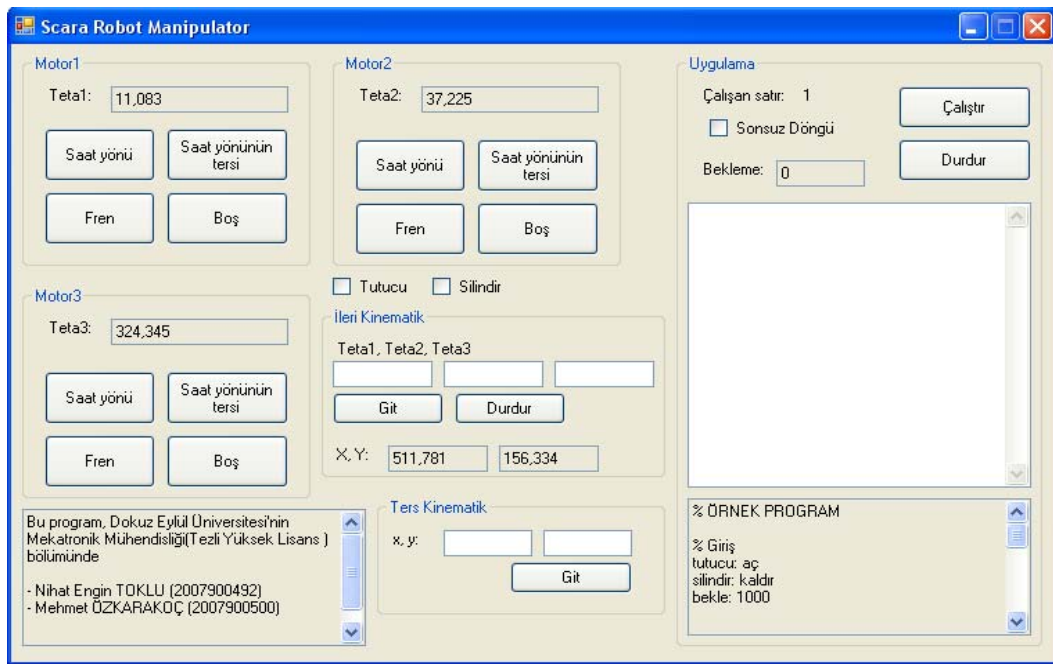


Şekil 5.2 Kapalı çevrimli kontrol

Konum kontrolünün gerçekleştirilebilmesi için sisteme (Scara robot manipulatör) kapalı çevrimli kontrol uygulanmıştır.

## 5.2 Kontrol Programı

Visual Studio programı ile yazılan kontrol programında seçme (tutma) ve yerleştirme uygulaması gerçekleştirilmektedir. Seçme ve yerleştirme uygulaması, herhangi bir koordinatta bulunan bir parçanın bulunduğu koordinattan alınıp, hedef koordinata götürülüp bırakılması işlemidir. Bu uygulama sadece parça taşımada değil, montaj gibi uygulamalarda da kullanılmaktadır. Şekil 5.3’de kontrol programı ekran görüntüsü gösterilmiştir.



Şekil 5.3 Scara robot manipulatörün kontrol programı

Kontrol programında, Motor1, Motor2 ve Motor3 kutucuklarındaki ilgili butonlara basarak motorları birbirinden bağımsız olarak saat ibresi yönünde, saat ibresinin tersi yönünde döndürebiliriz, frenletebiliriz, serbest bırakabiliriz. Tutucunun kapanmasını, silindirin +Z yönünde hareket etmesini ilgili kutucukları işaretleyerek sağlayabiliriz.

İleri Kinematik kısmında Teta1, Teta2 ve Teta3 açılarının değerlerini derece cinsinden programa girerek motorların girilen değerlere ulaşmaya kadar dönmesi sağlanır ve X, Y kutucuklarında da uç işlevcinin ulaştığı milimetre cinsinden koordinatı hesaplatırız. İleri Kinematik kısmında, programa tüm açı değerlerini girmemize gerek bulunmamaktadır.

Sadece girdimiz açı değerlerine, ilgili motorlar ulaşmaya çalışırken, işlem yapmadığımız motor durumunu muhafaza edecektir. Eğer sisteme girdiğimiz açı değerleri robotun çalışma alanı dışında kalıyorsa, program “Erişilemez nokta!” hatası verecek ve robot hareket etmeyecektir.

