

T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİR KOLUN HAREKETLERİNİ TAKİP EDEN DÖRT DÖNEL
MAFSALLI ROBOT KOLU TASARIMI ve DENEYSEL ARAŞTIRILMASI

Özcan ÇETİNKAYA

YÜKSEK LİSANS TEZİ

MAKİNA MÜHENDİSLİĞİ ANA BİLİM DALI

Bu tez 20 Ocak 2009 tarihinde Aşağıdaki Jüri Tarafından Kabul Edilmiştir.

Yrd. Doç.Dr.
Hilmi KUŞÇU
Jüri Başkanı (Danışman)

Prof.Dr.-Ing.
Ahmet CAN
Üye

Prof.Dr.
Selçuk ARIN
Üye

ÖZET

Günümüzde bilgisayar sistemlerinin gelişimine paralel olarak bilgisayar destekli tasarım (C.A.D.) ve üretim (C.A.M.) sistemleri de gelişme göstermiştir. Bilgisayar destekli üretim sistemlerinde genellikle bir bilgisayar tarafından denetlenen bir veya birden fazla robot kolu (manipülâtör) ve bunlara bağlı uç işlevcileri kullanılmaktadır.

Bu çalışmada; Dört serbestlik derecesine sahip bir robot kolu ve buna bağlı bir uç işlevcisinin yine dört serbestlik derecesine sahip bir robot kolu ve bilgisayar ortamında hazırlanmış bir kontrol programı yardımıyla konum kontrolü ele alınmıştır. Bu amaçla kollardan birinin eklemlerine eklem açılarını algılaması için hassas potansiyometreler yerleştirilmiş ve Master Kol adı verilmiştir. Diğer kolun eklemlerine ise master kolun eklem açılarına ve kontrol programında sanal eklemlerde üretilen eklem açılarına göre hareket eden RC servo motorlar yerleştirilmiş ve Slave kol adı verilmiştir. Master ve Slave Kolun kontrol programı yani bilgisayarla olan bilgi alışverişi bilgisayara takılan ADC DAq kartı yardımıyla yapılmaktadır. Ayrıca Master koldan alınan eklem açılarına veya kontrol programında üretilen sanal eklem açılarına göre robot kolunun ileri kinematik hesaplamaları yapılarak uç işlevcisinin konum vektörleri bilgi amaçlı olarak programda gösterilir. Uç işlevcisinin olması istenilen üç boyutlu koordinat düzlemindeki x, y, z koordinatlarına göre Ters kinematik hesaplamaları yapılarak olası eklem açıları bilgi amaçlı olarak programda gösterilir.

Kontrol programından yapılan seçime göre Slave kol; Master kolun hareketlerini (Eklem Açılarını) birebir takip eder, Kontrol programındaki sanal potansiyometrelerin açılarını takip eder, Kontrol programında uç işlevcisinin üç boyutlu koordinat düzleminde olması istenen noktaya hareket eder ve Kontrol programında belirlenen koordinatlara belirlenen periyotla seri hareketler yapar.

Anahtar kelimeler: manipülâtör, robot kolu, konum kontrolü, ileri kinematik, ters kinematik

SUMMARY

Nowadays, computer assisted design and manufacturing systems have showed an improvement in parallel with the improvement of the computer systems. One or more manipulators which are mostly controlled by a computer and apex processors bounded in these manipulators are used in computer assisted manufacturing systems.

In this study, position control is taken on with the assistance of a manipulator which has four degrees of freedom and a bounded apex processor, also again a manipulator which has four degrees of freedom and a control programme which is setup with a computer. For this purpose, sensitive potentiometers are located in one of manipulator's joints in order to sense joint angles and that is called as master arm (master manipulator). In joints of the other manipulator, RC servo motors which move according to joint angles of master arm and prepared in control programs' virtual joints are located and named as slave arm. Control programme of master and slave arms (i.e. exchange of information with computer) are committed with the aid of ADC DAq card which is mounted to computer. Besides, advanced-kinematic computations of robot arm are taken upon joint angles of master arm or virtual joint angles which are generated at control programme and position vectors of apex processor are shown in the programme as information. Possible joint angles are also shown in the programme by calculating the reverse kinematics of apex processor according to x, y, z coordinates on three-dimensional coordinate platform.

Slave Arm; as to choice made from control programme: It follows the transactions of master arm (joint angles). It follows the angles of virtual potentiometers in control programme. It moves to the expected point on three-dimensional coordinate platform where apex processor is wanted to reach in control programme and it makes fast acts to the stated coordinates with designated periods.

Key Words: manipulator, robot arm (robot lever), position control, advanced-kinematics, reverse-kinematics

ÖNSÖZ

Teknoloji insanların gündelik ve iş hayatındaki gereksinimleri sonucu ortaya çıkmış ve bu gereksinimler sonucu gelişmeye devam etmektedir. Bilgisayarın icadı ile birlikte Bilgisayar Destekli tasarım ve Üretim teknolojisi gelişmiştir.

Robot teknolojisi de sanayi devriminden sonra makineleşmeye dayalı olarak üretim hatlarındaki üretim süresini azaltmak, standart ürünler üretebilmek, insan hayatı için tehlikeli ve bulunması imkânsız yerlerde işlem ve üretim yapabilmek amacıyla geliştirilmiştir. Üretim ortamları öncelikle klasik üretim tezgâhları daha sonra kam kontrollü ve pim kontrollü üretim tezgâhlarından oluşmaktaydı ve üretim hızı ve kalitesi arttırılmaya çalışılmıştır. Robot teknolojisinin gelişmesiyle birlikte üretim ortamları enaz insan bağımlı, standart ürünler çıkaran, neredeyse yüzde sıfır hatayla üretim yapabilen, insanoğlunun zorlukla yapabileceği işlerin yapılabilirdiği üretim bantlarına dönüşmüş ve üretim ortamları bütünüyle insandan bağımsız düşünüp kontrol edilebilen üretim sistemlerine dönüşmeye doğru gitmektedir. Bu nedenle robot teknolojisinin önemi yakın gelecekte daha da artarak devam etmektedir.

Bu çalışma ülkemizdeki robot kolu konum kontrolü üzerine yapılan çalışmalara bir alternatif sunmaktadır.

TEŞEKKÜRLER

Bu tez çalışmamın başından sonuna kadar emeği geçen ve beni bu konuya yönlendiren, engin bilgi ve tecrübesini benimle paylaşan saygıdeğer hocam ve danışmanım Sayın Yrd. Doç. Dr. Müh. Hilmi KUŞÇU 'ya tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

Bu tez çalışmasındaki malzeme teminini sağlayan Trakya Üniversitesi Bilimsel Araştırma Projeleri çalışanlarına teşekkür ederim.

Ayrıca Yüksek lisans öğrenimi boyunca ilgi ve desteğini hiç eksiltmeyen Mühendislik ve Mimarlık Fakültesi Öğretim Üyesi Yrd. Doç. Dr. Müh. Yılmaz ÇAN 'a teşekkür ederim.

İÇİNDEKİLER

ÖZET	iii
SUMMARY	iv
ÖNSÖZ	v
TEŞEKKÜRLER	vi
İÇİNDEKİLER	vii
SİMGELER.....	x
ŞEKİLLER LİSTESİ	xi
TABLolar LİSTESİ.....	xix
1 GİRİŞ	1
2 KAYNAK BİLGİSİ	3
3 METARYEL VE METOD	6
3.1 Robot Tarihi, Tanımı ve Yapısı	6
3.1.1 Robotların Sınıflandırılması.....	11
3.1.1.1 Serbestlik Derecelerine Göre Robotlar	13
3.1.1.2 Kontrol Yöntemlerine Göre Robotlar	17
3.1.1.3 Robot Hareketlendiricilerinin Kullandığı Güç Kaynağına Göre	17
3.1.1.4 Keskinlik Derecelerine Göre Robotlar.....	18
3.1.1.5 İki Harfli Kodlamayla Yapılan Sınıflandırma.....	19
3.1.2 Benzetim	22

3.1.2.1	Robot Benzetim Programlarının Sınıflandırılması	22
3.1.3	Genel Tanımlamalar	23
3.1.3.1	Konum	23
3.1.3.2	Yönelim	25
3.1.3.3	Öteleme Operatörü (Konum Vektörü)	29
3.1.3.4	Dönme Operatörü	29
3.1.3.5	Homojen Dönüşüm Matrislerinin İleri Yönlü Çarpılması	32
3.1.3.6	Homojen Dönüşüm Matrislerinin Önden Çarpılması	34
3.1.3.7	Dönüşüm Matrisinin Özellikleri	36
3.1.4	Özel Yönelim Tanımlamaları	39
3.1.4.1	Roll-Pitch-Yaw (XYZ Sabit) Açılı seti	39
3.1.4.2	ZYX Euler Açılı Seti	43
3.2	İleri Yön Kinematiği ve Ters Kinematik	46
3.2.1	İleri Yön (Düz) Kinematiği	46
3.2.1.1	Eklem Değişkenlerinin Belirlenmesi	46
3.2.1.2	Koordinat Sistemlerinin Eklemlere Yerleştirilmesi	49
3.2.1.3	İleri Kinematik Problemlerinin Çözümünde Kullanılan Yaklaşımlar	51
3.2.1.4	Robotlarda Kullanılan Bilek Düzenleşimleri	57
3.2.1.5	Altı Serbestlik Derecesine Sahip Robotun İleri Yön Kinematiği	59
3.2.2	Ters Kinematik	62
3.2.2.1	Ters Kinematik Problemlerine Analitik Çözüm Yaklaşımı	65

3.3	Teze Konu Olan Projenin Tanıtımı	73
3.3.1	Projede Kullanılan Sistem Elemanları	74
3.3.1.1	Hareketleri Algılayan Kol (Master Kol) ve Yapısı	74
3.3.1.2	Hareketleri Takip Eden Kol (Slave Kol) ve Yapısı	76
3.3.1.3	Servo Motor Sürücü Kontrol Kartı	80
3.3.1.4	Analog-Sayısal Dönüştürücü Veri Giriş ve Çıkış Kartı	111
3.3.2	Projede Kullanılan Kontrol Yazılımı	117
3.3.2.1	LabVIEW Grafiksel Programlama Dili	118
3.3.2.2	Analog-Sayısal Veri Toplama Kartı Sürücüleri	126
3.3.2.3	MATLAB Script ile İleri ve Ters Kinematik Hesaplamalar	132
3.3.2.4	Kontrol Programı ve Sekmeler	150
4	HEDEFLENEN SONUÇLAR VE ÖNERİLER	188
4.1	Hedeflenen Sonuçlar	188
4.2	Öneriler	189
5	TEZ SIRASINDA YAPILAN ÇALIŞMALAR	190
5.1	Uluslararası Yayınlar	190
5.2	Ulusal Yayınlar	190
	KAYNAKLAR	191
	ÖZGEÇMİŞ	193

SİMGELER

a_{i-1}	:İki eksen arasındaki uzuv uzunluğu (°)
α_{i-1}	:İki komşu eksen arasındaki eksen açısı (°)
d_i	:Üst üste çıkan bağlar arasındaki eklem kayması veya kaçıklığı (mm)
θ_{i-1}	:İki komşu uzuv arasındaki eklem açısı (°)
${}^A\hat{X}_B$: \hat{X}_B Birim vektörünün {A} koordinat sistemine göre yönelimi
AP	:P noktasının {A} koordinat sistemine göre konumu
R	:Dönme Matrisi (3x3)
B_R	:{A} koordinat sistemini {B} koordinat sistemi cinsinden dönüşüm matrisi
${}^A_B R^T$: ${}^B_A R$ Dönme matrisinin devriği veya transpozesi
${}^B_A R^{-1}$: ${}^B_A R$ Dönme matrisinin tersi
I	:Birim matrisi
n	:Dönme matrisinin normal vektörü (3x1)
s	:Dönme matrisinin kayma vektörü (3x1)
a	:Dönme matrisinin yaklaşım vektörü (3x1)
α	:Z eksenini boyunca yapılan açı (°)
β	:Y eksenini boyunca yapılan açı (°)
γ	:X eksenini boyunca yapılan açı (°)
l	:Uzuv uzunluğu (cm)
T	:Dönüşüm matrisi (4x4)
ms	:Milisaniye
μs	:Mikrosaniye
P_x	:Konum vektörünün x eksenini değeri (mm)
P_y	:Konum vektörünün y eksenini değeri (mm)
P_z	:Konum vektörünün z eksenini değeri (mm)
ADC	:Analog Digital Converter (Analog Sayısal Çevirici)
DAQ	:Data Acquisition (Veri Toplama)
AIO	:Analog Input Output

ŞEKİLLER LİSTESİ

Şekil 3.1 Döner Eklem	10
Şekil 3.2 Prizmatik Eklem	10
Şekil 3.3 Robot ve Çalışma Düzlemleri	11
Şekil 3.4 Seri Robot	12
Şekil 3.5 Paralel Robot.....	12
Şekil 3.6 Kartezyen Robot ve Çalışma Alanı.....	14
Şekil 3.7 Silindirik Robot ve Çalışma Alanı.....	14
Şekil 3.8 Küresel Robot ve Çalışma Alanı.....	15
Şekil 3.9 Döner Robot ve Çalışma Alanı.....	15
Şekil 3.10 İki Harfli Kodlamada Kullanılan Semboller.....	20
Şekil 3.11 Robotları İki Harfli Kodlamada Tanımlayan Sembolik Şekilleri.....	21
Şekil 3.12 ${}^A P$ Vektörü.....	24
Şekil 3.13 ${}^A P_{BORG}$ Vektörü	24
Şekil 3.14 ${}^D P$ Vektörü	25
Şekil 3.15 {B} koordinat sisteminin {A} koordinat sistemine göre yönelimi.....	25
Şekil 3.16 \hat{X}_B birim vektörünün {A} koordinat sistemine göre yönelimi.....	26
Şekil 3.17 {B} koordinat sisteminin {A} koordinat sistemine göre yönelimi.....	28
Şekil 3.18 {B} koordinat sistemi	28
Şekil 3.19 P vektörünün Z eksenini etrafında θ açısıyla döndürülmesi	30

Şekil 3.20 P vektörünün Y eksenini etrafında θ açısıyla döndürülmesi.....	31
Şekil 3.21 P vektörünün X eksenini etrafında θ açısıyla döndürülmesi.....	32
Şekil 3.22 {B} koordinat sisteminin X eksenini etrafında -45° döndürülmesi.....	33
Şekil 3.23 {B} koordinat sisteminin Y eksenini etrafında $+90^\circ$ döndürülmesi	33
Şekil 3.24 {B} koordinat sistemi döndürülerek elde edilen {A} koordinat sistemi	33
Şekil 3.25 {A} koordinat sisteminin Z eksenini etrafında 45° döndürülmesi.....	35
Şekil 3.26 {A} koordinat sisteminin Y eksenini etrafında 90° döndürülmesi	35
Şekil 3.27 {B} koordinat sistemi döndürülerek elde edilen {A} koordinat sistemi	35
Şekil 3.28 Uç İşlevcisi	36
Şekil 3.29 {A} koordinat sistemine göre {B} koordinat sistemi cinsinden ${}^B({}^A P)$ Vektörü	37
Şekil 3.30 Roll-Pitch-Yaw Açısı seti.....	40
Şekil 3.31 Uç işlevcisi.....	42
Şekil 3.32 {B} koordinat sisteminin Z_B eksenini boyunca α açısıyla döndürülmesi.....	43
Şekil 3.33 {B} koordinat sisteminin Y_B eksenini boyunca β açısıyla döndürülmesi.....	44
Şekil 3.34 {B} koordinat sisteminin X_B eksenini boyunca γ açısıyla döndürülmesi.....	44
Şekil 3.35 Dönme Eksenlerinin Belirlenmesi	47
Şekil 3.36 Dönme Eksenlerine Z, Y ve X Eksenlerinin Yerleştirilmesi.....	47
Şekil 3.37 a_{i-1} uzuv uzunluğu.....	48
Şekil 3.38 d_i eklem kaçıklığı	48

Şekil 3.39 α_{i-1} eksen açısı	48
Şekil 3.40 θ_i eklem açısı	49
Şekil 3.41 İki boyutlu düzlemde hareket eden robot kolu	51
Şekil 3.42 İki boyutlu düzlemde hareket eden P noktasının konumu	52
Şekil 3.43 İki boyutlu düzlemde hareket eden kolun başlangıç değerleri.....	52
Şekil 3.44 Dönme Eksenlerine Z Eksenlerinin Yerleştirilmesi	52
Şekil 3.45 X Eksenlerinin Yerleştirilmesi.....	53
Şekil 3.46 Sağ El kuralına göre Y Eksenlerinin Yerleştirilmesi	53
Şekil 3.47 Euler Bileklik Düzenleşimi.....	57
Şekil 3.48 Eklem Kaçıklıklı Bileklik Düzenleşimi.....	58
Şekil 3.49 Euler Bilekli 6 Serbestlik Derecesine Sahip Scara Robot	59
Şekil 3.50 Euler Bilekli Scara Robota Eksenlerin Yerleştirilmesi.....	60
Şekil 3.51 Eklem Değişkenleri ile Kartezyen Uzay Dönüşümü	62
Şekil 3.52 Fiziksel Çözüm	63
Şekil 3.53 Matematiksel Çözüm	63
Şekil 3.54 Puma Robotun Aynı Noktaya Dört Farklı Şekilde Ulaşması	64
Şekil 3.55 İki boyutlu düzlemde hareket eden robot kolu	66
Şekil 3.56 İki boyutlu düzlemde hareket eden P noktasının konumu	71
Şekil 3.57 Ters Kinematik Probleminin Gerçek Ve Sanala Çözümlerine Ait Konumları	72
Şekil 3.58 Master kol	74

Şekil 3.59 Master kol ölçüleri	75
Şekil 3.60 Çeşitli potansiyometreler	75
Şekil 3.61 Potansiyometrenin içyapısı	76
Şekil 3.62 Potansiyometrelerin eklemlere yerleştirilmiş hali	76
Şekil 3.63 Slave kol	77
Şekil 3.64 Slave kolun ölçüleri	77
Şekil 3.65 Tipik bir hobby servo motor	78
Şekil 3.66 0,5ms 'lik modülasyon genişliği ve motor milinin aldığı açı değeri	79
Şekil 3.67 1,5ms 'lik modülasyon genişliği ve motor milinin aldığı açı değeri	79
Şekil 3.68 2,5ms 'lik modülasyon genişliği ve motor milinin aldığı açı değeri	79
Şekil 3.69 PIC 16F877'nin bacak yapısı.....	83
Şekil 3.70 PIC 16F877'nin fiziksel görünüşü.....	83
Şekil 3.71 ADCON0 Kaydedicisinin Bit Yerleşimi	88
Şekil 3.72 ADCON1 Yazmacının Bit Yerleşimi	89
Şekil 3.73 A/D Blok Diyagramı.....	90
Şekil 3.74 A/D Transfer Fonksiyonu	94
Şekil 3.75 Biçimlendirme bitlerinin bellekteki bit dizilimi	96
Şekil 3.76 Servo Motor Sürücü Kartı Devre Şematiği.....	99
Şekil 3.77 Baskı Devreye Elemanların Yerleşimi.....	99
Şekil 3.78 Baskı devre.....	100

Şekil 3.79 ISIS simülasyon programında Servo motorların kontrol sinyallerinin osiloskop görüntüsü	105
Şekil 3.80 Devre şeması.....	109
Şekil 3.81 Osiloskop 1	110
Şekil 3.82 Osiloskop 2	110
Şekil 3.83 JAC AIO3320 Kart ve ekipmanları	111
Şekil 3.84 AIO 3320 kartın fiziksel ölçüleri ve Blokları[12]	112
Şekil 3.85 JS51026 Bağlantı Terminali[12].....	113
Şekil 3.86 ADP9201 Bağlantı Terminali[12]	113
Şekil 3.87 JS51026 bağlantı terminali	115
Şekil 3.88 DB37PF kablosu	115
Şekil 3.89 ADP9201 bağlantı terminali	116
Şekil 3.90 ADP9201 20 pinli kablo	116
Şekil 3.91 Kontrol programına ait kullanıcı arayüzü.....	118
Şekil 3.92 Ön panel.....	121
Şekil 3.93 Blok diyagram.....	121
Şekil 3.94 Kontrol ve Gösterge nesnesi	122
Şekil 3.95 Kontrol paleti	123
Şekil 3.96 Kontrol ve Gösterge terminalleri	124
Şekil 3.97 Fonksiyon paleti.....	125
Şekil 3.98 inital.vi sürücüsüne ait Ön panel ve Blok Diyagramı[13]	127

Şekil 3.99 close.vi sürücüsüne ait Ön panel ve Blok Diyagramı[13]	127
Şekil 3.100 Case Sturacture yapısına True sinyali gönderilmesi.....	128
Şekil 3.101 Case Sturacture yapısına False sinyali göndrilmesi.....	128
Şekil 3.102 info.vi sürücüsüne ait Ön panel ve Blok Diyagramı[13]	129
Şekil 3.103 smart_AtoD.vi sürücüsüne ait Ön panel ve Blok diyagramı[13].....	129
Şekil 3.104 Out_DA_data.vi sürücüsüne ait Ön panel ve blok diyagramı[13].....	130
Şekil 3.105 Read_point.vi sürücüsünün Ön panel ve Blok Diyagramı[13].....	131
Şekil 3.106 Set_point.vi sürücüsünün Ön panel ve Blok diyagramı[13].....	132
Şekil 3.107 Dört serbestlik dereceli robot kolun matematiksel modeli	133
Şekil 3.108 Kontrol programının kullanıcı arayüzü.....	151
Şekil 3.109 Simülasyon başlatma ve durdurma butonları.....	151
Şekil 3.110 Güç Kaynağı açık durumda iken blok diyagram durumu.....	152
Şekil 3.111 Güç Kaynağı kapalı durumda iken blok diyagram durumu.....	152
Şekil 3.112 İleri Kinematik – Eklem Açılarını Ver	153
Şekil 3.113 Sanal potansiyometreleri.....	153
Şekil 3.114 Eklemlere ait konum vektörleri	154
Şekil 3.115 Slave kolun anlık üç boyutlu grafiği.....	154
Şekil 3.116 Eklem Değişkenleri.....	155
Şekil 3.117 Eklem açılarının Sin ve Cos karşılıkları	155
Şekil 3.118 İleri kinematik MATLAB scripti	156
Şekil 3.119 Giriş değişkenleri Blok diyagramı	159