Ters kinematik kısmında, hedef koordinatımızı programa gireriz, program bu noktaya ait eklem değişkenlerini (açı değerleri) Newton-Raphson metodu ile hesaplar. Hesaplanan açı değerleri çalışma alanı içerisinde ise motorlar uç işlevci hedefe ulaşana kadar sürülür. Eğer hesaplanan açı değerleri çalışma alanı dışında kalıyorsa, program “Erişilemez nokta!” hatasını verir ve motorları tahrik etmez.

Uygulama kısmında, örnek programda belirtilen komutlar yardımı ile istediğimiz tutma yerleştirme işlemi gerçekleştirebiliriz. Bu uygulamanın tekrarlı olması istenir ise “Sonsuz Döngü” kutucuğu işaretlenir. Tablo 5.1’de uygulama kısmı komutları verilmiştir.

Tablo 5.1 Uygulama kısmı komutları

<b>KOMUT</b>	<b>FONKSİYON</b>
tutucu:aç	Tutucu açılır.
tutucu:kapa	Tutucu kapanır.
silindir:kaldır	Silindirin rodu +Z yönünde hareket eder ve en üst konuma ulaşır.
silindir:indir	Silindirin rodu -Z yönünde hareket eder ve en alt konuma ulaşır.
açılara_git:60;-60	Motorlar belirtilen açılara ulaşır.
noktaya_git:295;361	Uç işlevci, koordinatı belirtilen noktaya ulaşır.
bekle:1000	Hareket, milisaniye cinsinden süre kadar durur.

Kontrol programı motorların hareket miktarını, geri besleme elemanları potansiyometreler aracılığı ile 10 ms aralıklar ile takip eder. Potansiyometrelerden

elde edilen veriler program tarafından denklem 5.1’de yerine konarak eklem açıları derece cinsinden hesaplanır.

$$\text{Anlık Açı Değeri} = \frac{(\text{Referans Gerilim} - \text{Anlık Gerilim})}{\left(\frac{\text{Maksimum Gerilim}}{\text{Toplam Açı Miktarı}}\right)} \quad (5.1)$$

Denklem 5.1’deki “Referans Gerilim” değeri motorun tahrik ettiği kolun X eksenine ile yaptığı açı 0° iken, ilgili motora bağlı potansiyometrenin sabit bacağına +5V uygulandığında, değişken bacağından okuduğumuz gerilim değeridir. Sırası ile 1, 2 ve 3 numaralı motorlara bağlı potansiyometrelerin “Referans Gerilimleri”, 1,58, 1,94 ve 5V’dur. “Maksimum Gerilim”, potansiyometrenin sabit bacağına +5V uyguladığımızda fiziksel sınırlamalar sebebi ile elde edilebilecek en büyük gerilim miktarıdır. Sırası ile 3,08, 3,58 ve 5V’dur. “Toplam Açı Miktarı”, motorların saat ibresi ve saat ibresi tersi yönünde çalışabildikleri alanın tamamıdır. Sırası ile 192°, 250° ve 450°’dir.



## BÖLÜM ALTI

### SONUÇ

Scara robot manipulatörün DC motorlarını, silindirini ve tutucusunu hareket ettirebilmek için sürücü devreler tasarlanmıştır. Uç işlevcinin hedeflenen noktaya ulaşması için kontrol programı yazılmış, program ile sürücü devreler arasında irtibatı sağlamak için uygun veri toplama kartı seçilmiştir.

Yazılan kontrol programında hesaplanan eklem değişkenlerinin değerleri ve uç işlevcinin koordinatları, Matlab' de yazılan ileri ve ters kinematik döngülerinde elde edilen sonuçlar ile uyumlu olduğu gözlemlenmiştir.

Robotun yaptığı hareketler sonrasında eklem değişkenlerinin değerlerinin, kontrol programında hesap edilen değerlere göre  $\pm 2$  derecelik bir farklılık gösterdiği gözlemlenmiştir. Bu farklılığın sebebi, kontrol elemanları ve yazılım ile ilgili değil, motorlara bağlı dişlilerin arasındaki boşluk miktarının fazlalığıdır.