Şekil 3.120 3D Curve nesnesinin blok diyagramı.....	160
Şekil 3.121 Out_DA_data.vi sürücüsünün 1. eklem için düzenlenmiş hali.....	161
Şekil 3.122 1 nolu eklem açısının Sayısal değere dönüştürülmesi	162
Şekil 3.123 smart_AtoD.vi Blok diyagramı.....	162
Şekil 3.124 smart_AtoD.vi blok diyagramının 1 nolu eklem açısını okumak için düzenlenmiş hali	163
Şekil 3.125 Gerilimin açı bilgisine dönüştürülmesi.....	163
Şekil 3.126 Potansiyometredeki açı sınırlaması.....	164
Şekil 3.127 İleri Kinematik – Eklem Açılarını Ver sekmesine ait kullanıcı arayüzü ...	164
Şekil 3.128 Ters kinematik MATLAB scripti ve Blok diyagramı.....	165
Şekil 3.129 Uç işlevcisi açıklığını kontrol eden potansiyormetre.....	180
Şekil 3.130 Uç işlevcisinin X,Y,Z koordinat girişi.....	180
Şekil 3.131 Bulunan Eklem Açıları	181
Şekil 3.132 İterasyon Blok diyagramı.....	181
Şekil 3.133 İterasyon seçim listesi.....	181
Şekil 3.134 3B Koordinata Göre Eklem Açılarını Bulma sekmesinin kullanıcı arayüzü	182
Şekil 3.135 Koordinat Listesinden Seçme sekmesine ait kullanıcı arayüzü.....	183
Şekil 3.136 Koordinat Seçim butonu	183
Şekil 3.137 Koordinat Seçim listesi	184
Şekil 3.138 Koordinat seçim blok diyagramı.....	184

Şekil 3.139 Seri Hareket sekmesinin blok diyagramındaki Time Wait fonksiyonu	186
Şekil 3.140 Seri Hareket işlem süresi ve numarası	186
Şekil 3.141 Seri Hareket sekmesinin kullanıcı arayüzü	187
Şekil 4.1 Eklem açılarına göre konum vektörleri ve üç boyutlu grafiği	188
Şekil 4.2 Üç boyutlu koordinatlara göre eklem açıları ve üç boyutlu grafiği	189

TABLULAR LİSTESİ

Tablo 3.1 Robot ve Çalışma Düzlemleri[3]	11
Tablo 3.2 Robot Düzenleşimlerinin Avantaj ve Dezavantajları[3].....	16
Tablo 3.3 D-H Değişkenleri	50
Tablo 3.4 D-H Değişkenleri	54
Tablo 3.5 D-H Değişkenleri	57
Tablo 3.6 D-H Değişkenleri	58
Tablo 3.7 D-H Değişkenleri	60
Tablo 3.8 PIC 16F877’de Pin Tanımlamaları	84
Tablo 3.9 ADCON1 Yazmacı tablosu	89
Tablo 3.10 Osilatör seçim tablosu.....	96
Tablo 3.11 Çeşitli osilatör frekanslarında SERVO_? Değişkenleri için kullanılacak formül.....	104
Tablo 3.12 JS51026 bağlantı terminali pin tablosu.....	114
Tablo 3.13 ADP9201 bağlantı terminali pin tablosu	116
Tablo 3.14 D-H Değişkenleri.....	133
Tablo 3.15 Seri hareket eklem açısı tablosu	185

1 GİRİŞ

Bu çalışmada dört serbestlik derecesine ve uç işlevcisine sahip robot kolunun yine dört serbestlik derecesine sahip bir robot kolu ve kontrol programı tarafından kontrol edilmesi gerçekleştirilmiştir. Robot kollarından birinin eklemlerine eklem açılarını algılamak amacıyla hassas telli potansiyometreler yerleştirilmiş ve adına master kol adı verilmiştir. Diğer robot kolunun eklemlerine RC servo motorlar yerleştirilmiş ve adına slave kol adı verilmiştir. Slave kolun hareket yeteneği kontrol programında kullanıcının yapacağı seçime göre değişmektedir.

Slave kol;

- Master kolun hareketlerini (Eklem Açılarını) birebir takip eder
- Kontrol programındaki sanal potansiyometrelerin açılarını takip eder
- Kontrol programında uç işlevcisinin üç boyutlu koordinat düzleminde olması istenen noktaya hareket eder
- Kontrol programında belirlenen koordinatlara belirlenen periyotla seri hareketler yapar.

Master kolun eklemlerindeki potansiyometrelerinden alınan eklem açısı bilgilerinin yani analog forma dönüştürülmüş gerilim bilgisinin bilgisayar ortamında işlenebilmesi ve Master kol veya kontrol programından alınan eklem açısı bilgisinin Slave kolu kontrol edebilmesi amacıyla Analog Digital Converter Data Acquisition (ADC DAQ) kart kullanılmıştır. Bu kart aracılığıyla Master koldan alınan eklem açısı bilgisi veya kontrol programında üretilen eklem açısı bilgisi slave koldaki servo motorları kontrol amacıyla servo motor kontrol kartına gönderilir. Servo motor kontrol kartı ADC DAQ veri toplama kartından alınan ve analog forma dönüştürülmüş eklem açısı bilgilerine göre servo motorları kontrol etmek amacıyla frekansı 50Hz Modülasyon genişliği 0.5ms ile 2.5ms arasında değişen bir sinyal üretir.

Master koldan alınan eklem açılarına veya kontrol programında üretilen sanal eklem açılarına göre slave kolun eklemlerine ve uç işlevcisine ait konum vektörleri

hesaplanıp bilgi amacıyla programda gösterilir. Yine kontrol programında uç işlevcisinin üç boyutlu koordinat düzleminde bulunması istenen noktanın koordinatları verildiğinde program slave kolun olası eklem açılarını hesaplayarak hem slave kolun hareket etmesini sağlar hemde bilgi amacıyla eklemlerin konum vektörleri bilgi amacıyla programda gösterilir. Ayrıca uç işlevcisinin açıklığı yine master kol üzerindeki potansiyometre veya program üzerindeki sanal potansiyometre yardımıyla kontrol edilir.

Bu çalışma bir robot kolu konum kontrolünün bilgisayar destekli veri toplama kartıyla yapılabilmesi için gerekli ileri ve ters kinematik değişkenlerinin bulunup bir kontrol yazılımının oluşturulması amaçlanmıştır.

2 KAYNAK BİLGİSİ

Dünya üzerinde robot kolları üzerine birçok çalışmanın yapıldığı görülmektedir. Robot teknolojisi yakın tarihte ortaya çıktığı sanılsa da robot teknolojisinin tarihi M.Ö. 3000 'li yıllara dayanmaktadır.

İlk sibernetikçi (Bir insanın veya otomatik bir makinenin, modern tekniğin kaynakları çerçevesinde herhangi bir işi yönetmesini veya belli bir amaca ulaşmasını sağlayan bilim.) kabul edilen Ebul-iz İsmail bin ar-Razzaz el-Cezeri (MS 12yy) Sadece suyun kaldırma ve basınç gücünü kullanarak tamamen yeni bir teknik ve sistem kurmuş, çok yönlü otomatik hareketler elde edebilmiştir. Robot kelimesi İngilizce diline 1921 yılında oyun yazarı Karel Capek tarafından R. U. R. (Rossum's Universal Robots) adlı dramasıyla tanıtılmıştır. Bu çalışmada, robotlar insanlara benzeyen fakat yorulmadan çalışan makinelerdi. Önceleri robotlar, insan işçilerin yerini alsınlar diye üretilirken günümüzde insan robot çekişmesi yok olarak insanlar lehine sonuçlanmıştır. 1940'ların sonlarına doğru Oak Ridge ve Argonne Ulusal Laboratuarlarında radyoaktif malzemeleri işlemek için uzaktan kontrollü mekanik manipülatörlerin araştırma programları başlatıldı. Bu sistemler "Ana-Uydu"(Master-Slave) tipindeydiler ve insan operatörler tarafından yapılan el ve kol hareketlerini büyük bir sadakatle yerine getirmekteydiler. Bu çalışma bizim çalışmamıza yakın bir çalışmadır. Ana manipülatöre kullanıcı tarafından bir dizi hareket verilmekte uydu olansa bu hareketleri mümkün olduğu kadar hızlı takip etmekteydi. Sonraları sisteme güç geri beslemesi eklenerek ana manipülatörün, uydu ve onun çevresi arasındakileri hissetmesi sağlanmıştır. Devol ve Joseph F. Engelberger 1959 yılında Unimaton Inc. tarafından tanıtılan ilk endüstriyel robotlara öncülük etmişlerdir. Bu cihazın özelliği bir manipülatörle bir bilgisayarın uyumlu çalışması sonucu ortaya çıkan birçok değişik görevi otomatik olarak gerçekleştiren bir makine olmasıdır. 1960'larda bu cihazların esnekliklerinin duyuşal geri beslemeyle önemli ölçüde geliştirilebileceği belirginleşmiştir. 1962'de H. A. Ernst bilgisayar kontrollü dokunma sensörlü mekanik bir elin gelişiminden bahsetmiştir. MH-1 diye anılan bu cihaz blokları algılayabiliyor ve bu blokları operatör yardımı olmaksızın istif yapabiliyordu. Bu çalışma uyarlamalı davranış kapasiteli bir robotun ilk örneklerinden biridir. Aynı zaman dilimi içerisinde Tomovic ve Boni (1962) nesneyi hisseden ve bir motora iki kavrama kalıbından birini seçmesi için bir geri besleme

sinyali sađlayan basınç sensörlü bir prototip el geliřtirdiler. El nesneye temas ettiđi zaman nesne boyutuyla ve ađırlıđıyla orantılı bilgi bu basınç sensörleri tarafından bilgisayara gönderilmiřti. 1963’de Amerikan Makine ve Dökümhane řirketi (American Machine and Foundry Company) VERSATRAN ticari robotunu tanıttı. Aynı yılın bařlangıcında manipülatörler için Roehampton kolu ve Edinburgh kolu gibi deđişik kol tasarımları geliřtirilmiřtir.

1968’de McCarthy ve meslektařları Stanford Yapay Zeka laboratuvarında elleri, gözleri ve kulakları (manipülatörler, TV kameralar, ve mikrofon) olan bir bilgisayar geliřtirmişlerdir. Konuşulanların farkına varan, bir masanın üzerindeki dađınık blokları gören ve komutlar dođrultusunda onları düzene koyan bir sistem göstermişlerdi. Bu zaman içerisinde Pieper (1968) bilgisayar kontrollü bir manipülatörün kinematik problemi üzerinde çalışırken Kahn ve Roth (1971) kısıtlanmış bir kolun dinamiđi ve kontrolü üzerine çalışmışlar.

Robotlarda garip gelişmelerden biri General Elektrik řirketinin Amerika Birleşik Devletleri ordusu için geliřtirdiđi deneysel yürüyen kamyonla 1969’da meydana gelmiştir. Aynı yıl içerisinde Boston kolu ve bir sonraki yılda kamerayla donatılmış ve bilgisayar kontrollü Stanford kolu geliřtirildi.

Robot bilimindeki ciddi çalışmalar bu kolların manipülatör olarak kullanılmasıyla başlamıştır. Stanford kolunun deđişik stratejiler kullanarak blokları otomatik olarak istif etmesiyle bir deney yapılmıştır. Bu o zamanlarda otomatik bir robot için çok karmaşık bir işti. 1974’de Cincinnati Milacron ilk bilgisayar kontrollü endüstriyel robotunu tanıttı. “The Tomorrow Tool”, yada T3 diye adlandırılan bu robot 45kg.’den fazla ađırlık kaldırabiliyor ve montaj hatlarında hareketli nesnelere takip edebiliyordu.

1970’lerde arařtırmaların büyük bir bölümü manipülatör işlemlerini kolaylařtırmak için harici sensörlerin kullanımına yoğunlaşmıştı. Stanford’ta Bolles ve Paul (1973) görsel ve kuvvet geri beslemesinin ikisini de kullanarak bilgisayar kontrollü bir Stanford kolunu PDP-10 bilgisayarına bađlayıp bir otomobil parçası üretiminde kullanmışlardır. Hemen hemen aynı zaman diliminde IBM’den Will ve Grossman (1975) bir bilgisayar kontrollü, dokunma ve kuvvet sensörlü manipülatörü 20 parçalı bir daktilo üretiminde kullanmak üzere geliřtirilmişlerdir. Inoue, MIT Yapay Zeka

Laboratuvarında güç geri beslemesine yapay zekanın uygulanması üzerine çalışmıştır(1974). Doğru montaj görevi içerisinde başlangıç konumlanmasını gerçekleştirmek için bir fener arama tekniği kullanıldı. Draper Laboratuvarında Nevins et al (1974) uyuma dayanan hissetme tekniklerini araştırmıştır. Bu çalışma Uzaktan Merkezi Uyum diye adlandırılan pasif uyumlu cihazların enstrümantasyonuna doğru geliştirilmiştir. Bu cihazlar yakın parçaların birleştirilmesi için manipülâtörün son birleşme yerinden destek tabakaya bağlıdır. Bejczy (1974) Jet Propulsion laboratuvarında uzay araştırma projeleri için geliştirilmiş Stanford Kolu üzerinde bilgisayar tabanlı tork kontrol tekniğini uygulamıştır.[1]

Günümüzde robot çalışmaları ileri kinematik, ters kinematik, dinamik, konum kontrolü, uç işlevcisi duyarlılığı, kontrol yazılımı ve operatörden bağımsız işlem yapabilme yeteneği kazanması için yapay zeka gibi bilim alanlarıyla ortak çalışmalar sonucu geliştirilmeye devam edilmektedir.

3 METARYEL VE METOD

Bu bölümde öncelikle robot tarihi, tanımı ve yapısı incelenip, Endüstride üretim bantlarında kullanılan robot kollarının sınıflandırılması yapılacaktır. Robot kollarının konum kontrolünün yapılabilmesi için ileri kinematik ve ters kinematik hesaplamaların nasıl yapılacağı ve bu hesaplamalarda kullanılan çeşitli yöntemler hakkında ayrıntılı bilgi verilecektir.

Çalışmanın uygulama safhasında dört serbestlik dereceli dönel ekleme sahip ve uç işlevcisi bulunan iki adet robot kolu yapıp kollardan birinin eklemlerine eklem hareketlerini algılayan potansiyometre diğer robot kolunun eklemlerine ise eklem hareketini sağlayacak servo motorlar yerleştirilecektir. Bilgisayara bağlı veri toplama kartı ve bu çalışma için tasarladığımız servo motor kontrol kartı aracılığıyla robot kolunun hareketleri takip etmesi sağlanacaktır. Master kolun eklem açılarının izlenebilmesi ve Slave kolun hareketleri takip edebilmesi amacıyla yine bu çalışmaya özel kontrol programı hazırlanacaktır.

3.1 Robot Tarihi, Tanımı ve Yapısı

Robot kelimesi ilk olarak 1920 yılında kullanılmış olsa da, robotlara ait ilk kavramlar ve robot benzeri ilk makinelere ait bilgiler M.Ö. 3000 yıllarına kadar uzanmaktadır. Eski Mısır, eski Yunan ve Anadolu medeniyetlerinde otomatik su saatleri benzeri makinelerin geliştirildiği bilinmektedir. Homerus 'un İlyada eserinde insan yapımı kadın hizmetçiler anlatılmaktadır. M.Ö. 100 yıllarında yaşamış olan İskenderiye 'li bir mühendisin otomatik açılan kapılar, fiskiyeler v.b. gibi düzenekleri su ve buhar gücü ile çalıştırdığı eski kitaplarda yazılmaktadır. Daha yeniçağlarda Leonardo Da Vinci 'nin yürüyen mekanik aslanı olduğu söylenmektedir. Bu süreç içinde özellikle batı dünyasında iyi bilinmeyen El Cezeri 'nin (MS 12 yy) robot teknolojisi konusunda çok sayıda ve zamanına göre çok ileri öneri ve uygulamaları bulunmaktadır. Robot kelimesi Çek ve Slovak dillerinde köle veya işçi anlamına gelir.

M.Ö. 800 Homeros İlyada adlı eserinde hareketli üçayaklılardan bahseder.

M.Ö. 350 Aristo insanların isteklerini anlayıp itaat eden mekanizmalar öngörür.

1350 Mekanik bir horoz Fransa'daki Strazburg Katedralinin tepesine yerleştirilir.

1801 Joseph-Marie Jacquard delikli kartlarla kontrol edilen otomatik dokuma tezgâhını icat eder.

1921 "Robot" sözcüğü ilk defa Çekoslovak oyun yazarı Capek tarafından kullanılır.

1938 Willard Pollard ve Harold Roselund spreyci boyacı yapan otomatik bir makine için eklemleri olan bir kol icat ederler.

1939 Westinghouse şirketi New York Dünya Fuarı 'nda sergilenmek üzere mekanik bir insan ve köpek tasarlar.

1942 Isaac Asimov "Runaround" adlı kitabında robotların üç kanununu ortaya koyar.

Bir robot bir insana zarar veremez veya kayıtsız kalarak bir insanın zarar görmesine neden olamaz. Birinci yasa ile çatışmamak şartı ile bir robot insanlar tarafından verilen emirlere uymak zorundadır. Birinci ve ikinci yasa ile çatışmamak şartı ile bir robot kendi varlığını korumalıdır.

1951 Raymond Goertz radyoaktif maddelerle ilgili çalışmalarda kullanılmak üzere uzaktan kumandalı bir kol tasarlar.

1954 Devol programlanabilen fabrika robotunu tasarlar Unimation.

1960 AMF firması Harry Johnson ve Veljko Milenkovic tarafından geliştirilen Versatran endüstriyel tasarımını dünyaya sunar.

1967 Ralph Moser, General Electric şirketinde yürüyen robotu tasarlar.

1967 Japonya ilk endüstriyel robotunu ithal eder.

1968 Görme organına ve yapay zekaya sahip ilk robot, Shakey, Stanford Araştırma Enstitüsünde geliştirilir.

1971 Cincinnati Milacron firması bilgisayar kontrollü robotunu piyasaya sürer.

1972 Shigeo Hirose, Tokyo Teknoloji Enstitüsünde bir öğrenci, yılan benzeri bir robot yapar.

- 1976 NASA Mars'a gidecek olan uzay araçlarına robot kollar yerleştirir.
- 1977 Asea Brown Boveri Ltd. şirketi mikrobilgisayar kontrollü robotları piyasaya sürer.
- 1979 Yamanashi Üniversitesi fabrikalardaki montaj hatlarında kullanılmak üzere Scara Kolu'nu tasarlar.
- 1983 Odetics şirketi, 6 bacaklı, yürüyen robotu piyasaya sürer.
- 1984 Waseda Üniversitesi'nde Wabot-2 adlı nota okuyup, elektronik org çalabilen robot yapılır.
- 1986 Honda Motor Corporation insansı bir robot geliştirmek amacı ile gizli bir proje başlatır.
- 1988 Danbury Hastanesi'nde ilk yardımcı robot göreve başlar.
- 1993 MIT 'den Rodney A. Brooks bir insan gibi yetiştirilen ve eğitilen robot Cob 'u yapmaya başlar.
- 1994 Dante II, Carnegie Mellon Üniversitesi'nde geliştirilen yürüyen robot Alaka 'da aktif bir volkana kesif gezisi yapar ve volkanik gaz örnekleri toplar.
- 1996 Honda, P2(prototipe-2), yürüyen insansı robot dünyaya tanıtılır.
- 1997 ilk yıllık robotlar arası futbol turnuvası "éRobocup" Japonya'da düzenlenir.
- 1997 NASA 'nin Pathfinder uzay aracı Mars'a iner ve "Sojourner" robotu Mars yüzeyinde keşif gezisi yapar.
- 2000 RoboCup 2000'de üç insansı robot ilk defa karşılaşır. Batı Avustralya Üniversitesi'nden Johnny Walker, Japonya'da Aoyama Gakuin Üniversitesi'nden Mk-II ve Pino.
- 2004 NASA Mars 'da hayat izi aramak görevi ile iki robot göndermiştir.(Spirit ve Opportunity)
- 2005 Türkiye Florence Nightingale Hastanesinde kullanılan dört kollu cerrah robot Da Vinci ile %100 başarılı ameliyatlar yapılmaktadır.[2]

Robotların şimdiye kadar birçok farklı tanımı yapılmıştır. Webster sözlüğünde robot, genellikle insanların gerçekleştirdikleri işlevleri yerine getiren otomatik araçlar olarak tanımlanmaktadır. Ancak bu tanıma göre mesela bir çamaşır makinesi de robot sayılabilmektedir.

Robotun, Amerikan Robot Enstitüsü tarafından yapılan tanımı ise, malzemelerin, parçaların ve araçların hareket ettirilebilmesi için tasarlanmış olan çok fonksiyonlu ve programlanabilir manipülatör veya farklı görevleri yerine getirebilmek için değişken programlı hareketleri gerçekleştirebilen özel araç şeklindedir.

Robot bir kaide üzerinde en az bir kol, tutma organları (genellikle pensler, vantuzlar veya elektromıknatıslar), pnömatik, hidrolik veya elektriksel sensörler ile konumu ve basınç algılayıcılarıyla, bilgi işlem organlarıyla donatılmış kontrollü mekanik manipülatörlerdir.

Robotlar verilen bir dizi görevleri önceden programlanmış hareketler aracılığıyla malzemeleri, materyalleri, el aletlerini veya özel donanımları hareket ettirmek amacıyla tasarlanmış çok işlevli manipülatörlerdir.

Sanayi robotunun en kapsamlı tanımı ve robot tiplerinin sınıflandırılması ISO 8373 standardında belirlenmiştir. Bu standarda göre bir robot şöyle tanımlanır; Endüstriyel uygulamalarda kullanılan, üç veya daha fazla programlanabilir eksenli olan, otomatik kontrollü, yeniden programlanabilir, çok amaçlı, bir yerde sabit duran veya hareket edebilen birden fazla motorlu makine veya el aletine robot denir.