## KAYNAKLAR

- Bingül, Z. ve Küçük, S. (2005) *Robot tekniği I*. İstanbul: Birsen Yayınevi.
- Braga, N. C. (2005). *Robotik mekatronik ve yapay zeka*. (H. Ünaydın ve A. Süer, Çev.). İstanbul: Bileşim Yayınevi. (Orijinal basım tarihi 2003)
- Duffy, J. (1980). Position analysis of planar chains. In *Planar mechanisms and manipulators* (17). London: Edward Arnold Ltd.
- Kreyszig, E. (2006). Newton's method for solving equations  $f(x) = 0$ . In *Advanced engineering mathematics* (9th ed.) (790 - 792). Singapore: John Wiley & Sons, Inc.
- L298 dual full-bridge driver. (n.d.). Retrieved October 22, 2007, from <http://www.datasheetcatalog.org/datasheet/stmicroelectronics/1773.pdf>
- Murray, R. M., Li, Z., & Sastry, S. S. (1994). Manipulator jacobian. In *A mathematical introduction to robotic manipulation* (115). NY: CRC Press.
- Ogata, K. (2002). Closed loop control versus open loop control. In *Modern control engineering* (6 - 7). Minnesota: Prentice Hall.
- Özdem, N. (1968). *Elektrotekniğin esasları*. İzmir: Doğruluk Matbaası.
- Sciavicco, L., & Siciliano, B. (2005). Kinematic singularities. In *Modeling and control of robot manipulators* (2nd ed.) (89 - 94). Glasgow: Springer.
- Spong, M. W., Hutchinson, S., & Vidyasager, M. (2006). Actuator Dynamics. In *Robot modeling and control* (205 - 208). NY: John Wiley & Sons, Inc.
- Tsai, L. W. (1999). Denavit- Hartenbeg homogeneous transformation matrices. In *Robot analysis* (60 - 65). NY: John Wiley & Sons, Inc.

## EKLER

### EK 1

```

clc;clear;
% teta1,teta2,teta3 girdileri derece cinsinden, h1, l1, l2, l3, d4 mm cinsinden!!!
%tet2,tet2,tet3 radyana çevrilmiş halleri.
syms h1 l1 l2 l3 teta1 teta2 teta3 d4 tet1 tet2 tet3
teta1=-5.0106;
teta2=34.7076;
teta3=0;
h1=75;
l1=170;
l2=250;
l3=145;
d4=75;
tet1=teta1*pi/180;
tet2=teta2*pi/180;
tet3=teta3*pi/180;
T0=[cos(tet1) -sin(tet1) 0 l1
    sin(tet1) cos(tet1) 0 0
    0 0 1 h1
    0 0 0 1];
T1=[cos(tet2) -sin(tet2) 0 l2
    sin(tet2) cos(tet2) 0 0
    0 0 1 0
    0 0 0 1];
T2=[cos(tet3) -sin(tet3) 0 l3
    sin(tet3) cos(tet3) 0 0
    0 0 1 0
    0 0 0 1];
T3=[1 0 0 0
    0 1 0 0

```

```
    0 0 1 -d4  
    0 0 0 1];  
T4=T0*T1*T2*T3  
%T4=simple(T0*T1*T2*T3)  
px=T4(1,4)  
py=T4(2,4)  
pz=T4(3,4)
```

**EK 2**

```

clc;clear;
syms h1 l1 l2 l3 teta1 teta2 teta3 d4 tet1 tet2 tet3 r11 r12 r13 r21 r22 r23 r31 r32 r33
px py pz
l1=170;
l2=250;
l3=145;
%teta1=tet1*pi/180;
%teta2=tet2*pi/180;
%teta3=tet3*pi/180;
T0=[cos(teta1) -sin(teta1) 0 l1
    sin(teta1) cos(teta1) 0 0
    0 0 1 h1
    0 0 0 1];
T1=[cos(teta2) -sin(teta2) 0 l2
    sin(teta2) cos(teta2) 0 0
    0 0 1 0
    0 0 0 1];
T2=[cos(teta3) -sin(teta3) 0 l3
    sin(teta3) cos(teta3) 0 0
    0 0 1 0
    0 0 0 1];
T3=[1 0 0 0
    0 1 0 0
    0 0 1 -d4
    0 0 0 1];
T4=[r11 r12 r13 px
    r21 r22 r23 py
    r31 r32 r33 pz
    0 0 0 1];