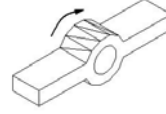
Yukarıdaki tanımlarda da görüldüğü gibi robot; canlılara benzer işlevleri olan ve davranış biçimleri sergileyen makinelerdir.[2]

Bilgisayar destekli tasarım ve buna paralel olarak bilgisayar destekli üretimin gelişmesiyle endüstriyel robot ihtiyacı ve kullanımı artmıştır. Robotları diğer makinelerden ayıran en önemli özellik robotların birden fazla programlanabilmeleri ve mekanik sistemlerinin yanında kontrol sistemi içermesidir. Kısacası robot bilgisayar kontrollü endüstriyel manipülatördür. Robot bilimi elektrik, elektronik, makine, bilgisayar ve matematik mühendisliğinin içine girdiği disiplinler arası bir bilim dalıdır. Makine mühendisliği robotların dinamik ve statik yapısıyla, bilgisayar mühendisliği robotun kontrolünde kullanılacak programla ve elektrik mühendisliği de robotta kullanılacak algılayıcı tasarımıyla ilgilenir.

Tüm robotların sahip olduğu uzuv ve eklemlere göre bir çalışma uzayı vardır. Bu çalışma uzayı robotun uç işlevcisinin serbest olarak hareket edebileceği alandır. Robot

manipülâtörünün eklem açıları yardımıyla ana çerçeveye göre uç işlevcisinin konumu ve yöneliminin bulunmasına ileri kinematik, uç işlevcisinin konum ve yönelim bilgileriyle robot manipülâtörünün eklem değişkenlerinin bulunmasına ise ters kinematik denir. Kinematik nesnelere hareketiyle ilgilendir. Robot kinematiği ile robotun kuvvet, hız ve ivme analizi yapılabilir.

Bir robot birbirinden bağımsız olarak hareket edebilen prizmatik (kayar) veya döner tip eklemlerden ve bu eklemleri birbirine bağlayan uzuvlardan oluşur. Döner eklem (Şekil 3.1) iki uzuv arasında dönme hareketine izin verir. Prizmatik eklem (Şekil 3.2) ise iki uzuv arasında doğrusal harekete izin verir. Dönel eklemlerde dönme hareketinden dolayı oluşan yer değiştirmeye eklem açısı, prizmatik eklemlerde uzuvlar arası doğrusal hareketten oluşan yer değiştirmeye ise eklem kayması denir. Dönel eklemlerde eklem değişkeni eklem açısı, prizmatik eklemlerde ise eklem değişkeni eklem kaymasıdır.



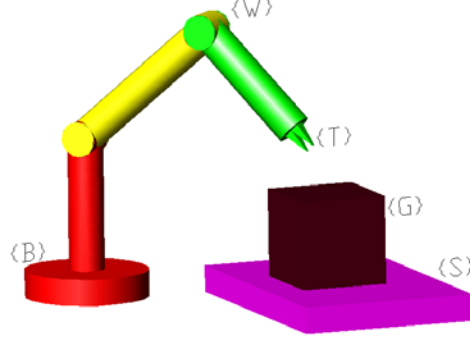
Şekil 3.1 Döner Eklem



Şekil 3.2 Prizmatik Eklem

Manipülâtörler birbirlerine bağlanmış rijid cisimlerin açık uçlu kinematik zinciri olarak kabul edilirler. Bu Zincirin bir ucu ana çevreye bağlı iken diğer ucu uç işlevcisine bağlıdır. Sonuç olarak manipülâtörün hareketi her bir uzuvun diğerine göre hareketlerinin toplamıdır. Bu kinematik zinciri oluşturan ifadeler Robotun konum ve yönelimini içeren homojen dönüşüm matrisinden oluşur. Manipülâtör hareketini belirlemek için öncelikle rijid cismin uzaydaki yönelimi ve konumunun belirlenmesi gerekir. Rijit cismin kartezyen uzaydaki yönelimi dönme matrisi ile rijit cismin konumu ise konum vektörü ile bulunur. Üç boyutlu uzayda herhangi bir noktaya ulaşmak için 6

serbestlik derecesi yeterlidir. Serbestlik derecesi altıdan fazla olan robotlarda artıklık meydana gelir. Artıklık bir eklemin taradığı alanı başka bir ekleminde taraması olarak tanımlanabilir.



Şekil 3.3 Robot ve Çalışma Düzlemleri

Ana Çerçeve {B}	Robotun hareket etmeyen parçası yani kinematik zincirin ilk halkası
İstasyon Çerçeve {S}	Robotun bütün hareketlerini yaptığı çerçevedir.
Bilek Çerçevesi {W}	Manipülâtörün son uzuvu olup kinematik zincirin son halkasıdır.
Araç Çerçevesi {T}	Robotun hangi işlevi gerçekleştirmesi isteniyorsa ona uygun bir eleman yerleştirilir.
Hedef Çerçevesi {G}	Robotun işlem yapacağı nesnenin üzerindeki çerçevedir.

Tablo 3.1 Robot ve Çalışma Düzlemleri[3]

3.1.1 Robotların Sınıflandırılması

Robotlar temel olarak seri ve paralel robotlar olmak üzere iki gruba ayrılır. Seri robotlar bir dizi eklemler ve bu eklemleri birbirine bağlayan uzuvlardan oluşur. Seri robotlar geniş bir çalışma uzayına ve az sayıda mekanik parçaya sahiptir. Paralel

robotlar ise ana çerçeve ile uç işlevcisi arasında birden fazla paralel uzuvlardan bir araya gelmiştir. Paralel robotlar seri robotlara göre çok sağlam bir mekanik yapıya sahiptir. Bu iki robot türü taşıyacakları kütle için mekanik yapılarının kütlesine oranı bakımından karşılaştırıldıklarında bu oran seri robotlarda çok küçük, paralel robotlarda ise çok büyüktür. Kısacası seri robotlar daha küçük kütleli işlerde, paralel robotlar ise büyük kütleli işlerde kullanılır.



Şekil 3.4 Seri Robot



Şekil 3.5 Paralel Robot

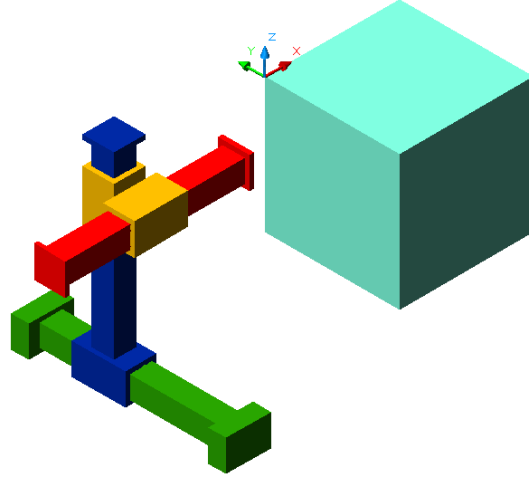
Robotlar serbestlik derecelerine, eklemlerin döndürülmesinde kullanılan güç kaynağına, kontrol yöntemlerine, keskinlik derecelerine göre ve iki harfli kod olmak üzere beş farklı şekilde sınıflandırılabilirler.

3.1.1.1 Serbestlik Derecelerine Göre Robotlar.

Endüstride kullanılan robotlar genellikle altı serbestlik derecesine sahip robotlardır. Robotlar serbestlik derecesine göre sınıflandırılırken ilk üç uzuvun eklem özelliğine bakılır. Örneğin ilk üç uzuv prizmatik ekleme sahipse bu robota Kartezyen (Prismatic Prismatic Prismatic –PPP), ilk uzuv döner ikinci ve üçüncü uzuv prizmatik ekleme sahipse bu robota Silindirik (Revolute Prismatic Prismatic –RPP), ilk iki uzuv döner üçüncü uzuv prizmatik ekleme sahip ve bütün eklemler birbirine paralelse bu robota Scara (RRP), ilk iki uzuv döner ve üçüncü uzuv prizmatik ekleme sahipse bu robota Küresel robot denir. Ayrıca ilk üç uzuvu döner ekleme sahipse bu robota da Döner robot denir.

3.1.1.1.1 Kartezyen Robot Düzenleşimi

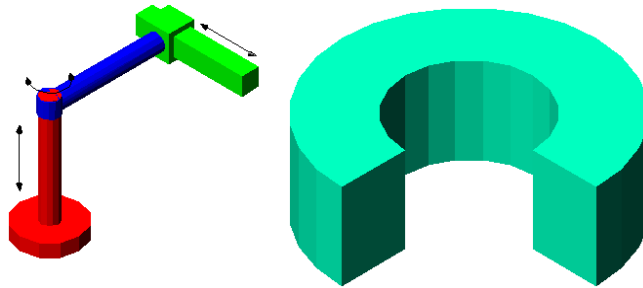
İlk üç eklemi prizmatik eklem olacak şekilde tasarlanmış robota kartezyen manipülatör denir. Kinematik düzenleşimi en basit olan robot türüdür. Mekanik yapıları sağlam olmasına karşın çalışma uzayındaki hareket yetenekleri oldukça düşüktür. Kartezyen tip manipülatörler daha çok büyük hacimli ve ağırlıklı nesnelerin bir yerden bir yere taşınmasında kullanılır.



Şekil 3.6 Kartezyen Robot ve Çalışma Alanı

3.1.1.1.2 Silindirik Robot Düzenleşimi

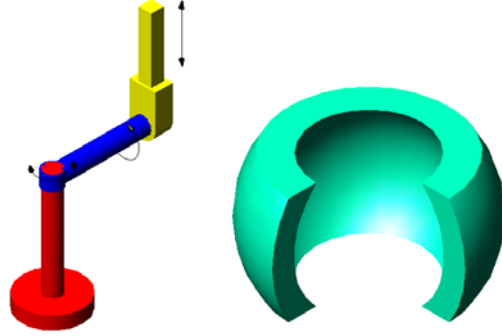
İlk eklemi döner ikinci ve üçüncü eklemi prizmatik tip eklemden oluşan manipülatöre silindirik tip manipülatör denir. Mekanik yapıları sağlam olmasına karşın bilek konum doğruluğu yatay harekete bağlı olarak değişir. Kartezyen robot gibi büyük hacimli ve ağırlıklı nesnelere taşınmasında kullanılır. Prizmatik eklemlerde hareketlendirici olarak genellikle hidrolik silindir kullanılır.



Şekil 3.7 Silindirik Robot ve Çalışma Alanı

3.1.1.1.3 Küresel Robot Düzenleşimi

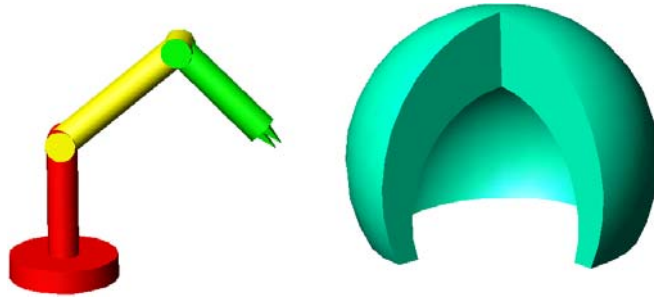
Küresel manipülatörde ilk iki eklem döner üçüncü eklem prizmatik eklemden oluşur. Mekanik yapıları kartezyen robota göre sağlam olmamasına rağmen çalışma uzayındaki hareket yetenekleri oldukça yüksektir.



Şekil 3.8 Küresel Robot ve Çalışma Alanı

3.1.1.1.4 Döner Robot Düzenleşimi

İlk üç eklemi döner ekleme sahip olan manipülatöre dönел manipülatör denir. Çalışma uzayındaki hareket yetenekleri en yüksek olan robottur.



Şekil 3.9 Döner Robot ve Çalışma Alanı

Robot Düzenleşimi	Avantajları	Dezavantajları
Kartezyen	<p>Kinematik denklemleri basittir.</p> <p>Mekanik yapıları çok sağlamdır.</p> <p>Çalışma uzayının her noktasında geniş çaplı hareket kabiliyeti aynıdır.</p> <p>Kinematik yapısı basit olduğundan yeni eleman eklenmesi kolaydır.</p>	<p>Çalışma uzayının boyutu robotun boyutundan küçüktür.</p> <p>Robot kendi ana gövdesine ulaşamaz</p> <p>Prizmatik eklemlerin çalışma ortamındaki tozlardan korunması zordur.</p>
Silindirik	<p>Ana çerçevenin döner olmasından dolayı uç işlevcisinin hızı yüksektir.</p> <p>Kinematik denklemleri basittir.</p> <p>Kartezyen robota göre daha geniş çalışma uzayına sahiptir.</p>	<p>Küresel robotlara göre daha küçük bir çalışma uzayına sahiptir.</p> <p>Geniş çaplı hareket kabiliyeti kol uzunluğuna göre değişir.</p>
Küresel	<p>Çok büyük çalışma uzayına sahiptir.</p>	<p>Kinematik denklemleri karmaşık olduğundan kontrolleri de zordur.</p> <p>Geniş çaplı hareket kabiliyeti her noktada farklıdır.</p>
Dönel	<p>Çok büyük çalışma uzayına sahiptir.</p> <p>Tamamı dönel olan eklemlerin hareket ettirilmesi kolaydır.</p> <p>Çok esnek ve hızlıdırlar.</p>	<p>Geniş çaplı hareket yetenekleri her noktada farklıdır.</p> <p>Kinematik denklemleri karmaşık olduğundan kontrolleri de zordur.</p>

Tablo 3.2 Robot Düzenleşimlerinin Avantaj ve Dezavantajları[3]

3.1.1.2 Kontrol Yöntemlerine Göre Robotlar

Kontrol yöntemlerine göre robotlar iki sınıfa ayrılırlar

3.1.1.2.1 Noktasal Kontrol Edilen Robotlar

Bu tür robotların belirlenmiş bir çalışma alanı yoktur. Serbestlik derecesi 6 ‘dan küçüktür ve genellikle bir nesneyi bir yerden başka bir yere taşımak amacıyla kullanılır.

3.1.1.2.2 Sürekli Yörünge Kontrollü Robotlar

Bu tip robotlar önceden belirlenmiş bir yörüngeyi izleyecek şekilde kontrol edilirler.

3.1.1.3 Robot Hareketlendiricilerinin Kullandığı Güç Kaynağına Göre

Temel olarak elektrik motorları, pnömatik ve hidrolik olmak üzere üç gruba ayrılır.

3.1.1.3.1 Elektrik Motoruyla Hareket Verilenler

Eklemlerin hareket ettirilmesinde özel tasarımı elektrik motorlarının kullanıldığı robotlardır. Bu amaçla dönme açıları 1° ‘nin altına kadar indirilebilen DC Servo ve

adım motorları kullanılır. Dc Servo motorlar sahip oldukları redüktör sayesinde düşük gerilimle yüksek tork sağlarlar. Kontrol edilmesi kolaydır.

3.1.1.3.2 Hidrolik Silindir İle Hareket Verilenler

Bu tip robotlarda eklemlere hidrolik silindirler aracılığıyla hareket verilir. Sağladıkları düşük enerji ile çok yüksek tork sayesinde endüstride çok ağır işlerde kullanılır. Performansları doğrusal olmadığından elektrik motorlarına göre kontrolleri daha zordur.

3.1.1.3.3 Pnömatik Silindir İle Hareket Verilenler

Bu tip robotlarda hidrolik silindirli robotlara benzerler ancak pnömatik silindirde hava kullanıldığı için silindirin yeterli ataletini sağlayacak hava basıncının sağlanamamasından dolayı kontrolleri zordur ancak yapısı basittir.

3.1.1.4 Keskinlik Derecelerine Göre Robotlar

Çözünürlük, doğruluk ve yenilenebilirlik olarak üç gruba ayrılır.

3.1.1.4.1 Çözünürlük

Çok küçük bir yer değiştirmeyi gerçekleştirme yeteneğidir.

3.1.1.4.2 Doğruluk

Her robot için yazılmış bir program vardır. Robotun doğruluğu da uç işlevcisinin bu programı gerçekleştirme derecesidir.

3.1.1.4.3 Yenilenebilirlik

Yenilenebilirlik robotun uç işlevcisinin birçok işlevi gerçekleştirdikten sonra tekrar aynı noktaya gelebilme özelliğidir.

3.1.1.5 İki Harfli Kodlamayla Yapılan Sınıflandırma

Huang ve Milenkovic robot türlerini tanımlamak için iki harften oluşan bir kod geliştirmişlerdir. Buna göre ilk harf birinci eklemnin özelliğini ve ikinci eklem göre nasıl döndüğünü açıklamakta, ikinci harf ise üçüncü eklemi ve ikinci eklem ile üçüncü eklem arasındaki ilişkiyi tanımlar.

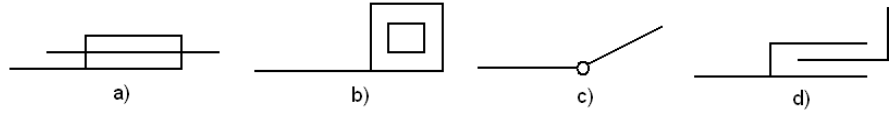
Kullanılan Harf ve Semboller

S: Kayma (Şekil 3.10 a)

C: Kayma ekseninde dik dönme (Şekil 3.10 b)

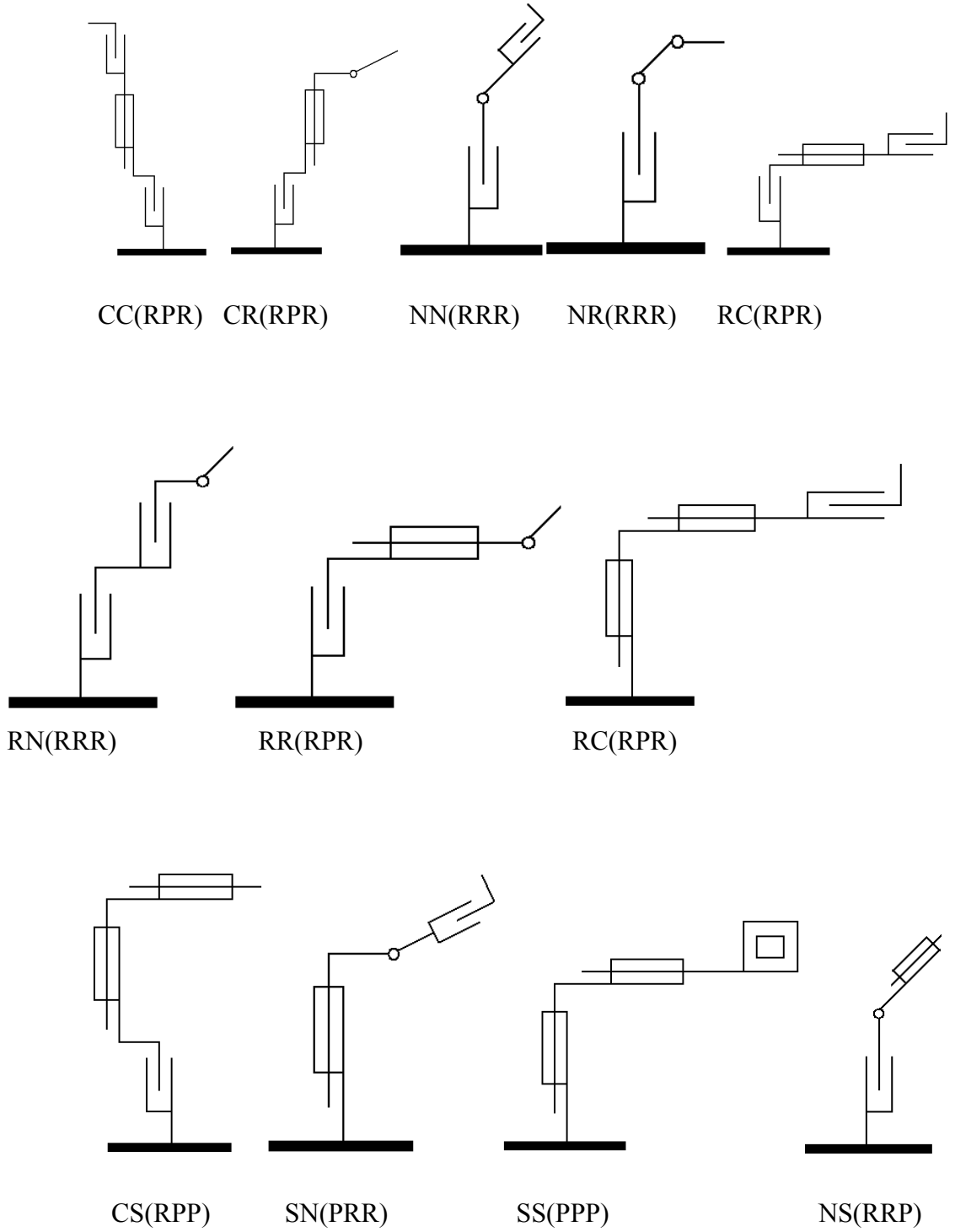
N: Dönme ekseninde dik dönme (Şekil 3.10 c)

R: Kayma eksenine dik dönme veya dönme eksenine paralel dönme (Şekil 3.10 d)



Şekil 3.10 İki Harfli Kodlamada Kullanılan Semboller

Huang ve Milenkovic robot uzuvları için 16 adet 2 harf kombinasyonu kullanmıştır. Bu birleşimlerin tamamı kullanışlı ve farklı değildir. Kullanışlı bir uzuv üç boyutlu uzayda geniş çaplı hareket yeteneğine sahip olmalı, farklılık ise her bir uzuvun diğer kategoriler arasında farklı olmasıdır.



Şekil 3.11 Robotları İki Harfli Kodlamada Tanımlayan Sembolik Şekilleri

Huang ve Milenkovic bu kodlardan CN, NC, RS ve SR 'yi kullanışlı ve farklı bulmamıştır. Buna rağmen RS endüstride sıklıkla kullanılan SCARA robotu temsil eder. RS ve SR robotlarının kullanışlı ve farklı bulunmamasına rağmen endüstride sıklıkla

kullanılırlar. Bu iki robotun kullanışlı ve farklı kabul edilmemesinin sebebi CS robotu ile aynı alanı taramalarından kaynaklanmaktadır.[3]

3.1.2 Benzetim

Benzetim yani simülasyon fiziksel bir etkinliği gerçekleştirmeden önce bilgisayar ortamında gerekli programlar yardımıyla fiziksel etkinliği canlandırma işlemidir. Robot programları gerçek robotlar üzerinde denenmeden önce benzetim programları yardımıyla test edilir. Programların test edilmesinin avantajları vardır. Eğer robot programı yazıldıktan sonra benzetim programlarıyla test edilmeden robota yüklenirse, robot programındaki yanlışlıklar ve eksiklikler nedeniyle robot çevresine ve kendine zarar verebilir. Bu nedenle robot programları yazıldıktan sonra benzetim programları ile test edilmesi zaman ve para tasarrufu sağlar. İyi bir robot benzetim programı uzuvların konumlarını, yönelimlerini ve çevresini iyi bir şekilde modelleyebilmelidir.[3]

3.1.2.1 Robot Benzetim Programlarının Sınıflandırılması

Çevrimiçi ve Çevrimdışı Programlama Olmak Üzere İki Gruba Ayrılır.

3.1.2.1.1 Çevrimiçi Programlama

Çevrimiçi programlama robot üretim aşamasındayken gerçekleştirilir. Robotun uç işlevcisini, bir kullanıcı arabirimi vasıtasıyla eş zamanlı olarak istenen noktaya ulaştırmak için kullanılır.

3.1.2.1.2 Çevrimdışı Programlama

Çevrimdışı programlamada robot üretime sokulmadan önce programlama yapılır ve programlamadan sonra robot üretime sokulur. Bu tür robotlar sürekli tekrar eden işlemlerde kullanılır. Robot başka bir işlem gerçekleştirmek için programlanacaksa önce üretimden çıkarılır yeni program yüklenir ve tekrar üretime sokulur.

3.1.3 Genel Tanımlamalar

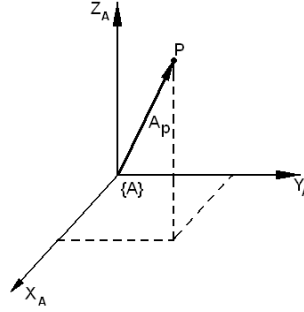
Robotun konum ve yöneliminin belirlenebilmesi için üç boyutlu uzayda robotun kendisinde dahil olmak üzere her nesneye bir koordinat sisteminin yerleştirilmesi gerekir. Nesnelere ve bu nesnelere yerleştirilen koordinat sistemleri evrensel çerçeve içinde bulunur. Yapacağımız tüm tanımlamalar evrensel çerçeveye veya kartezyen koordinat sistemine göre gerçekleştirilecektir.

3.1.3.1 Konum

Bir nokta, koordinat sistemi tanımlanmak suretiyle evrensel çerçeve içerisinde her hangi bir yere konumlanabilir. Üç boyutlu uzayda bir nokta seçilen koordinat sisteminin merkezine göre tanımlanmış 3×1 boyutlu bir vektörle gösterilir ve bu vektöre konum vektörü denir. Bu vektör hangi koordinat sistemine göre tanımlanmışsa ona göre isimlendirilir.

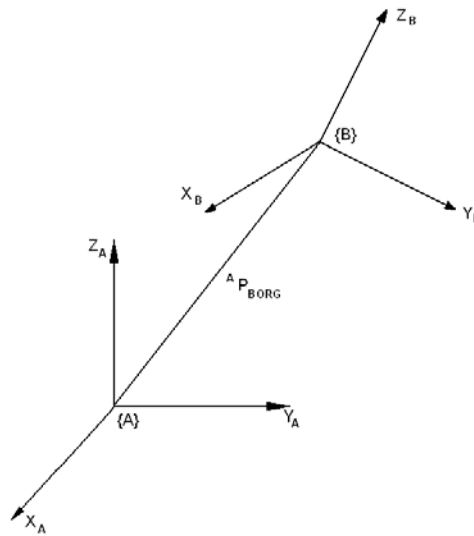
Örneğin P noktasının $\{A\}$ koordinat sistemine göre konumu ${}^A P$ şeklinde bir vektörle ifade edilir. Ve matematiksel olarak aşağıdaki gibi gösterilir.

$${}^A P = \begin{bmatrix} {}^A P_x \\ {}^A P_y \\ {}^A P_z \end{bmatrix} \quad (1.1)$$

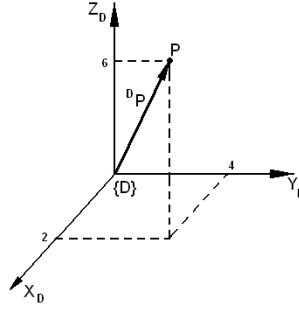
Şekil 3.12 ${}^A P$ Vektörü

{A} ve {B} koordinat sistemlerinin merkezleri arasındaki uzaklık A noktasıyla uç işlevcisi arasındaki uzaklıktır ve ${}^A P_{BORG}$ şeklinde gösterilir. ${}^A P_{BORG}$ Vektörü ${}^A P_{BORG_x}$, ${}^A P_{BORG_y}$ ve ${}^A P_{BORG_z}$ bileşenlerinden oluşur ve matematiksel olarak aşağıdaki gibi gösterilir.

$${}^A P_{BORG} = \begin{bmatrix} {}^A P_{BORG_x} \\ {}^A P_{BORG_y} \\ {}^A P_{BORG_z} \end{bmatrix} \quad (1.2)$$

Şekil 3.13 ${}^A P_{BORG}$ Vektörü

Örnek –1.1 Üç boyutlu uzayda $P=(2,4,6)$ noktasının $\{D\}$ koordinat sistemine göre konumu şekildeki gibi gösterilir.

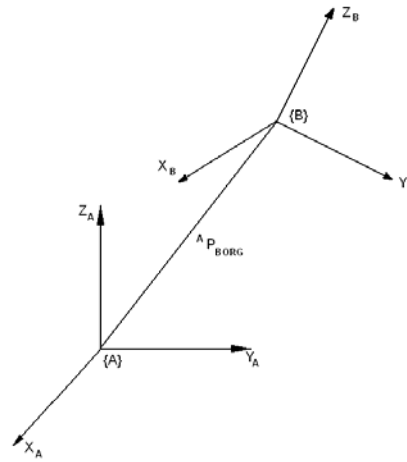


Şekil 3.14 ${}^D P$ Vektörü

3.1.3.2 Yönelim

Yönelim bir koordinat sisteminin başka bir koordinat sistemine göre dönme miktarıdır ve 3×3 boyutlu bir matrisle ifade edilir, bu matrise de dönme matrisi denir. Bir katı cisim başka bir referans koordinat sistemine göre tanımlamak için öncelikle katı cisim koordinat sistemi yerleştirilir. [3]

Aşağıdaki şekildeki uç işlevcisine $\{B\}$ koordinat sistemi yerleştirilerek $\{A\}$ koordinat sistemine göre yönelimi tanımlanır.

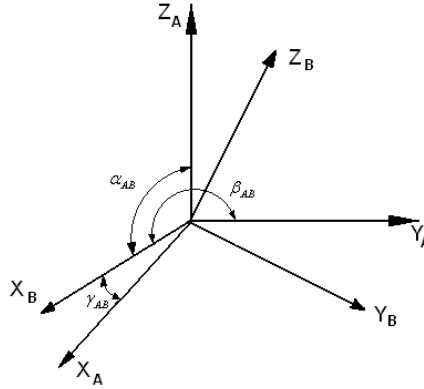


Şekil 3.15 $\{B\}$ koordinat sisteminin $\{A\}$ koordinat sistemine göre yönelimi

Uç işlevcisine yerleştirilen {B} koordinat sistemini, {A} koordinat sistemi cinsinden ifade etmek için birim vektörler kullanılır ve {B} koordinat sisteminin birim vektörleri cinsinden ifadesi $\{B\} = \hat{X}_B, \hat{Y}_B, \hat{Z}_B$, {B} koordinat sisteminin vektörlerinin {A} koordinat sistemi cinsinden ifadesi ise $\{A\} = {}^A\hat{X}_B, {}^A\hat{Y}_B, {}^A\hat{Z}_B$ bu şekilde elde edilen 3x3 boyutundaki matrise dönme matrisi denir ve R ile ifade edilir. {B} koordinat sisteminin yönelimini {A} koordinat sistemi cinsinden ifadesi ise ${}^A_B R$ ile gösterilir. {B} koordinat sisteminin yönelimini {A} koordinat sistemine göre X,Y ve Z eksenlerindeki dönme miktarlarını gösterir.

$${}^A_B R = \begin{bmatrix} \hat{X}_B & \hat{Y}_B & \hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (1.3)$$

{B} koordinat sistemi ile {A} koordinat sistemleri aşağıdaki gibi çakışık olduğunda {B} koordinat sisteminde yer alan \hat{X}_B birim vektörünün yönelimi {A} koordinat sistemine göre ${}^A\hat{X}_B$ şeklinde gösterilir ve aşağıdaki gibi tanımlanır.



Şekil 3.16 \hat{X}_B birim vektörünün {A} koordinat sistemine göre yönelimi

$${}^A\hat{X}_B = \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} = \begin{bmatrix} |X_B||X_A| \cos \gamma_{AB} \\ |X_B||Y_A| \cos \beta_{AB} \\ |X_B||Z_A| \cos \alpha_{AB} \end{bmatrix} = \begin{bmatrix} \hat{X}_B * \hat{X}_A \\ \hat{X}_B * \hat{Y}_A \\ \hat{X}_B * \hat{Z}_A \end{bmatrix} \quad (1.4)$$

Bu ifadeden yararlanılarak aşağıdaki homojen dönüşüm matrisi elde edilir.

$${}^A_B R = \begin{bmatrix} \hat{X}_B, \hat{Y}_B, \hat{Z}_B \end{bmatrix} = \begin{bmatrix} \hat{X}_B * \hat{X}_A & \hat{Y}_B * \hat{X}_A & \hat{Z}_B * \hat{X}_A \\ \hat{X}_B * \hat{Y}_A & \hat{Y}_B * \hat{Y}_A & \hat{Z}_B * \hat{Y}_A \\ \hat{X}_B * \hat{Z}_A & \hat{Y}_B * \hat{Z}_A & \hat{Z}_B * \hat{Z}_A \end{bmatrix} \quad (1.5)$$

{A} koordinat sistemini {B} koordinat sistemi cinsinden tanımlayan ${}^B_A R$ dönüşüm matrisi, ${}^A_B R$ matrisinin transpozese yani devriğine eşittir.

$${}^B_A R = {}^A_B R^T \quad (1.6)$$

${}^B_A R$ Dönme matrisinin tersi ${}^B_A R^{-1}$ olsun. ${}^B_A R$ Matrisiyle ${}^B_A R^{-1}$ matrisinin çarpımı birim matrisi verir.

${}^B_A R^{-1} * {}^B_A R = I$ I birim matrisi ifade etmektedir. Eşitliğin her iki tarafını ${}^B_A R^T$ matrisiyle çarparsak

$${}^B_A R^T * {}^B_A R^{-1} * {}^B_A R = I * {}^B_A R^T \quad (1.7)$$

$${}^B_A R^T * {}^B_A R = \begin{bmatrix} \hat{X}_A \\ \hat{Y}_A \\ \hat{Z}_A \end{bmatrix} \begin{bmatrix} \hat{X}_A & \hat{Y}_A & \hat{Z}_A \end{bmatrix} \quad (1.8)$$

$${}^B_A R^T * {}^B_A R = \begin{bmatrix} \hat{X}_A \cdot \hat{X}_A & \hat{X}_A \cdot \hat{Y}_A & \hat{X}_A \cdot \hat{Z}_A \\ \hat{Y}_A \cdot \hat{X}_A & \hat{Y}_A \cdot \hat{Y}_A & \hat{Y}_A \cdot \hat{Z}_A \\ \hat{Z}_A \cdot \hat{X}_A & \hat{Z}_A \cdot \hat{Y}_A & \hat{Z}_A \cdot \hat{Z}_A \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I \quad (1.9)$$

$${}^B_A R^{-1} * {}^B_A R = I \text{ ifadesini} \quad (1.10)$$

$${}^B R^T * {}^B R^{-1} * {}^B R = I * {}^B R^T \text{ ifadesinde yerine koyarsak} \quad (1.11)$$

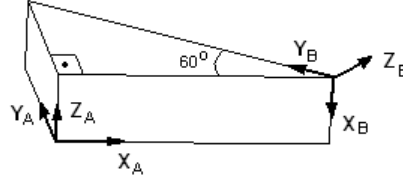
$${}^B R^{-1} = {}^B R^T \text{ Elde edilir.} \quad (1.12)$$

Bu eşitlikten de bir matrisin tersini almak istediğimizde o matrisin sadece devriğini almak yeterlidir.

{B} koordinat sistemi ile {A} koordinat sisteminin merkezleri aşağıdaki gibi çakışık olduğunda {B} koordinat sistemine göre tanımlanmış bir noktayı {A} koordinat sistemine göre tanımlamak için aşağıdaki matematiksel ifade kullanılır.

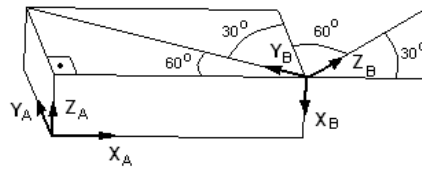
$${}^A P = {}^A R {}^B P \quad (1.13)$$

Örnek -1.2 Aşağıdaki şekilde verilen {B} koordinat sisteminin {A} koordinat sistemine göre yönelimini veren dönme matrisini bulalım.



Şekil 3.17 {B} koordinat sisteminin {A} koordinat sistemine göre yönelimi

{A} koordinat sistemi öncelikle Y eksenini etrafında 90°, daha sonrada X eksenini etrafında 30° döndürüldüğünde {B} koordinat sistemi elde edilir.



Şekil 3.18 {B} koordinat sistemi

Buna göre {B} koordinat sisteminin {A} koordinat sistemine göre yönelimini veren dönme matrisi aşağıdaki gibi olur.

$${}^A_B R = \begin{bmatrix} \cos 90 & \cos 120 & \cos 30 \\ \cos 90 & \cos 30 & \cos 60 \\ \cos 180 & \cos 90 & \cos 90 \end{bmatrix} = \begin{bmatrix} 0 & -0,5 & 0,866 \\ 0 & 0,866 & 0,5 \\ -1 & 0 & 0 \end{bmatrix} \quad (1.14)$$

3.1.3.3 Öteleme Operatörü (Konum Vektörü)

Bir öteleme vektörü kullanılarak bir nokta belli bir koordinat sistemine göre tanımlanabilir.

3.1.3.4 Dönme Operatörü

Bir koordinat sisteminin herhangi bir eksen etrafında döndürülmesi iki veya üç boyutlu uzayda ifade edilir.[3]

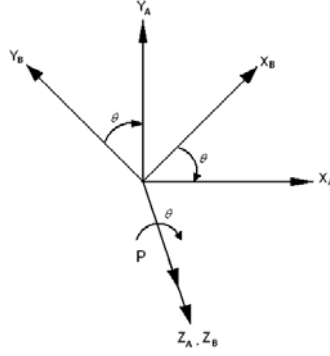
Bir P vektörünün sırasıyla Z, Y ve X eksenleri etrafında θ açısıyla döndürülmesinden elde edilen dönme matrisleri aşağıdaki gibidir.

P vektörünün sırasıyla Z eksen etrafında θ açısıyla döndürülmesinden elde edilen dönme matrisi

$${}^A_B R = \begin{bmatrix} \hat{X}_B, \hat{Y}_B, \hat{Z}_B \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.15)$$

Konum vektörü $\begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix}$ olan P vektörünün homojen dönüşüm matrisi

$$R_Z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & P_X \\ \sin \theta & \cos \theta & 0 & P_Y \\ 0 & 0 & 1 & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.16)$$



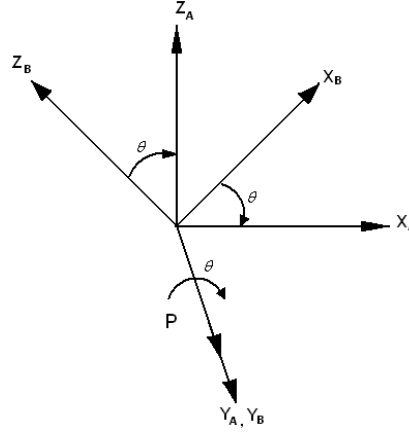
Şekil 3.19 P vektörünün Z eksenini etrafında θ açısıyla döndürülmesi

P vektörünün sırasıyla Y eksenini etrafında θ açısıyla döndürülmesinden elde edilen dönme matrisi

$${}^A_B R = \begin{bmatrix} \hat{X}_B, \hat{Y}_B, \hat{Z}_B \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (1.17)$$

Konum vektörü $\begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix}$ olan P vektörünün homojen dönüşüm matrisi

$$R_Y = \begin{bmatrix} \cos \theta & 0 & \sin \theta & P_X \\ 0 & 1 & 0 & P_Y \\ -\sin \theta & 0 & \cos \theta & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.18)$$



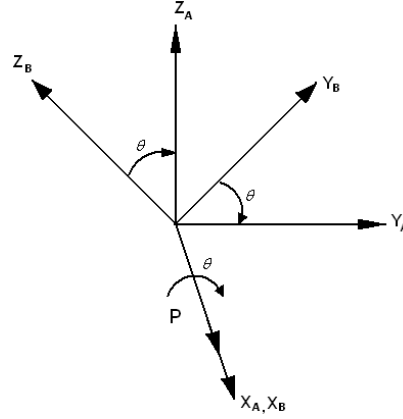
Şekil 3.20 P vektörünün Y eksenini etrafında θ açısıyla döndürülmesi

P vektörünün sırasıyla X eksenini etrafında θ açısıyla döndürülmesinden elde edilen dönme matrisi

$${}^A_B R = \begin{bmatrix} \hat{X}_B, \hat{Y}_B, \hat{Z}_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (1.19)$$

Konum vektörü $\begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix}$ olan P vektörünün homojen dönüşüm matrisi

$$R_X = \begin{bmatrix} 1 & 0 & 0 & P_X \\ 0 & \cos \theta & -\sin \theta & P_Y \\ 0 & \sin \theta & \cos \theta & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.20)$$



Şekil 3.21 P vektörünün X eksenine etrafında θ açısıyla döndürülmesi

3.1.3.5 Homojen Dönüşüm Matrislerinin İleri Yönlü Çarpılması

{B} koordinat sisteminin {A} koordinat sistemine göre tanımlanan dönüşüm matrisi aşağıdaki gibi ifade edilirse

$${}^A_B T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.21)$$

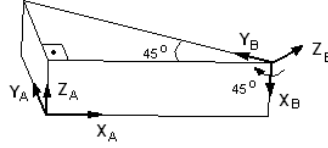
Dönüşüm matrislerinin ileri yönlü çarpma etkisini anlamak için ${}^A_B T$ dönüşüm matrisini başka bir dönüşüm matrisi $R_Y(\theta)$ ile çarpalım.

$$\begin{aligned} {}^A_B T \cdot R_Y(\theta) &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta & P_x \\ 0 & 1 & 0 & P_y \\ -\sin \theta & 0 & \cos \theta & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} r_{11} \cos \theta - r_{13} \sin \theta & r_{12} & r_{11} \sin \theta + r_{13} \cos \theta & p_x \\ r_{21} \cos \theta - r_{23} \sin \theta & r_{22} & r_{21} \sin \theta + r_{23} \cos \theta & p_y \\ r_{31} \cos \theta - r_{33} \sin \theta & r_{32} & r_{31} \sin \theta + r_{33} \cos \theta & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

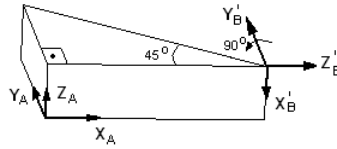
Çarpım sonucuna göre Y eksenine ait birim vektör çarpmadan önceki ${}^A T_B$ matrisindeki Y eksenine ait vektörle aynı çıkmıştır. Aynı durum konum vektörü içinde geçerlidir. Öyleyse bir dönüşüm matrisini başka bir dönüşüm matrisi ile ileri yönlü bir çarpma işlemine sokarsak öteleme veya dönme işlemi hareket eden koordinat sistemine göre gerçekleşir.[3]

Örnek –1.3 Şekildeki {B} koordinat sistemine göre tanımlanmış {A} koordinat sisteminin dönüşüm matrisini ileri yönlü çarpma yöntemi ile bulalım.

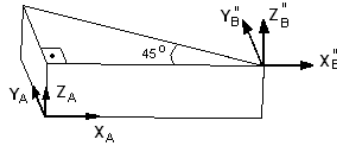
{B} koordinat sisteminin {A} koordinat sistemine dönüşebilmesi için öncelikle {B} koordinat sisteminin X eksenini -45° (Şekil 3.22) ve daha sonrada Y eksenini $+90^\circ$ (Şekil 3.23) döndürülmesi gerekir. Bu şekilde elde edilen dönüşüm matrislerinden ilki $R_X(-45)$ çarpma işleminden sonra ikinci dönüşüm matrisi $R_Y(+90)$ çarpma işleminden önce yazılır. Burada {B} koordinat sistemi döndürülerek {A} koordinat sistemi (Şekil 3.24) elde edilir.