```

$T_4 = T_0 \cdot T_1 \cdot T_2 \cdot T_3$  eşitliğinde  $T_0$  evriği alınıp,  $T_4$  ile çarpılır. Yeni eşitliğin sol tarafına "sol", sağ tarafına "sag" ismi verilmiştir.

```
%sol=simple(inv(T0)*T4)
```

```
%sag=simple(T1*T2*T3)
```

%Oluşan eşitliğin sol tarafında kalan matrisin 1.satır 4.sütun elemanın karesi alınıp, %2.satır 4.sütun elemanın karesine eklenmiş, sağ taraftaki matrisin 1.satır 4.sütun elemanın karesi alınıp, 2.satır 4.sütun elemanın karesine eklenmiş elde edilen ifadeler(eşitliğin sol tarafında kalan matris ile sağ tarafında matris) birbirine eklenmiştir.

```
%sag12=(sag(1,4))^2
```

```
%sag22=(sag(2,4))^2
```

```
%sag2=simple(sag12+sag22)
```

```
%sol12=(sol(1,4))^2
```

```
%sol22=(sol(2,4))^2
```

```
%sol2=simple(sol12+sol22)
```

%teta1 ve teta2 nin hesaplanması

```
px=545;
```

```
py=50;
```

```
a=(px^2-2*px*11+py^2+11^2-12^2-13^2)/(2*13*12);
```

```
tet2=atan(-sqrt(1-a^2)*(1/a));
```

```
teta2=tet2*(180/pi)
```

```
teta1=(atan(py/(px-11))+atan(-(sqrt(py^2+(px-11)^2-  
(12+13*cos(tet2))^2))*(1/(12+13*cos(tet2))))*(180/pi)
```

```
%ilk=(atan(py/(px-11)))*(180/pi)
```

```
%son=(atan((sqrt(py^2+(px-11)^2-  
(12+13*cos(tet2))^2))*(1/(12+13*cos(tet2))))*(180/pi)
```

**EK 3**

**Public Function** Calculate() **As Double**

**Dim** teta1, teta2, K1, K2, h **As Double**

**Dim** F1, F2, v1, v2, vv1, vv2, detj **As Double**

**Dim** donaded **As Integer**

**Dim** L1, L2, L3 **As Double**

L1 = KolUzunluk1

L2 = KolUzunluk2

L3 = KolUzunluk3

teta1 = 3.3

teta2 = 0.1

h = 0.00000001

donaded = 0

**Dim** j(2, 2), jek(2, 2)

**Do**

K1 = teta1

K2 = teta2

F1 = L1 + L2 \* Math.Cos(teta1) + L3 \* Math.Cos(teta1 + teta2) - px

F2 = L2 \* Math.Sin(teta1) + L3 \* Math.Sin(teta1 + teta2) - py

j(1, 1) = -L2 \* Math.Sin(teta1) - L3 \* Math.Sin(teta1 + teta2)

j(1, 2) = -L3 \* Math.Sin(teta1 + teta2)

j(2, 1) = L2 \* Math.Cos(teta1) + L3 \* Math.Cos(teta1 + teta2)

j(2, 2) = L3 \* Math.Cos(teta1 + teta2)

v1 = -F1

v2 = -F2

**For** z = 1 **To** 2

**For** k = 1 **To** 2

jek(z, k) = j(z, k)

**Next** k

**Next** z

j(1, 1) = jek(2, 2)

j(2, 2) = jek(1, 1)

```
j(1, 2) = -jek(1, 2)
j(2, 1) = -jek(2, 1)
detj = jek(1, 1) * jek(2, 2) - jek(1, 2) * jek(2, 1)
For z = 1 To 2
For k = 1 To 2
j(z, k) = j(z, k) / detj
Next k
Next z
vv1 = j(1, 1) * v1 + j(1, 2) * v2
vv2 = j(2, 1) * v1 + j(2, 2) * v2
teta1 = teta1 + vv1
teta2 = teta2 + vv2
If Math.Abs(K1 - teta1) <= h And Math.Abs(K2 - teta2) <= h Then Exit Do
donaded = donaded + 1
If donaded > 700000 Then
Exit Do
End If
Loop
teta1_degrees = Rad2Deg(teta1) Mod 360
teta2_degrees = Rad2Deg(teta2) Mod 360

End Function
```