Şekil 3.22 {B} koordinat sisteminin X eksenini -45° döndürülmesi



Şekil 3.23 {B} koordinat sisteminin Y eksenini $+90^\circ$ döndürülmesi



Şekil 3.24 {B} koordinat sistemi döndürülerek elde edilen {A} koordinat sistemi

$$\text{Matematiksek ifadesi } {}^A_B R = R_Y(+90) * R_X(-45) \quad (1.22)$$

$$\begin{aligned} {}^A_B R &= \begin{bmatrix} \cos 90 & 0 & \sin 90 \\ 0 & 1 & 0 \\ -\sin 90 & 0 & \cos 90 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-45) & -\sin(-45) \\ 0 & \sin(-45) & \cos(-45) \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0,707 & 0,707 \\ 0 & -0,707 & 0,707 \end{bmatrix} = \begin{bmatrix} 0 & -0,707 & 0,707 \\ 0 & 0,707 & 0,707 \\ -1 & 0 & 0 \end{bmatrix} \end{aligned} \quad (1.23)$$

3.1.3.6 Homojen Dönüşüm Matrislerinin Önden Çarpılması

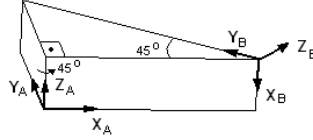
Dönüşüm matrislerinin önden çarpma etkisini anlamak için ${}^A_B T$ dönüşüm matrisini başka bir dönüşüm matrisi $R_Y(\theta)$ ile çarpalım.

$$\begin{aligned} R_Y(\theta) \cdot {}^A_B T &= \begin{bmatrix} \cos \theta & 0 & \sin \theta & P_X \\ 0 & 1 & 0 & P_Y \\ -\sin \theta & 0 & \cos \theta & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} r_{11} \cos \theta + r_{31} \sin \theta & r_{11} \cos \theta + r_{32} \sin \theta & r_{11} \sin \theta + r_{33} \cos \theta & p_x \cos \theta + p_z \sin \theta \\ r_{21} & r_{22} & r_{23} & p_y \\ -r_{11} \cos \theta + r_{31} \sin \theta & -r_{12} \cos \theta + r_{32} \sin \theta & -r_{13} \cos \theta + r_{33} \sin \theta & -p_x \sin \theta + p_z \cos \theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (1.24)$$

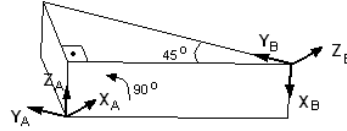
Bu çarpım sonucuna göre 2 satır çarpmadan önceki ${}^A_B T$ matrisindeki 2. satırla aynı çıkmıştır. Öyleyse bir dönüşüm matrisini başka bir dönüşüm matrisi ile önden çarpma işlemine sokarsak öteleme veya dönme işlemi sabit referans koordinat sistemine göre gerçekleşir.[3]

Örnek -1.4 Bir önceki örnekteki dönüşüm matrislerini önden çarpma işlemine sokalım.

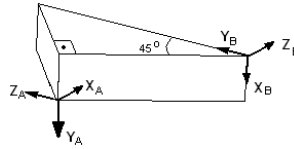
{A} koordinat sisteminin {B} koordinat sistemine dönüşebilmesi için öncelikle {A} koordinat sisteminin Z ekseninde 45° (Şekil 3.25) ve daha sonrada Y ekseninde 90° (Şekil 3.26) döndürülmesi gerekir. Bu şekilde elde edilen dönüşüm matrislerinden ilki $R_z(+45)$ çarpma işleminden önce ikinci dönüşüm matrisi $R_y(+90)$ çarpma işleminden sonra yazarak önden çarpma işlemine sokalım. Burada {B} koordinat sistemi döndürülerek {A} koordinat sistemi (Şekil 3.27) elde edilir.



Şekil 3.25 {A} koordinat sisteminin Z ekseninde 45° döndürülmesi



Şekil 3.26 {A} koordinat sisteminin Y ekseninde 90° döndürülmesi



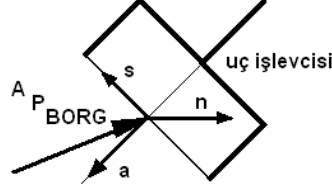
Şekil 3.27 {B} koordinat sistemi döndürülerek elde edilen {A} koordinat sistemi

$$\text{Matematiksek ifadesi } {}^A_B R = R_z(45) * R_y(+90) \quad (1.25)$$

$$\begin{aligned} {}^A_B R &= \begin{bmatrix} \cos(45) & -\sin(45) & 0 \\ \sin(45) & \cos(45) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos 90 & 0 & \sin 90 \\ 0 & 1 & 0 \\ -\sin 90 & 0 & \cos 90 \end{bmatrix} \\ &= \begin{bmatrix} 0,707 & -0,707 & 0 \\ 0,707 & 0,707 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -0,707 & 0,707 \\ 0 & 0,707 & 0,707 \\ -0,707 & 0 & 0 \end{bmatrix} \end{aligned} \quad (1.26)$$

3.1.3.7 Dönüşüm Matrisinin Özellikleri

Dönüşüm matrisi dönme matrisi ve konum vektöründen oluşur.



Şekil 3.28 Uç İşlevçisi

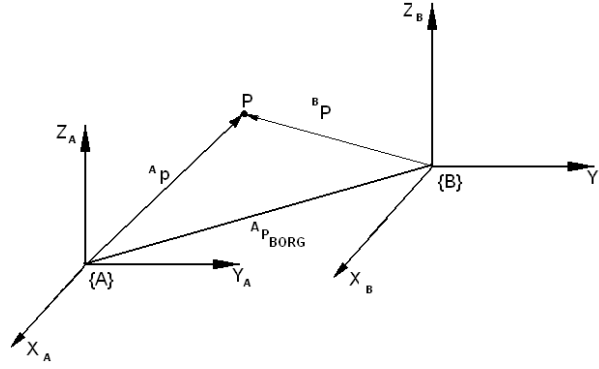
Şekilde görüldüğü gibi bir uç işlevçisi bir hedefe yöneldiği zaman bu yönelimi uç işlevçisinin normal vektörü $n = [n_x \ n_y \ n_z]^T$, kayma vektörü $s = [s_x \ s_y \ s_z]^T$ ve yaklaşım vektörü $a = [a_x \ a_y \ a_z]^T$ olmak üzere üç vektörle ifade edilir.

$$\begin{aligned}
 {}^A_B T &= \begin{bmatrix} {}^A_B R & {}^A P_{BORG} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.27)
 \end{aligned}$$

${}^A_B T$ Dönüşüm matrisi aşağıdaki gibi ifade edilsin

$${}^A_B T = \begin{bmatrix} {}^A_B R & {}^A P_{BORG} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.28)$$

${}^A_B T$ Matrisinin tersini ${}^A_B T^{-1}$ şeklinde gösterelim. ${}^A_B T^{-1}$ Matrisinin tersini bulmak bilinen yöntemlerle oldukça zordur. Daha önceden öğrendiğimiz gibi bir dönme matrisinin tersi o matrisin devriğine eşittir. ${}^A_B R^{-1} = {}^B_A R = {}^A_B R^T$ Ancak bu durum konum vektörü için geçerli değildir.



Şekil 3.29 {A} koordinat sistemine göre {B} koordinat sistemi cinsinden ${}^B({}^A P)$ Vektörü

Konum vektörünün ${}^A T_B$ matrisinin bir fonksiyonu şeklinde yazılması gerekir.

P noktasının {A} koordinat sistemine göre konumu

$${}^A P = {}^A_B R \cdot {}^B P + {}^A P_{BORG}, \quad (1.29)$$

{B} koordinat sistemine göre konumu ise

$${}^B P = {}^B_A R \cdot {}^A P + {}^B P_{AORG} \quad (1.30)$$

Şeklinde ifade edilir. {A} koordinat sistemine göre {B} koordinat sistemi cinsinden ${}^B({}^A P) = {}^B_A R \cdot {}^A P + {}^B P_{AORG}$ (1.31)

Şeklinde ifade edilir.

${}^A P = {}^A P_{BORG}$ olsun buna göre ${}^A P = {}^A_B R \cdot {}^B P + {}^A P_{BORG}$ denkleminde yerine koyarsak

${}^B({}^A P_{BORG}) = {}^B_A R \cdot {}^A P + {}^B P_{AORG}$ elde edilir. ${}^A P_{BORG}$ {A} koordinat sistemi ile {B} koordinat sistemi merkezleri arasındaki uzaklık olduğuna göre bu uzaklığı {B} koordinat sistemi cinsinden tanımladığımızda ${}^B({}^A P_{BORG}) = 0$ olur. Elde edilen formülde yerine koyarsak

$$\begin{aligned} 0 &= {}^B_A R \cdot {}^A P_{BORG} + {}^B P_{AORG} \\ {}^B P_{AORG} &= -{}^B_A R \cdot {}^A P_{BORG} \end{aligned} \quad (1.32)$$

$${}^B R = {}^A R^T \quad (1.33)$$

olduğu bilindiğine göre sonuç olarak aşağıdaki denklem elde edilir.

$${}^B P_{AORG} = -{}^A R^T \cdot {}^A P_{AORG} \quad (1.34)$$

Bulunan ${}^B R = {}^A R^T$ ve ${}^B P_{AORG} = -{}^A R^T \cdot {}^A P_{AORG}$ denklemleri

$${}^A T^{-1} = \begin{bmatrix} {}^A R & {}^A P_{BORG} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.35)$$

Denkleminde yerine koyarsak

$${}^A T^{-1} = \begin{bmatrix} {}^A R^T & -{}^A R^T \cdot {}^A P_{BORG} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.36)$$

Denklemleri elde edilir.

Bu çözümlerden anlaşıldığı gibi bir dönüşüm matrisinin tersini almak için öncelikle dönme matrisinin devriği alınır ve dönme matrisi kısmına yazılır. Elde edilen dönme matrisinin devriği ile konum vektörü çarpılıp sonucun -1 ile çarpımı konum vektörü kısmına yazılır. [3]

Örnek -1.5

$${}^A T = \begin{bmatrix} 0,707 & 0 & 0,707 & 2 \\ 0 & 1 & 0 & 1 \\ -0,707 & 0 & 0,707 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ Matrisinin tersini bulalım} \quad (1.37)$$

$${}^A R = \begin{bmatrix} 0,707 & 0 & 0,707 \\ 0 & 1 & 0 \\ -0,707 & 0 & 0,707 \end{bmatrix} \text{ Dönüşüm matrisinin dönme matrisi} \quad (1.38)$$

$${}^A_B R^T = \begin{bmatrix} 0,707 & 0 & -0,707 \\ 0 & 1 & 0 \\ 0,707 & 0 & 0,707 \end{bmatrix} \text{ Bulunur} \quad (1.39)$$

$$-{}^A_B R^T \cdot {}^A P_{BORG} = - \begin{bmatrix} 0,707 & 0 & -0,707 \\ 0 & 1 & 0 \\ 0,707 & 0 & 0,707 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -2,828 \end{bmatrix} \quad (1.40)$$

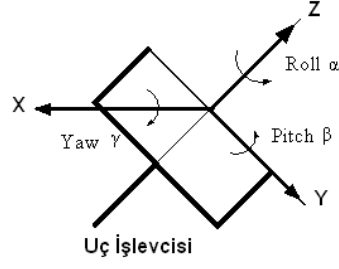
$${}^A_B T^{-1} = \begin{bmatrix} 0,707 & 0 & -0,707 & 0 \\ 0 & 1 & 0 & -1 \\ 0,707 & 0 & 0,707 & -2,828 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ elde edilir.} \quad (1.41)$$

3.1.4 Özel Yönelim Tanımlamaları

Şimdiye kadar koordinat sistemlerinin birbirlerine göre yönelimini 3x3 boyutundaki dönme matrisleri ile tanımladık. Ancak koordinat sistemleri arasındaki yönelimin tanımlanması için Roll-Pitch-Yaw ve Euler Açılı Sistemi olmak üzere en fazla iki farklı yöntem kullanılır.

3.1.4.1 Roll-Pitch-Yaw (XYZ Sabit) Açılı seti

Roll-Pitch-Yaw açılı setinde dönme işlemi hareket etmeyen sabit koordinat çerçevesine göre gerçekleştiğinden bu yöntemde sabit açılı sistemi veya Roll-Pitch-Yaw açılı sistemi denir. Eğer insanın eli avuç içi aşağı bakmak üzere ileri yönde yere paralel tutulursa Roll elin kendi eksenini etrafından döndürülmesi, Pitch elin aşağı-yukarı hareket ettirilmesi, Yaw ise elin sağa sola hareket ettirilmesidir.[3]



Şekil 3.30 Roll-Pitch-Yaw Açısı seti

{B} koordinat sisteminin {A} koordinat sistemine göre dönme işlemi XYZ sabit açı sistemine göre aşağıdaki gibi bulunur.

İlk önce merkezleri çakışık olan {B} koordinat sistemi şekilde görüldüğü gibi X_A eksenini boyunca γ , Y_A eksenini boyunca β ve Z_A eksenini boyunca α açısıyla döndürülür. X_A, Y_A ve Z_A eksenleri boyunca dönme işlemi sabit {A} koordinat sistemine göre ifade edilir. Matematiksel olarak ifadesi ise

$$R_{XYZ}(\gamma, \beta, \alpha) = R_Z(\alpha)R_Y(\beta)R_X(\gamma) \quad (1.42)$$

$$R_{XYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \quad (1.43)$$

$$= \begin{bmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{bmatrix}$$

{B} koordinat sistemine göre {A} koordinat sisteminin sabit açı sistemindeki α , β ve γ açı kümesi aşağıdaki gibi olur.

$$R_{XYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (1.44)$$

$$= \begin{bmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{bmatrix}$$

Yukarıdaki her iki matristeki (1,1), (2,1) ve (3,1) hücrelerindeki elemanlar birbirlerine eşitlendikten sonra (1,1) ve (2,1) matris elemanlarının kareleri alınıp toplanırsa

$$r_{11} = \cos \alpha \cos \beta \quad (1.45)$$

$$\text{ve } r_{21} = \sin \alpha \cos \beta \quad (1.46)$$

$$\begin{aligned} r_{11}^2 + r_{21}^2 &= \cos^2 \alpha \cos^2 \beta + \sin^2 \alpha \cos^2 \beta = (\cos^2 \alpha + \sin^2 \alpha) \cos^2 \beta = \cos^2 \beta \\ r_{11}^2 + r_{21}^2 &= \cos^2 \beta \\ \cos \beta &= \pm \sqrt{r_{11}^2 + r_{21}^2} \end{aligned} \quad (1.47)$$

$$\begin{aligned} \frac{\sin \beta}{\cos \beta} &= \frac{-r_{31}}{\sqrt{r_{11}^2 + r_{21}^2}} \\ \beta &= \arctan \left(\frac{-r_{31}}{\sqrt{r_{11}^2 + r_{21}^2}} \right) \end{aligned} \quad (1.48)$$

(1,1) ve (2,1) matris elemanlarını $\cos \beta$ 'ya bölerek α açısı elde edilir.

$$\frac{\sin \alpha \cos \beta}{\cos \beta} = \frac{r_{21}}{\cos \beta} \quad \sin \alpha = \frac{r_{21}}{\cos \beta} \quad (1.49)$$

$$\frac{\cos \alpha \cos \beta}{\cos \beta} = \frac{r_{11}}{\cos \beta} \quad \cos \alpha = \frac{r_{11}}{\cos \beta}$$

$$\alpha = \arctan \left(\frac{r_{21}}{\cos \beta} / \frac{r_{11}}{\cos \beta} \right) = \arctan(r_{21}/r_{11}) \quad (1.50)$$

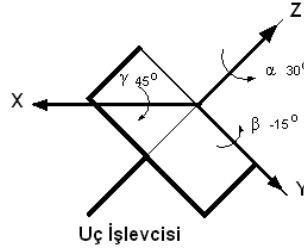
(3,2) ve (3,3) matris elemanlarını $\cos \beta$ 'ya bölerek α açısı elde edilir.

$$\frac{\cos \beta \sin \gamma}{\cos \beta} = \frac{r_{32}}{\cos \beta} \quad \sin \gamma = \frac{r_{32}}{\cos \beta} \quad (1.51)$$

$$\frac{\cos \gamma \cos \beta}{\cos \beta} = \frac{r_{33}}{\cos \beta} \quad \cos \alpha = \frac{r_{33}}{\cos \beta}$$

$$\gamma = \arctan\left(\frac{r_{32}}{\cos \beta} / \frac{r_{33}}{\cos \beta}\right) = \arctan(r_{32}/r_{33}) \quad (1.52)$$

Örnek –1.6 Robotun uç işlevcisi sabit koordinat sisteminin x eksenini boyunca $\gamma = 45^\circ$, Y eksenini boyunca $\beta = -15^\circ$ ve Z eksenini boyunca $\alpha = 30^\circ$ döndürülerek elde edilecek dönme matrisini bulalım



Şekil 3.31 Uç işlevcisi

$$R_{XYZ}(45, -15, 30) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (1.53)$$

$$= \begin{bmatrix} \cos 30 \cos(-15) & \cos 30 \sin(-15) \sin 45 - \sin 30 \cos 45 & \cos 30 \sin(-15) \cos 45 + \sin 30 \sin 45 \\ \sin 30 \cos(-15) & \sin 30 \sin(-15) \sin 45 + \cos 30 \cos 45 & \sin 30 \sin(-15) \cos 45 - \cos 30 \sin 45 \\ -\sin(-15) & \cos(-15) \sin 45 & \cos(-15) \cos 45 \end{bmatrix} \quad (1.54)$$

$$R_{XYZ}(45, -15, 30) = \begin{bmatrix} 0,8365 & -0,5120 & 0,1950 \\ 0,4829 & 0,5208 & 0,7038 \\ 0,2588 & 0,6830 & 0,6830 \end{bmatrix} \quad (1.55)$$

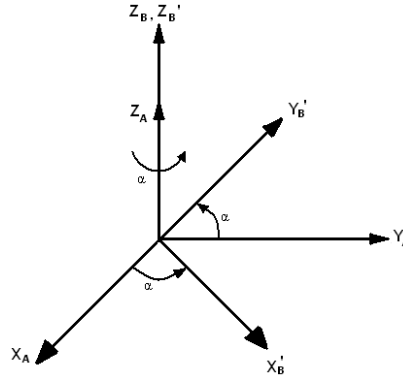
3.1.4.2 ZYX Euler Açısı

Bir koordinat sisteminin yönelimini bir başka koordinat sistemine göre tanımlama için kullanılan yöntemlerden biriside Euler açı seti yöntemidir. Bu açı setinde dönme işlemi sabit koordinat sistemi yerine hareket eden koordinat sistemine göre gerçekleştirilir.

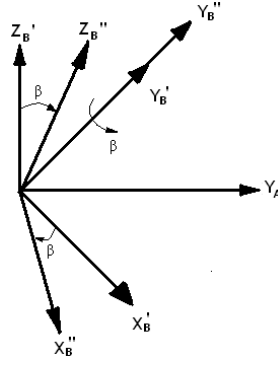
{B} koordinat sisteminin {A} koordinat sistemine göre dönme işlemi ZYX Euler açı sistemine göre aşağıdaki gibi bulunur.

İlk önce merkezleri çakışık olan {B} koordinat sistemi şekilde görüldüğü gibi sırasıyla Z_B eksenı boyunca α , Y_B eksenı boyunca β ve X_B eksenı boyunca γ açısıyla döndürülür. X_A, Y_A ve Z_A eksenleri boyunca dönme işlemi hareketli {B} koordinat sistemine göre ifade edilir.[3] Matematiksel olarak ifadesi ise

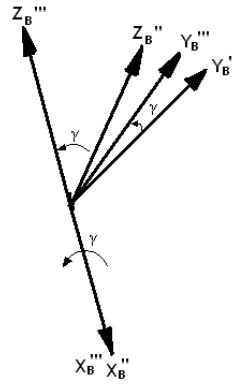
$${}^A_B R_{Z'Y'X'}(\alpha, \beta, \gamma) = R_Z(\alpha)R_Y(\beta)R_X(\gamma) \quad (1.56)$$



Şekil 3.32 {B} koordinat sisteminin Z_B eksenı boyunca α açısıyla döndürülmesi



Şekil 3.33 {B} koordinat sisteminin Y_B eksenini boyunca β açısıyla döndürülmesi



Şekil 3.34 {B} koordinat sisteminin X_B eksenini boyunca γ açısıyla döndürülmesi

$$\begin{aligned}
 {}^A R_{Z'Y'X'}(\alpha, \gamma, \beta) &= \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma & -\cos \alpha \cos \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \\ \sin \alpha \cos \beta \cos \gamma - \cos \alpha \sin \gamma & -\sin \alpha \cos \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \\ -\sin \beta \cos \gamma & \cos \beta \sin \gamma & \cos \beta \end{bmatrix} \quad (1.57)
 \end{aligned}$$

Örnek -1.7 Uç işlevçisi hareketli koordinat sistemine yerleştirilen bir robot Z eksenini boyunca α , Y eksenini boyunca β ve X eksenini boyunca γ açısıyla döndürüldüğünde aşağıdaki dönme matrisi elde ediliyorsa α , β ve γ açılarını bulalım.

$${}^A_B R_{Z'Y'X'}(\alpha, \gamma, \beta) = \begin{bmatrix} -0,30619 & -0,88388 & 0,35355 \\ 0,91856 & -0,17678 & 0,35355 \\ -0,25 & 0,43301 & 0,86603 \end{bmatrix} \quad (1.58)$$

$${}^A_B R_{Z'Y'X'}(\alpha, \gamma, \beta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (1.59)$$

$$= \begin{bmatrix} \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma & -\cos \alpha \cos \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \\ \sin \alpha \cos \beta \cos \gamma - \cos \alpha \sin \gamma & -\sin \alpha \cos \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \\ -\sin \beta \cos \gamma & \cos \beta \sin \gamma & \cos \beta \end{bmatrix}$$

$$\beta = \arctan\left(\frac{\sqrt{r_{31}^2 + r_{32}^2}}{r_{33}}\right) = \arctan\left(\frac{\sqrt{(-0,25)^2 + (0,43301)^2}}{0,86603}\right) = 29,999^\circ \quad (1.60)$$

$$\alpha = \arctan\left(\frac{r_{23}}{\sin \beta} / \frac{r_{33}}{\sin \beta}\right) = \arctan\left(\frac{0,35355}{\sin 30} / \frac{0,35355}{\sin 30}\right) = 45^\circ \quad (1.61)$$

$$\gamma = \arctan\left(\frac{r_{32}}{\sin \beta} / \frac{r_{33}}{\sin \beta}\right) = \arctan\left(\frac{0,43301}{\sin 30} / \frac{-0,25}{\sin 30}\right) = 60^\circ \quad (1.62)$$

3.2 İleri Yön Kinematığı ve Ters Kinematik

3.2.1 İleri Yön (Düz) Kinematığı

Bir robot ana çerçevesinden araç çerçevesine doğru birbirine prizmatik veya döner eklemlerle bağlanmış seri uzuvlardan oluşur. İki uzuv arasındaki ilişki bir homojen dönüşüm matrisiyle açıklanır. Eklem dönüşüm matrislerinin ard arda çarpılmasıyla ana çerçeve ile araç çerçeve arasındaki ilişki tanımlanır. Bu ilişki manipülatörün araç çerçevesinin konumunu ve yönelimini ana çerçeveye göre belirtir. Kısaca ileri yön kinematığı eklem değişkenleri ile uç işlevcisinin konumu ve yönelimini ana çerçeveye göre hesaplar diyebiliriz.

Her bir ekleme bir koordinat sistemi yerleştirilse komşu iki eklem arasındaki ilişki bir ${}^{i-1}T_i$ dönüşüm matrisi elde edilir. İlk ekleme ait dönüşüm matrisi ilk eklem ile ana çerçeve arasındaki ilişkiyi tanımlarken, son ekleme ait dönüşüm matrisi uç işlevcisi ile son eklem arasındaki ilişkiyi ifade eder. Arka arkaya sıralanan bu eklem dönüşüm matrisleriyle ana çerçeve ile araç çerçevesi arasındaki ilişki tanımlanır. Bu ilişkiye de ileri kinematik denir. Ana çerçeve ile araç çerçevesi arasındaki ilişki

$${}^0T_N = {}^0T_1 {}^1T_2 \dots \dots {}^{N-1}T_N \text{ şeklinde tanımlanır.} \quad (2.1)$$

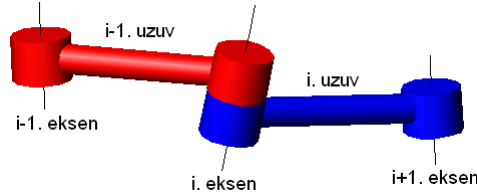
3.2.1.1 Eklem Değişkenlerinin Belirlenmesi

Robotların eklem değişkenlerinin belirlenmesi için birçok kinematik yöntem belirlenmiştir. Kinematik problemler kartezyen üç boyutlu ve kartonom dört boyutlu olmak üzere iki farklı uzayda gerçekleştirilir.[4] Kartezyen uzayda üstel yöntem, Pieper-Roth yöntemi ve Denavit-Hartenberg yöntemi kullanılır. Ancak en fazla tercih edilen yöntem Denavit-Hartenberg yöntemidir.[5]

3.2.1.1.1 Denavit-Hartenberg Yöntemi

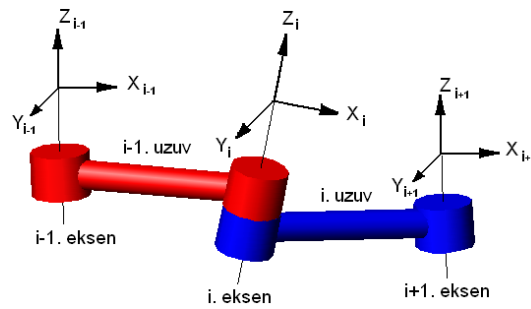
Bu yöntemde dört ana değişken kullanılarak robot kinematiği çıkarılır. Bu değişkenler iki eksen arasındaki uzuv uzunluğu a_{i-1} , iki komşu eksen arasındaki eksen açısı α_{i-1} , üst üste çıkan bağlar arasındaki eklem kayması(kaçıklığı) d_i ve iki komşu uzuv arasındaki eklem açısı θ_{i-1} 'dir. Bu dört değişkende D-H Değişkenidir.

Bu değişkenleri belirlemek için öncelikli olarak şekilden görüldüğü gibi robotun dönme eksenleri belirlenir ve dönme eksenleri uzuvlardan bir fazla olacak şekilde numaralandırılır.



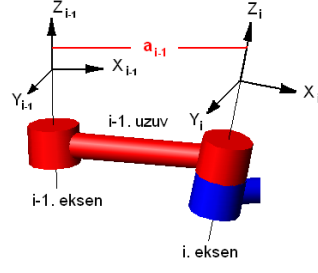
Şekil 3.35 Dönme Eksenlerinin Belirlenmesi

Daha sonra bu eksenlerin her birine bir koordinat sistemi yerleştirilir ve uzuv dönme eksenini aşağıdaki şekilde görüldüğü gibi koordinat sisteminin Z eksenini olarak kabul edilir.



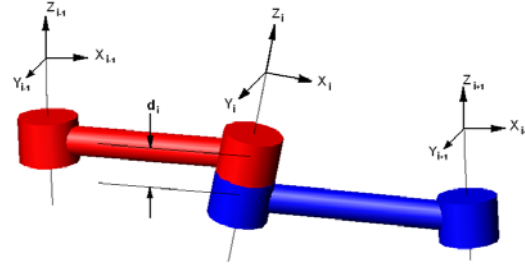
Şekil 3.36 Dönme Eksenlerine Z, Y ve X Eksenlerinin Yerleştirilmesi

Aşağıdaki şekildeki gibi X_{i-1} yönünde uzanan Z_{i-1} ile Z_i eksenleri arasındaki dik uzaklığa a_{i-1} uzuv uzunluğu denir.



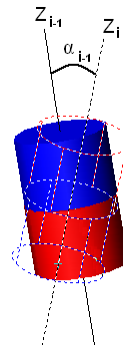
Şekil 3.37 a_{i-1} uzuv uzunluğu

Aşağıdaki şekildeki gibi Z_i yönünde uzanan X_{i-1} ile X_i eksenleri arasındaki dik uzaklığa d_i eklem kaçıklığı denir.



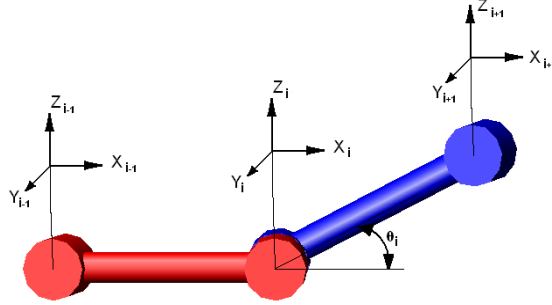
Şekil 3.38 d_i eklem kaçıklığı

Aşağıdaki şekildeki gibi Z_{i-1} eksenini ile Z_i eksenini arasındaki X_i boyunca ölçülen açığa α_{i-1} eksen açısı denir.



Şekil 3.39 α_{i-1} eksen açısı

Aşağıdaki şekildeki gibi X_{i-1} eksenini ile X_i eksenini arasındaki Z_i boyunca ölçülen açıya θ_i eklem açısı denir.



Şekil 3.40 θ_i eklem açısı

3.2.1.2 Koordinat Sistemlerinin Eklemlere Yerleştirilmesi

Koordinat sistemleri eklemlere yerleştirilirken sırasıyla aşağıdaki işlemler gerçekleştirilir.

Öncelikle eklem eksenlerinin dönme ve kayma yönleri belirlenerek bu eksene paralel bir doğru çizilir.

Eklem eksenleri döner eklemler için dönme yönü Z, prizmatik ekler için kayma yönü Z eksenini olarak belirlenir.

Z eksenine dik ve uzuv boyunca uzanan eksen X eksenini olarak belirlenir.

Z ve X eksenleri belirlendikten sonra Y eksenini sağ el kuralına göre belirlenir.

Arka arkaya gelen iki eklem için dönme ve kayma yönleri aynı ise Z eksenini belirlendikten sonra kol boyunca X eksenini belirlenir ve sağ el kuralına göre Y eksenini belirlenir.

Sıfır ve birinci eksenler üst üste kabul edilir.

Bir robotun eklemlerine koordinat sistemleri yerleştirilirken birinci eksenin dönme yönü Z eksenini olarak belirlendikten sonra bu ekleme X eksenini döndürüldüğünde komşu iki Z eksenini üst üste çıkışacak şekilde bir X eksenini yerleştirilir.

Koordinat sistemleri eklemlere yerleştirildikten sonra D-H değişkenleri bulunur ve aşağıdaki tabloya yazılır. Robotun hareket etmesiyle değişmeyen parametreler a_{i-1} uzuv uzunlukları ve α_{i-1} eksen açılarıdır. Değişen parametreler ise eklem döner ise θ_{i-1} eklem açısı eğer eklem prizmatik ise d_{i-1} eklem kaçıklığıdır.

Eksen No	D-H Değişkenleri				Eklem Değişkeni
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i veya θ_i
1	α_0	a_0	d_1	θ_1	d_1 veya θ_1
2	α_1	a_1	d_2	θ_2	d_2 veya θ_2
3	α_2	a_2	d_3	θ_3	d_3 veya θ_3

Tablo 3.3 D-H Değişkenleri

$$\begin{aligned}
 {}^{i-1}T &= R_x(\alpha_{i-1})D_x(a_{i-1})R_z(\theta_i)D_z(d_i) \\
 &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1}d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)
 \end{aligned}$$

Her bir ekleme ait genel dönüşüm matrisi elde edilir.

3.2.1.3 İleri Kinematik Problemlerinin Çözümünde Kullanılan Yaklaşımlar

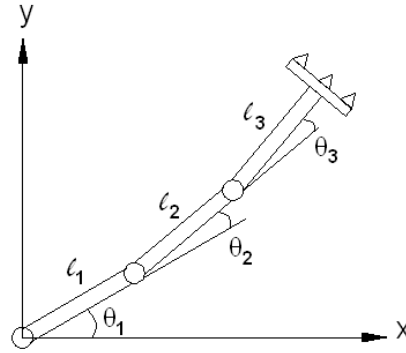
3.2.1.3.1 Geometrik Yaklaşım

Bu yaklaşım manipülatör duruşuna bağlı olarak oluşan geometrik şekilden yararlanır.

3.2.1.3.2 Cebirsel Yaklaşım

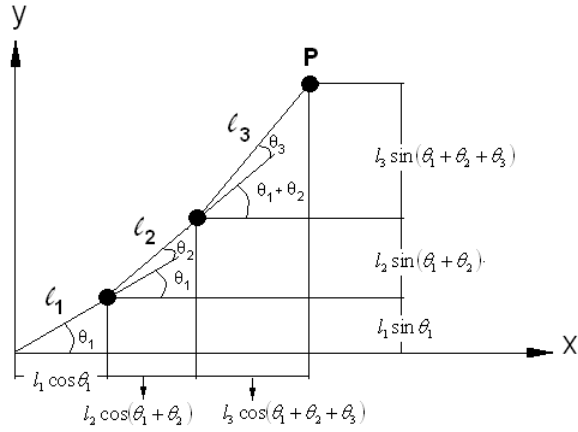
Bu yaklaşım manipülatörün parametreleri ve eklem değişkenleri arasındaki cebirsel ilişkilerden yararlanır

İki boyutlu düzlemde hareket eden robot kolunun geometrik yaklaşım kullanılarak ileri yön kinematiğinin bulunması



Şekil 3.41 İki boyutlu düzlemde hareket eden robot kolu

Öncelikle robotun iki boyutlu düzlemde aldığı şekil çizilerek uç işlevcisini ifade eden P noktasının konumu bulunur.



Şekil 3.42 İki boyutlu düzlemde hareket eden P noktasının konumu

$$P_x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (2.3)$$

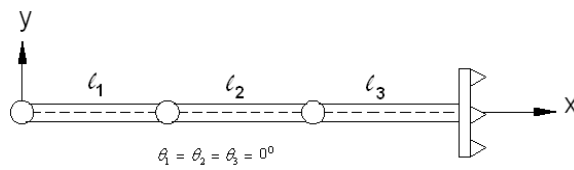
P noktasının X Eksenindeki izdüşümü

$$P_y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (2.4)$$

P noktasının Y Eksenindeki izdüşümü

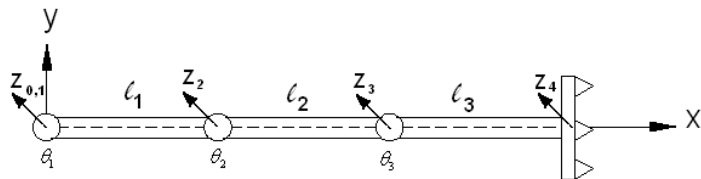
Aynı kolun ileri yön kinematikliğini D-H yöntemi ile bulalım.

Öncelikle kolun başlangıç değerlerine göre eklemlere koordinat sistemleri yerleştirilir.



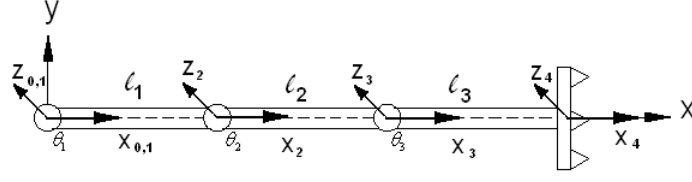
Şekil 3.43 İki boyutlu düzlemde hareket eden kolun başlangıç değerleri

İkinci adımda dönme eklemlerine Z eksenleri yerleştirilir.



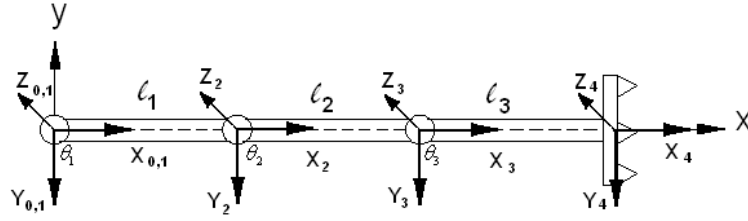
Şekil 3.44 Dönme Eksenlerine Z Eksenlerinin Yerleştirilmesi

Üçüncü adımda Z eksenine dik ve uzuv boyunca uzanan X eksenleri yerleştirilir.



Şekil 3.45 X Eksenlerinin Yerleştirilmesi

Dördüncü adımda Sağ El kuralına göre Y eksenleri yerleştirilir.



Şekil 3.46 Sağ El kuralına göre Y Eksenlerinin Yerleştirilmesi

Son adımda da Koordinat sistemleri eklemlere yerleştirildikten sonra D-H değişkenleri belirlenir ve tabloya eklenir. (Z Eksenini bizden kâğıda doğrudur.)

Öncelikle D-H değişkenlerinden sabit olan parametreler belirlenir.

$Z_{0,1}, Z_2, Z_3$ ve Z_4 , eksenlerinin dönme yönleri aynı olduğundan

$\alpha_0, \alpha_1, \alpha_2$ ve α_3 açıları 0 'dır. Aynı şekilde 0. ve 1. eklemler üst üste olduğundan

X_0 yönünde uzanan Z_0 ile Z_1 eksenleri arasında $a_0 = 0$

X_1 yönünde uzanan Z_1 ile Z_2 eksenleri arasında $a_1 = l_1$

X_2 yönünde uzanan Z_2 ile Z_3 eksenleri arasında $a_2 = l_2$

X_3 yönünde uzanan Z_3 ile Z_4 eksenleri arasında $a_3 = l_3$

Şimdi ise D-H değişkenlerinden değişen parametrelerini bulalım ancak her bir eklem için yalnızca bir parametre değişken olabilir kısacası döner eklem için eklem açısı, kayar eklem içinse eklem kaçıklığıdır.

Örneğimizdeki tüm eklemler döner tip olduğundan d_1, d_2, d_3 ve d_4 eklem kaçıklıkları 0, $\theta_1, \theta_2, \theta_3$ ve θ_4 açıları değişkendir.

Eksen No	D-H Değişkenleri				Eklem Değişkeni
	α_{i-1}	a_{i-1}	d_i	θ_i	
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i veya θ_i
1	0	0	0	θ_1	θ_1
2	0	l_1	0	θ_2	θ_2
3	0	l_2	0	θ_3	θ_3
4	0	l_3	0	0	0

Tablo 3.4 D-H Değişkenleri

Elde edilen tabloya göre her ekleme ait değişkenleri aşağıdaki genel matriste yerine koyarak dönüşüm matrislerini bulalım.

$${}^{i-1}_i T = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Birinci eklem için dönüşüm matrisi

$${}^0_1 T = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_0 \\ \sin \theta_1 \cos \alpha_0 & \cos \theta_1 \cos \alpha_0 & -\sin \alpha_0 & -\sin \alpha_0 d_1 \\ \sin \theta_1 \sin \alpha_0 & \cos \theta_1 \sin \alpha_0 & \cos \alpha_0 & \cos \alpha_0 d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$$= \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

İkinci eklem için dönüşüm matrisi

$$\begin{aligned}
 {}^1_2T &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_1 \\ \sin \theta_2 \cos \alpha_1 & \cos \theta_2 \cos \alpha_1 & -\sin \alpha_1 & -\sin \alpha_1 d_2 \\ \sin \theta_2 \sin \alpha_1 & \cos \theta_2 \sin \alpha_1 & \cos \alpha_1 & \cos \alpha_1 d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{2.7}$$

Üçüncü eklem için dönüşüm matrisi

$$\begin{aligned}
 {}^2_3T &= \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_2 \\ \sin \theta_3 \cos \alpha_2 & \cos \theta_3 \cos \alpha_2 & -\sin \alpha_2 & -\sin \alpha_2 d_3 \\ \sin \theta_3 \sin \alpha_2 & \cos \theta_3 \sin \alpha_2 & \cos \alpha_2 & \cos \alpha_2 d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_2 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{2.8}$$

Dördüncü eklem için dönüşüm matrisi

$$\begin{aligned}
 {}^3_4T &= \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & a_3 \\ \sin \theta_4 \cos \alpha_2 & \cos \theta_4 \cos \alpha_3 & -\sin \alpha_3 & -\sin \alpha_3 d_4 \\ \sin \theta_4 \sin \alpha_2 & \cos \theta_4 \sin \alpha_3 & \cos \alpha_3 & \cos \alpha_3 d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & l_3 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{2.9}$$

Elde edilen dört ekleme ait dönüşüm matrisleri birbirleriyle çarpılarak manipülatöre ait dönüşüm matrisi elde edilir.

$$\begin{aligned}
{}^0_4T &= {}^0_1T {}^1_2T {}^2_3T {}^3_4T \\
&= \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_2 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)
\end{aligned}$$

$$= \begin{bmatrix} c(\theta_1 + \theta_2)c\theta_3 - s(\theta_1 + \theta_2)s\theta_3 & -s(\theta_1 + \theta_2)c\theta_3 + c(\theta_1 + \theta_2)s\theta_3 & 0 & l_3(c(\theta_1 + \theta_2)c\theta_3 - s(\theta_1 + \theta_2)s\theta_3) + l_2c(\theta_1 + \theta_2) + l_1c\theta_1 \\ s(\theta_1 + \theta_2)c\theta_3 + c(\theta_1 + \theta_2)s\theta_3 & c(\theta_1 + \theta_2)c\theta_3 - s(\theta_1 + \theta_2)s\theta_3 & 0 & l_3(s(\theta_1 + \theta_2)c\theta_3 + c(\theta_1 + \theta_2)s\theta_3) + l_2s(\theta_1 + \theta_2) + l_1s\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

elde edilen dönüşüm matrisinin konum vektöründen

$$P_x = l_3 \cos(\theta_1 + \theta_2 + \theta_3) + l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \quad (2.12)$$

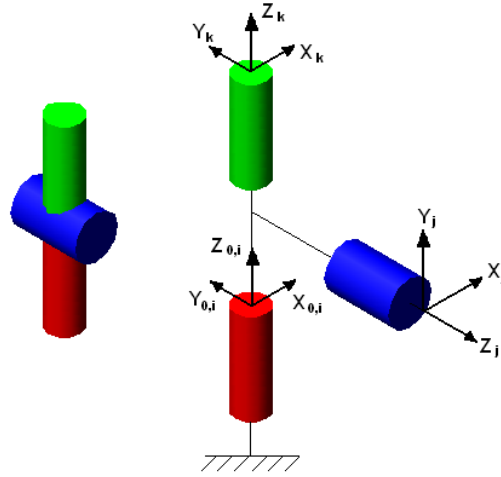
$$P_y = l_3 \sin(\theta_1 + \theta_2 + \theta_3) + l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \quad (2.13)$$

Denklemleri elde edilir.

3.2.1.4 Robotlarda Kullanılan Bilek Düzenleşimleri

Endüstriyel robotlarda Euler ve eklem kaçıklıklı bilek olmak üzere iki tip bilek düzenleşimi kullanılır. Euler bilekli düzenleşimde üç eksen bir noktada kesişirken eklem kaçıklıklı bilekte eklemlerin kesişmesi eklem kayması ve uzuv uzunlukları ile engellenir.[3]

3.2.1.4.1 Euler Bileklik Düzenleşimin İleri Yön Kinematığı



Şekil 3.47 Euler Bileklik Düzenleşimi

Öncelikle D-H Değişkenlerine ait tablo oluşturulur

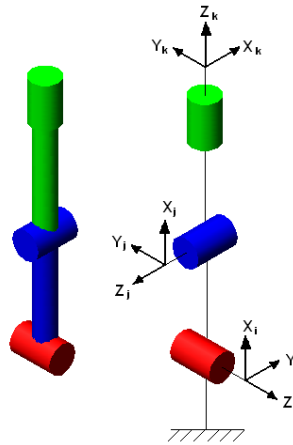
Eksen No	D-H Değişkenleri				Eklem Değişkeni
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i veya θ_i
1	0	0	0	θ_i	θ_i
2	90	0	0	θ_j	θ_j
3	-90	0	0	θ_k	θ_k

Tablo 3.5 D-H Değişkenleri

Bu tabloya göre de bilek düzenleşimine ait ileri yön kinematiğine ait dönüşüm matrisi bulunur.

$$\begin{aligned}
 {}^0_k T &= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_j & -s\theta_j & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_j & c\theta_j & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_k & -s\theta_k & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_k & -c\theta_k & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c\theta_i c\theta_j c\theta_k - s\theta_i s\theta_k & -c\theta_i c\theta_j s\theta_k - s\theta_i c\theta_k & -c\theta_i s\theta_j & 0 \\ s\theta_i c\theta_j c\theta_k + c\theta_i s\theta_k & -s\theta_i c\theta_j s\theta_k + c\theta_i c\theta_k & -s\theta_i s\theta_j & 0 \\ s\theta_j c\theta_k & -s\theta_j s\theta_k & s\theta_j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)
 \end{aligned}$$

3.2.1.4.2 Eklem Kaçıklıklı Bileklik Düzenleşiminin İleri Yön Kinematiği



Şekil 3.48 Eklem Kaçıklıklı Bileklik Düzenleşimi

Öncelikle D-H Değişkenlerine ait tablo oluşturulur

Eksen No	D-H Değişkenleri				Eklem Değişkeni
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i veya θ_i
1	0	0	0	θ_i	θ_i
2	90	a	0	θ_j	θ_j
3	-90	0	d	θ_k	θ_k

Tablo 3.6 D-H Değişkenleri

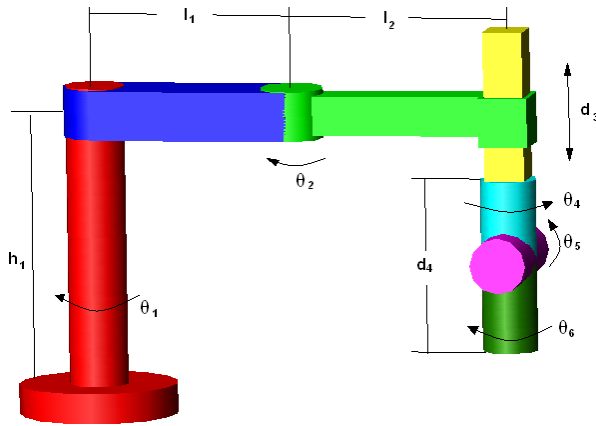
Bu tabloya göre de bilek düzenleşimine ait ileri yön kinematiğine ait dönüşüm matrisi bulunur.

$$\begin{aligned}
 {}^0_k T &= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_j & -s\theta_j & 0 & a \\ 0 & 0 & -1 & 0 \\ s\theta_j & c\theta_j & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_k & -s\theta_k & 0 & 0 \\ 0 & 0 & 1 & d \\ -s\theta_k & -c\theta_k & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c\theta_i c\theta_j c\theta_k - s\theta_i s\theta_k & -c\theta_i c\theta_j s\theta_k - s\theta_i c\theta_k & -c\theta_i s\theta_j & d(-c\theta_i s\theta_j) + a(c\theta_i) \\ s\theta_i c\theta_j c\theta_k + c\theta_i s\theta_k & -s\theta_i c\theta_j s\theta_k + c\theta_i c\theta_k & -s\theta_i s\theta_j & d(-s\theta_i s\theta_j) - a(s\theta_i) \\ s\theta_j c\theta_k & -s\theta_j s\theta_k & s\theta_j & d(c\theta_j) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)
 \end{aligned}$$

3.2.1.5 Altı Serbestlik Derecesine Sahip Robotun İleri Yön Kinematiği

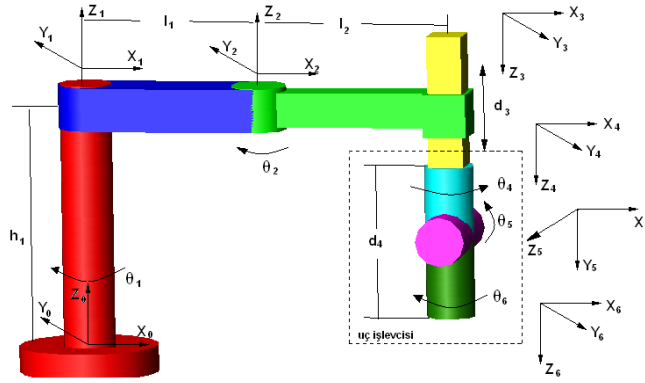
Endüstriyel robotlara Euler veya eklem kaçıklıklı bilek eklendiğinde 6 serbestlik dereceli robotlar elde edilir. Euler bileklikli robotlar endüstride hafif yüklerde tercih edilirken, eklem kaçıklıklı robotlar ağır yüklerin kaldırılması gibi işlemlerde tercih edilir.[6]

Örnek –2.1 Şekil 3.49 ‘daki Scara diye tanımlana 6 serbestlik dereceli robotun ileri yön kinematiğini bulalım



Şekil 3.49 Euler Bilekli 6 Serbestlik Derecesine Sahip Scara Robot

Öncelikle koordinat sistemlerini her bir ekleme yerleştirelim ve ardından D-H parametrelerine ait tabloyu oluşturalım.



Şekil 3.50 Euler Bilekli Scara Robota Eksenlerin Yerleştirilmesi

i	α_{i-1}	a_{i-1}	d_i	θ_i	Değişken
1	0	0	h_1	θ_1	θ_1
2	0	l_1	0	θ_2	θ_2
3	180	l_2	d_3	0	d_3
4	0	0	d_4	θ_4	θ_4
5	-90	0	0	θ_5	θ_5
6	90	0	0	θ_6	θ_6

Tablo 3.7 D-H Değişkenleri

Her ekleme ait dönüşüm matrisleri bulunur

$${}^0_1T = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

$${}^1_2T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

$${}^3_4T = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & 0 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

$${}^4_5T = \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin \theta_5 & -\cos \theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.20)$$

$${}^5_6T = \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.21)$$

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T \quad (2.22)$$

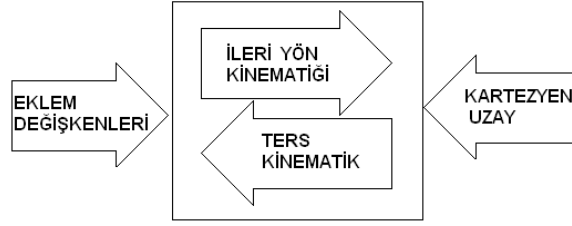
$${}^0_6T = \begin{bmatrix} (c(\theta_1 + \theta_2)c\theta_4 + s(\theta_1 + \theta_2)s\theta_4)c\theta_5c\theta_6 - (c(\theta_1 + \theta_2)c\theta_4 - s(\theta_1 + \theta_2)s\theta_4)s\theta_6 \\ (s(\theta_1 + \theta_2)c\theta_4 - c(\theta_1 + \theta_2)s\theta_4)c\theta_5c\theta_6 - (s(\theta_1 + \theta_2)c\theta_4 + c(\theta_1 + \theta_2)s\theta_4)s\theta_6 & \cdots \\ & s\theta_5c\theta_6 \\ & 0 \end{bmatrix} \quad (2.23)$$

$$\begin{aligned} & - (c(\theta_1 + \theta_2)c\theta_4 + s(\theta_1 + \theta_2)s\theta_4)c\theta_5s\theta_6 - (c(\theta_1 + \theta_2)c\theta_4 - s(\theta_1 + \theta_2)s\theta_4)c\theta_6 \\ \cdots & - (s(\theta_1 + \theta_2)c\theta_4 - c(\theta_1 + \theta_2)s\theta_4)c\theta_5s\theta_6 - (s(\theta_1 + \theta_2)c\theta_4 + c(\theta_1 + \theta_2)s\theta_4)c\theta_6 & \cdots \\ & - s\theta_5c\theta_6 \\ & 0 \end{aligned} \quad (2.24)$$

$$\begin{aligned} & - (c(\theta_1 + \theta_2)c\theta_4 + s(\theta_1 + \theta_2)s\theta_4)s\theta_5l_2c(\theta_1 + \theta_2) + l_1c\theta_1 \\ \cdots & - (s(\theta_1 + \theta_2)c\theta_4 - c(\theta_1 + \theta_2)s\theta_4)s\theta_5l_2s(\theta_1 + \theta_2) + l_1s\theta_1 \\ & \quad - c\theta_6 \quad \quad \quad h_1 - d_3 - d_4 \\ & \quad \quad \quad 0 \quad \quad \quad 1 \end{aligned} \quad (2.25)$$

3.2.2 Ters Kinematik

Robotun uç işlevcisinin ana çerçeveye göre konumu ve yönelimi verildiğinde manipülâtörün bu konuma ve yönelime gelebilmesi için gerekli eklem deęişkenlerinin bulunmasıdır. Kısacası uç işlevcisinin konum ve yönelim verileriyle eklem deęişkenlerinin bulunması olarak da tanımlanabilir. Başka bir deęişle de uç işlevcisinin konum ve yönelimini kartezyen koordinat sisteminden eklem koordinat sistemine dönüştürme işlemi olarak da tanımlayabiliriz.

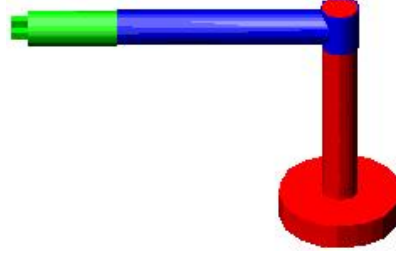


Şekil 3.51 Eklem Deęişkenleri ile Kartezyen Uzay Dönüştürümü

Ters kinematik aşağıdaki nedenlerden dolayı çözülmesi oldukça zor olan problemler içerir. [3]

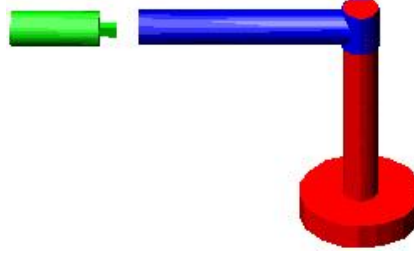
- Analitik olarak karmaşık, doğrusal olmayan denklemler içerir.
- Eklemlerin yapısına bağlıdır. Eğer robot prizmatik eklemlerden oluşuyorsa ters kinematik problemin çözümü kolaylaşırken, robottaki döner eklem sayısı arttıkça problemin çözümü de o derece zorlaşmaktadır.
- Her zaman matematiksel çözüm fiziksel çözümü temsil etmez.

Şekil 3.52 ‘de matematiksel çözümle fiziksel çözüm örtüşürken ikinci Şekil 3.53 ‘de örtüşmez.



$$\theta = \arctan 2(-k, p_z)$$

Şekil 3.52 Fiziksel Çözüm

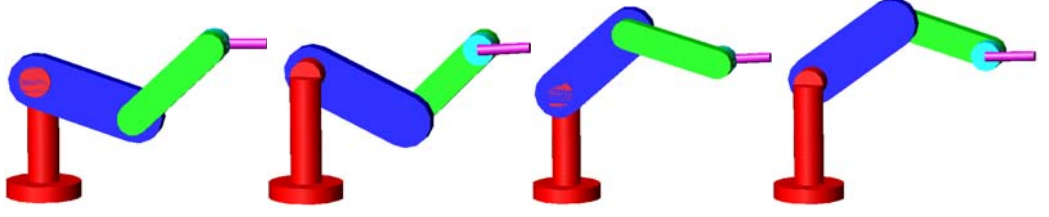


$$\theta = \arctan 2(k, -p_z)$$

Şekil 3.53 Matematiksel Çözüm

- Aynı uç işlevci düzenleşimi için birden fazla çözüm olabilir. Ters kinematik çözüm sayısı robotun serbestlik derecesinin yanında aynı zamanda eklem değişkenlerine de bağlıdır. Her bir ekleme uzuv uzunluğu ve eklem kaçıklığının olması çözüm sayısının artmasına neden olur. Örneğin 6R robotta her bir eklem için en azından bir uzuv uzunluğu ve eklem kaçıklığı olduğundan ters kinematik çözüm sayısı $2^6 = 64$ 'dür. Yalnız bu çözümlerin bir kısmı gerçek bir kısmı ise sanaldır.

Dönel eklemlerden oluşan robotlarda fiziksel çözüm sayısının fazla olması, üç boyutlu uzayda bir noktaya birkaç şekilde ulaşma imkânı sağlar.[7] Örneğim PUMA robotunun aynı noktaya dört farklı şekilde erişebildiğini gösterir.



Şekil 3.54 Puma Robotun Aynı Noktaya Dört Farklı Şekilde Ulaşması

- Ters kinematik problem verilen bir robot düzenleşimi için tamamen analitik olarak çözülebileceği gibi, analitik çözümün mümkün olmadığı durumlarda da sayısal yöntemler kullanılarak da çözülebilir. Tamamıyla kesin sonuç üreten analitik çözüme ait denklemler bilgisayara ortamında çok hızlı çalışırken, eklem açılarının iteratif olarak çözüldüğü sayısal çözüm ise bilgisayar ortamında analitik yöntemle göre yavaş çalışır. Bu nedenle robot tasarımcıları tasarımı basit analitik çözümü mümkün olana robot tasarımı üzerinde durmuşlardır.

Sayısal çözüm için kinematik eşitliklerin farklarının toplamını alan tahmin edeci ve düzeltici tip algoritmalar kullanılır. Ters kinematik problemin sayısal yöntemlerle çözümlenmesinde karşılaşılan en büyük sorun, Jakobyen matrisinin tekil olduğu noktalarda yazılan algoritmanın çözüm üretememesidir. Ayrıca başlangıç çözüm vektörü çözüme yönelik elemanlardan oluşmadığı zaman sayısal çözüm kararlı bir çözüm üretemez. Tamamıyla sayısal veya analitik çözümün gerçekleşmediği durumlarda eklem değişkenlerinden bir kaçını analitik olarak bulunup diğerleri sayısal olarak çözülür.[8]

3.2.2.1 Ters Kinematik Problemlerine Analitik Çözüm Yaklaşımı

Craig[9] tarafından tanımlanan altı serbestlik derecesine sahip bir robotun ileri yön kinematiği aşağıdaki gibi yazılır.

$${}^0T = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.26)$$

0T İleri yön kinematiği matrisi, konum ve yönelim verilerini içeren matris elemanları cinsinden Denklem 2.27 'deki gibi yazılıp Denklem 2.28 teki eşitlik sağlanır.

Denklem 2.27 'nin 1. kolonu uç işlevcisinin normal vektörünü, 2. kolonu kayma ve 3. kolonu ise yaklaşım vektörünü göstermektedir.

$${}^0T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.27)$$

$$[{}^0T]^{-1} {}^0T = [{}^1T]^{-1} {}^1T = [{}^1T]^{-1} {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.28)$$

$[{}^0T]^{-1} {}^0T = I$ olduğundan yukarıdaki denklem daha aşağıdaki gibi basit bir ifadeyle elde edilebilir.

$$[{}^0T]^{-1} {}^0T = {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.29)$$

$$[{}^0T_2]^{-1} {}^0T = {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.30)$$

$$[{}^0T_2 {}^1T_3]^{-1} {}^0T = {}^3T_4 {}^4T_5 {}^5T_6 \quad (2.31)$$

$$[{}^0T_2 {}^1T_3 {}^2T_4]^{-1} {}^0T = {}^4T_5 {}^5T_6 \quad (2.32)$$

$$[{}^0T_2 {}^1T_3 {}^2T_4 {}^3T_5]^{-1} {}^0T = {}^5T_6 \quad (2.33)$$

Ters kinematik çözüm gerçekleştirilirken kullanılan bazı trigonometrik eşitlikler.

$$\cos \theta = a \quad \text{ise} \quad \theta = \arctan 2\left(\pm \sqrt{1-a^2}, a\right) \quad (2.34)$$

$$\sin \theta = a \quad \text{ise} \quad \theta = \arctan 2\left(a, \pm \sqrt{1-a^2}\right) \quad (2.35)$$

$$\cos \theta = a \quad \text{ve} \quad \sin \theta = b \quad \text{ise} \quad \theta = \arctan 2(b, a) \quad (2.36)$$

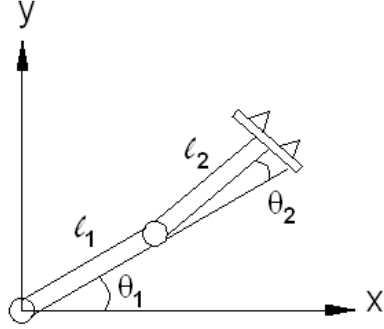
$$a \sin \theta + b \cos \theta = 0 \quad \text{ise} \quad (2.37)$$

$$\theta = \arctan 2(-b, a) \quad \text{veya} \quad \theta = \arctan 2(b, -a) \quad (2.38)$$

$$a \sin \theta + b \cos \theta = c \quad \text{ise} \quad (2.39)$$

$$\theta = \arctan 2(a, b) + \theta = \arctan 2\left(\pm \sqrt{a^2 + b^2 - c^2}, c\right) \quad (2.40)$$

Örnek -2.2



Şekil 3.55 İki boyutlu düzlemde hareket eden robot kolu

$${}^0_1T = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.41)$$

$${}^1_2T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.42)$$

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.43)$$

$${}^0_3T = {}^0_1T {}^1_2T {}^2_3T \quad (2.44)$$

İleri yön kinematiğine ait dönüşüm matrisinin her iki tarafını ${}^0_1T^{-1}$ ile çarpalım.

$${}^0_1T^{-1} {}^0_3T = {}^0_1T^{-1} {}^1_2T {}^2_3T \quad (2.45)$$

Bilindiği gibi ${}^0_1T^{-1} {}^0_1T = I$ olduğundan denklem aşağıdaki gibi olur.

$${}^0_1T^{-1} {}^0_3T = {}^1_2T {}^2_3T \quad (2.46)$$

$${}^0_1T^{-1} = \begin{bmatrix} {}^0_1R^T & -{}^0_1R^T P_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.47)$$

$${}^0_1R = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^0_1R^T = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.48)$$

0_1T Dönüşüm matrisinin konum vektörü sıfır olduğundan

$$-{}^0_1R^T P_1 = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.49)$$

$${}^0_1T^{-1} = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.50)$$

elde edilir. Bu matrisi 0_3T ileri kinematiği temsil eden matrisle çarpalım

$${}^0_3T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.51)$$

$${}^0_1T^{-1} {}^0_3T = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos \theta_1 r_{11} + \sin \theta_1 r_{21} & \cos \theta_1 r_{12} + \sin \theta_1 r_{22} & \cos \theta_1 r_{13} + \sin \theta_1 r_{23} & \cos \theta_1 p_x + \sin \theta_1 p_y \\ -\sin \theta_1 r_{11} + \cos \theta_1 r_{21} & -\sin \theta_1 r_{12} + \cos \theta_1 r_{22} & -\sin \theta_1 r_{13} + \cos \theta_1 r_{23} & -\sin \theta_1 p_x + \cos \theta_1 p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.52)$$

${}^0_1T^{-1} {}^0_3T$ Çarpımını bulduk. Şimdide ${}^1_2T {}^2_3T$ çarpımını bulalım ve iki sonucu birbirine eşitleyelim.

$${}^1_2T {}^2_3T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 + l_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \cos \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.53)$$

Bulunur.

$$\begin{bmatrix} \cos \theta_1 r_{11} + \sin \theta_1 r_{21} & \cos \theta_1 r_{12} + \sin \theta_1 r_{22} & \cos \theta_1 r_{13} + \sin \theta_1 r_{23} & \cos \theta_1 p_x + \sin \theta_1 p_y \\ -\sin \theta_1 r_{11} + \cos \theta_1 r_{21} & -\sin \theta_1 r_{12} + \cos \theta_1 r_{22} & -\sin \theta_1 r_{13} + \cos \theta_1 r_{23} & -\sin \theta_1 p_x + \cos \theta_1 p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 + l_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \cos \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.54)$$

$$1. \quad \cos \theta_1 r_{11} + \sin \theta_1 r_{21} = \cos \theta_2 \quad (2.55)$$

$$2. \quad -\sin \theta_1 r_{11} + \cos \theta_1 r_{21} = \sin \theta_2 \quad (2.56)$$

$$3. \quad r_{31} = 0 \quad (2.57)$$

$$4. \quad \cos \theta_1 r_{12} + \sin \theta_1 r_{22} = -\sin \theta_2 \quad (2.58)$$

$$5. -\sin \theta_1 r_{12} + \cos \theta_1 r_{22} = \cos \theta_2 \quad (2.59)$$

$$6. r_{32} = 0 \quad (2.60)$$

$$7. \cos \theta_1 r_{13} + \sin \theta_1 r_{23} = 0 \quad (2.61)$$

$$8. -\sin \theta_1 r_{13} + \cos \theta_1 r_{23} = 0 \quad (2.62)$$

$$9. r_{33} = 0 \quad (2.63)$$

$$10. \cos \theta_1 P_x + \sin \theta_1 P_y = l_2 \cos \theta_2 + l_1 \quad (2.64)$$

$$11. -\sin \theta_1 P_x + \cos \theta_1 P_y = l_2 \sin \theta_2 \quad (2.65)$$

$$12. P_z = 0 \quad (2.66)$$

Eşitlikleri elde edilir.

Ters kinematik çözümün tamamı mümkünse kol uzunlukları cinsinden elde edilmelidir. Bu nedenle 10. ve 11. denklemlerin her iki tarafının karesini alıp alt alta toplayalım.

$$\begin{aligned} & \cos^2 \theta_1 P_x^2 + \sin^2 \theta_1 P_y^2 = l_2^2 \cos^2 \theta_2 + 2l_1^2 l_2^2 \cos^2 \theta_2 + l_1^2 \\ & + \frac{(-\sin \theta_1)^2 P_x^2 + \cos^2 \theta_1 P_y^2 = l_2^2 \sin^2 \theta_2}{\cos^2 \theta_1 P_x^2 + \sin^2 \theta_1 P_y^2 + \sin^2 \theta_1 P_x^2 + \cos^2 \theta_1 P_y^2 = l_2^2 \cos^2 \theta_2 + 2l_1 l_2 \cos \theta_2 + l_1^2 + l_2^2 \sin^2 \theta_2} \end{aligned} \quad (2.67)$$

Eşitliğin sol tarafını P_x^2 ve P_y^2 , sağ tarafını da l_2^2 parantezine alalım.

$$(\cos^2 \theta_1 + \sin \theta_1^2) P_x^2 + (\sin^2 \theta_1 + \cos^2 \theta_1) P_y^2 = l_2^2 (\cos^2 \theta_2 + \sin^2 \theta_2) + 2l_1 l_2 \cos \theta_2 + l_1^2 \text{ olur.} \quad (2.68)$$

$$(\cos^2 \theta_1 + \sin \theta_1^2) = 1 \text{ 'dir.} \quad (2.69)$$

Bu durumda denklemi yeniden yazarsak

$$P_x^2 + P_y^2 = l_2^2 + 2l_1 l_2 \cos \theta_2 + l_1^2 \text{ olur.} \quad (2.70)$$

Buradan

$$\cos \theta_2 = \left(\frac{P_x^2 + P_y^2 - l_2^2 + l_1^2}{2l_1l_2} \right) \text{ bulunur.} \quad (2.71)$$

Bunu $\cos \theta = a$ ise $\theta = \arctan 2\left(\pm \sqrt{1-a^2}, a\right)$ denkleminde uyarlırsak (2.72)

$$\theta_2 = \arctan 2 \left(\pm \sqrt{1 - \left(\frac{P_x^2 + P_y^2 - l_2^2 + l_1^2}{2l_1l_2} \right)^2}, \left(\frac{P_x^2 + P_y^2 - l_2^2 + l_1^2}{2l_1l_2} \right) \right) \text{ bulunur.} \quad (2.73)$$

$$\cos \theta_1 P_x + \sin \theta_1 P_y = l_2 \cos \theta_2 + l_1 \quad (2.74)$$

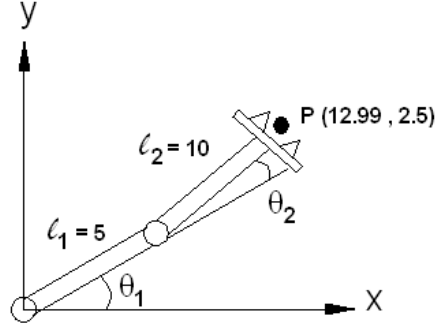
Bu ifadeyi de $a \sin \theta + b \cos \theta = c$ 'e uyarlırsak. (2.75)

$$P_x = a, P_y = b, l_2 \cos \theta_2 + l_1 = c \text{ Olur.} \quad (2.76)$$

$$\theta = \arctan 2(a, b) \pm \theta = \arctan 2\left(\sqrt{a^2 + b^2 - c^2}, c\right)$$

$$\theta_1 = \arctan 2(P_y, P_x) \pm \theta = \arctan 2\left(\sqrt{P_y^2 + P_x^2 - (l_2 \cos \theta_2 + l_1)^2}, (l_2 \cos \theta_2 + l_1)\right) \text{ elde edilir.}$$

Örnek –2.3 RR bir robotun kol uzunlukları $l_1 = 10$ ve $l_2 = 5$ veriliyor. Robotun uç işlevcisinin kartezyen uzaydaki konumu $P_x = 12,99$, $P_y = 2,5$ olması için birinci eklem θ_1 ve ikinci eklem θ_2 değişkeni kaç derece olmalıdır?



Şekil 3.56 İki boyutlu düzlemde hareket eden P noktasının konumu

$$\begin{aligned}
 \theta_2 &= \arctan 2 \left(\pm \sqrt{1 - \left(\frac{P_x^2 + P_y^2 - l_2^2 + l_1^2}{2l_1 l_2} \right)^2}, \left(\frac{P_x^2 + P_y^2 - l_2^2 + l_1^2}{2l_1 l_2} \right) \right) \\
 &= \arctan 2 \left(\pm \sqrt{1 - \left(\frac{12,99^2 + 2,5^2 - 5^2 + 10^2}{2 \cdot 5 \cdot 10} \right)^2}, \left(\frac{12,99^2 + 2,5^2 - 5^2 + 10^2}{2 \cdot 5 \cdot 10} \right) \right) \quad (2.77) \\
 &= \arctan 2 \left(\pm \sqrt{1 - (0,4999)^2}, (0,4999) \right) = \pm 60^\circ
 \end{aligned}$$

$\theta_2 = \pm 60^\circ$ Bulunur. Kosinüs çift özelliğe sahip olduğu için negatif ve pozitif değeri eşittir.

$$\begin{aligned}
 \theta_1 &= \arctan 2(P_y, P_x) \pm \theta = \arctan 2 \left(\sqrt{P_y^2 + P_x - (l_2 \cos \theta_2 + l_1)^2}, (l_2 \cos \theta_2 + l_1) \right) \\
 &= \arctan 2(2,5, 12,99) \pm \theta = \arctan 2 \left(\sqrt{2,5^2 + 12,99 - (5 \cos 60 + 10)^2}, (5 \cos 60 + 10) \right) \quad (2.78) \\
 &= 10,9 \pm 19,1
 \end{aligned}$$

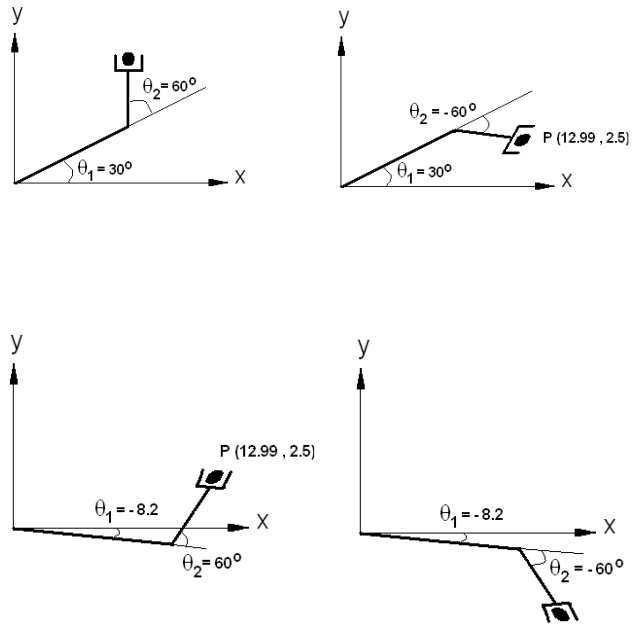
$$C_1 = \{\theta_1 = 10,9 + 19,1 = 30^\circ \quad \text{ve} \quad \theta_2 = +60^\circ\} \quad (2.79)$$

$$C_2 = \{\theta_1 = 10,9 + 19,1 = 30^\circ \quad \text{ve} \quad \theta_2 = -60^\circ\} \quad (2.80)$$

$$C_3 = \{\theta_1 = 10,9 - 19,1 = -8,2^\circ \text{ ve } \theta_2 = +60^\circ\} \quad (2.81)$$

$$C_4 = \{\theta_1 = 10,9 - 19,1 = -8,2^\circ \text{ ve } \theta_2 = -60^\circ\} \quad (2.82)$$

Görüldüğü gibi ters kinematik probleminin 4 tane çözüm kümesi bulundu. 1. ve 4. çözüm kümeleri uç işlevcisini hedefe ulaştıramazlar o halde 4 çözümün sadece iki tanesi gerçek çözümdür.



Şekil 3.57 Ters Kinematik Probleminin Gerçek Ve Sanala Çözümlerine Ait Konumları

3.3 Teze Konu Olan Projenin Tanıtımı

Bu çalışmada, dört serbestlik dereceli dört dönel ekleme ve bir uç işlevcisine sahip bir robot kolunun eklem hareketlerini birebir takip eden veya istenildiğinde bilgisayar ortamında hazırlanmış kontrol yazılımıyla hareket ettirilebilen bir robot kolunun oluşturulması amaçlanmıştır. Kollardan birine eklemlerdeki dönme hareketini analog sinyale çevirmek için potansiyometreler yerleştirilerek adına master kol adı verilmiştir. İkinci yani slave olarak adlandırılan kol eklemlerine ise master kolun eklem açılarını almasını sağlayan RC servo motorlar yerleştirilmiştir. Hazırlanan kontrol yazılımıyla master koldan alınan analog sinyaller JAC3320 ADC Veri toplama kartıyla bilgisayar ortamında işlenerek Slave kolun eklemlerine hareket veren servo motor sürücü kartına Analog sinyal olarak gönderilir. Servo motor sürücü kartına gelen analog sinyaller Pulse Width Modulation (Sinyal Darbe Genişliği) mantığıyla servo motorların konum kontrolünü sağlayan kontrol sinyallerine dönüştürülür.

Böylece hazırlanan yazılımdan yapılan seçime göre Slave kol;

- Master kolun hareketini birebir takip eder
- Yazılımla verilen eklem açılarına göre hareket ettirilir
- Yazılımla verilen uç işlevcisi koordinatlarına göre hareket ettirilebilir.

Kontrol yazılımı LabVIEW grafiksel programlama dili ile hazırlanmıştır. Bu yazılımda Master ve slave kolun eklem açıları, buna bağlı olarak konum vektörleri ve grafiği yazılımda anlık olarak gösterilmektedir. Konum vektörleri eklem değişkenlerinin anlık değişmelerine göre hazırlanan yazılım altında MATLAB scripti ile hesaplanmaktadır.

3.3.1 Projede Kullanılan Sistem Elemanları

Bu projede hareketleri algılayan master kol, master kolun hareketlerini takip eden slave kol, master koldan alınan analog sinyali bilgisayar ortamına sayısal veri olarak aktarıp slave kolun kontrol edilmesini sağlayan Veri toplama kartı, veri toplama kartından alınan analog sinyal ile servo motorların kontrolünü sağlayan Servo motor sürücü kartı ve tüm bu işlemleri bilgisayar ortamında yapılmasını sağlayan kontrol yazılımından oluşmaktadır.

3.3.1.1 Hareketleri Algılayan Kol (Master Kol) ve Yapısı

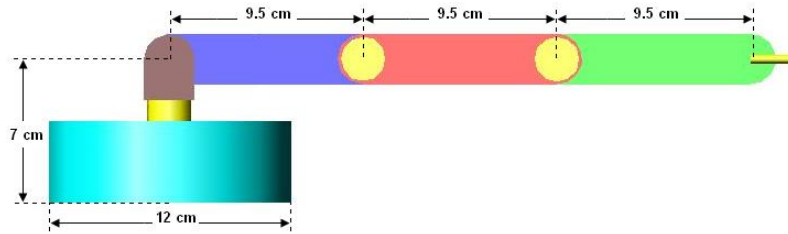
Hareketleri algılayan master kol insan eliyle hareket ettirilerek eklemleri bir açı değeri alır. Bu açı değerlerinin bilgisayar yazılımı ile okunabilmesi için bilgisayarın algılayabileceği bir biçime dönüştürülmesi gerekir. Bu nedenle eklemlere hassas potansiyometreler yerleştirilerek eklem açıları analog elektrik sinyaline dönüştürülür. Böylece bu analog sinyaller veri toplama kartıyla bilgisayar yazılımında işlenebilir bir biçime dönüştürülmüş olur.



Şekil 3.58 Master kol

3.3.1.1.1 Master Kolun Mekanik Yapısı

Master kol dört serbestlik dereceli dört dönel ekleme ve uç işlevcisinin açıklığını kontrol eden bir ekleme sahiptir. Şekilde görüldüğü gibi bir ana çerçeveye paralel dönen bir dönel eklem ve ana çerçeveye dik dönen üç dönel eklemden oluşmaktadır. Her eklemin dönme açısı 180° ile sınırlandırılmıştır. Ana çerçeve 6 mm Kavak Ağacı Kontraplağı, kollar ise 4 mm Kayın Ağacı Kontraplağından yapılmıştır. Kontraplağın seçilmesinin nedeni kolay bulunabilir ve işlenebilir olmasıdır. Aşağıdaki şekilde master kolun ölçülendirilmesi görülmektedir.



Şekil 3.59 Master kol ölçüleri

3.3.1.1.2 Gerilim Bölücü Mantığı ve Potansiyometreler

Potansiyometreler en basit anlamıyla üç uçlu ayarlı dirençlerdir. Potansiyometreler direnç değişikliğiyle gerilim bölücü olarak görev yaparlar. Aşağıda çeşitli potansiyometreler görülmektedir.



Şekil 3.60 Çeşitli potansiyometreler

Ve Şekil 3.61 'da potansiyometrelerin içyapısı gösterilmektedir.



Şekil 3.61 Potansiyometrenin içyapısı

Potansiyometrelerin sabit iki ucu arasına karbon film veya Nikel-Krom telden sarılmış bobin yerleştirilir. Orta uca bağlı fırça bu karbon film veya Nikel-krom tel üzerinde hareket ettirilerek direnç değişikliği sağlanır. Sabit uçlara bir gerilim uygulandığında orta uç hareket ettirildiğinde orta uç gerilimi sıfır ile kaynak gerilimi arasında herhangi bir değer alabilir. Aşağıda Şekil 3.62 'de potansiyometrenin master kol üzerine yerleştirilmiş hali görülmektedir.

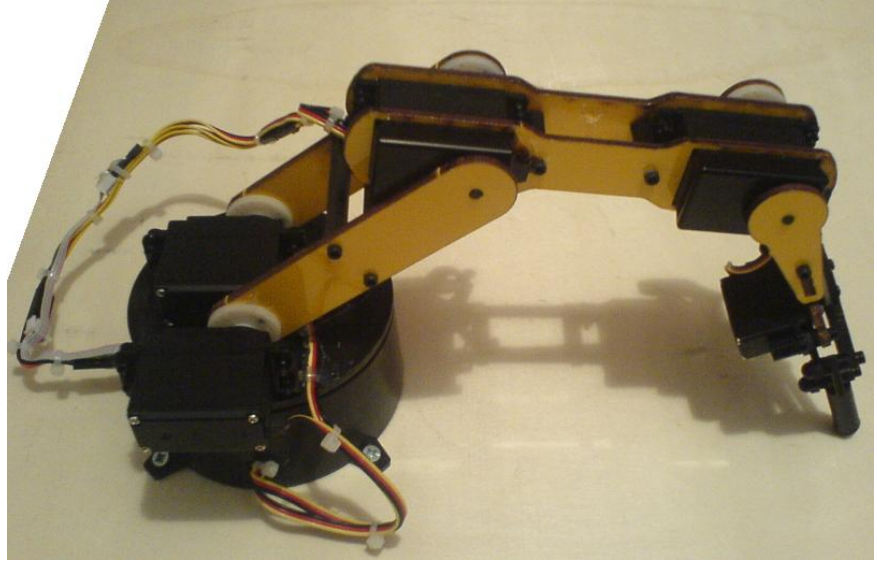


Şekil 3.62 Potansiyometrelerin eklemlere yerleştirilmiş hali

3.3.1.2 Hareketleri Takip Eden Kol (Slave Kol) ve Yapısı

Slave kol yazılıma bağlı olarak master kolun eklem hareketlerine veya yazılımla verilen eklem açılara göre hareket eder. Bunun için bilgisayarda işlenen eklem açılarını veri toplama kartı sayısal sinyalden analog sinyale dönüştürerek servo motorların pozisyon kontrolü için servo motor sürücü kartına gönderir. Analog sinyale dönüştürülen eklem açıları servo motor sürücü kartında Pulse Width Modulation

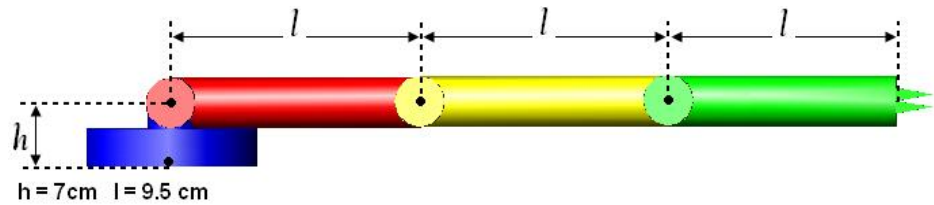
mantığıyla servo motorların mil pozisyonunu (açısını) kontrol eden bir sinyale dönüştürülür.



Şekil 3.63 Slave kol

3.3.1.2.1 Slave Kolun Mekanik Yapısı

Slave kol, master kol gibi dört serbestlik dereceli dört dönel ekleme ve bir tutacağa sahip uç işlevcisine sahiptir. Master kolun yapısıyla aynı olup, tasarımı eklemlere Rc servo motor yerleştirildiği için farklıdır. Her eklemin dönme açısı master kolda olduğu gibi 180° dir. Ana çerçeve plastikten yapılmış olup, kol parçaları 4 mm Lexan Polikarbonat malzemeden yapılmıştır. Aşağıdaki şekilde slave kolun ölçüleri görülmektedir.



Şekil 3.64 Slave kolun ölçüleri

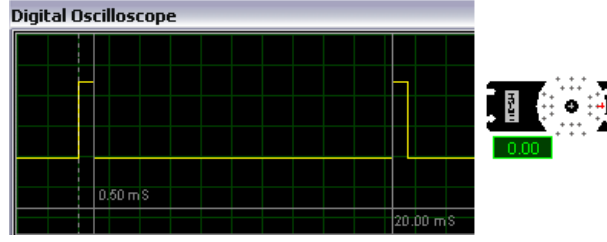
3.3.1.2.2 Servo Motorlar ve Pozisyon Kontrolü

Hobby amaçlı servo motorlar uçak, helikopter, gemi, robot ve bunun gibi model araçlarda pervane, kanat, kol gibi hassasiyet gerektiren parçaların hareket ettirilmesinde kullanılır Şekil 3.65.

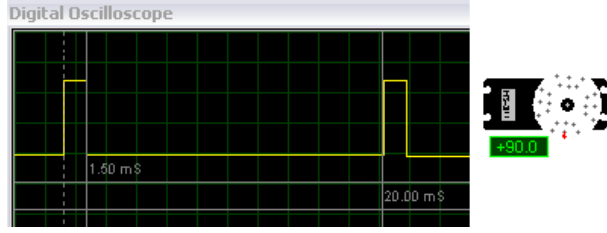


Şekil 3.65 Tipik bir hobby servo motor

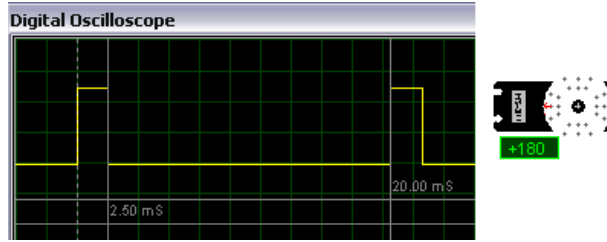
Model araçlarda sık kullanıldığı için piyasada RC (Radio Control) Servo motor olarak bilinir. Servo motorların içyapısındaki redüktör sayesinde 4-6V arasındaki küçük bir gerilim ile yüksek tork sağlarlar. Servo motorların mili normal bir doğru akım veya alternatif akım motorunun mili gibi sürekli dönme hareketi yapmaz. Servo motorların üç tane bağlantı ucu bulunur, bunlar Besleme uçları Vdd, Vss ve Mil pozisyonunu belirleyen kontrol sinyalinin uygulandığı Kontrol ucudur. Servo motorun mili kontrol pinine uygulanan sinyale göre 0–180° arasında bir açı değeri alır. Servo motorun sadece besleme uçlarına Doğru gerilim uygulanırsa herhangi bir hareket yapmaz. Servo motor milinin bir açı değeri alması için kontrol ucuna frekansı 50 Hz olan ve modülasyon genişliği 0,5 ms ile 2,5 ms arasında değişen bir sinyalin uygulanması gerekir. Milin alacağı açı değeri 0,5–2,5 ms ‘lik modülasyon genişliği ile doğru orantılı olarak 0–180° arasında değişir. Örneğin 0,5ms ‘lik bir modülasyon genişliğindeki bir sinyal ile mil 0° Şekil 3.66, 1,5ms ‘lik modülasyon genişliğinde 90° Şekil 3.67, ve 2,5 ms ‘lik modülasyon genişliğinde ise 180° açı değeri alır Şekil 3.68.



Şekil 3.66 0,5ms 'lik modülasyon genişliği ve motor milinin aldığı açı değeri



Şekil 3.67 1,5ms 'lik modülasyon genişliği ve motor milinin aldığı açı değeri



Şekil 3.68 2,5ms 'lik modülasyon genişliği ve motor milinin aldığı açı değeri

3.3.1.3 Servo Motor Sürücü Kontrol Kartı

Servo motor sürücü kartı kol eklemlerinden gelen analog açı bilgisini servo motorların mil pozisyonunu belirlemek için frekansı 50 Hz olan ve modülasyon genişliği 0,5 ms ile 2,5 ms arasında değişen bir sinyale dönüştürür. Servo motor sürücü kartı Analog-Sayısal dönüştürücü bloğu, Sayısal Çıkış bloğu ve PIC besleme bloğundan oluşmaktadır. Veri toplama kartından analog bilgiye dönüştürülen eklem açı bilgileri Mikrodenetleyici yardımıyla 8 bitlik bir sayısal veriye dönüştürülür. Bu 8 bitlik sayısal veri mikrodenetleyicide frekansı 50 Hz olan ve modülasyon genişliği 0,5 ms ile 2,5 ms arasında değişen bir sinyale dönüştürülerek sayısal bir çıkış üretilir. Bu çalışmada mikrodenetleyici olarak PIC16F877/A kullanılmıştır.

3.3.1.3.1 PIC16F877 Teknik Özellikleri

3.3.1.3.1.1 Genel Tanımlama

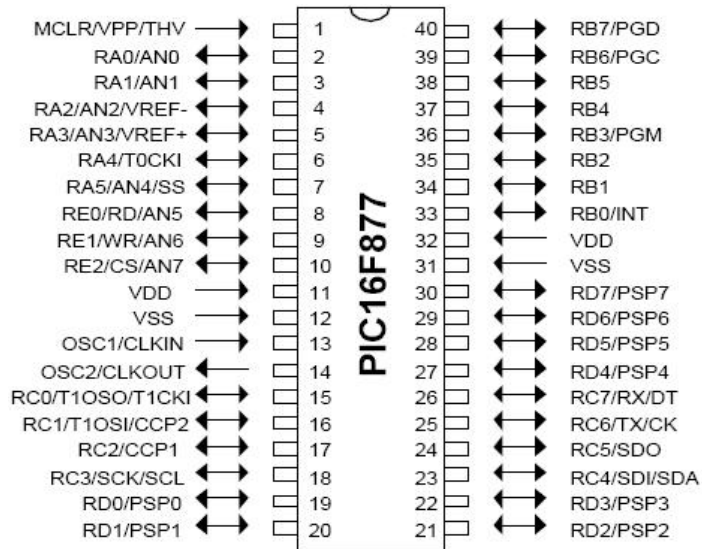
PIC 16F877 yüksek performanslı, CMOS, full-statik, 8 bit mikrodenetleyicidir. Tüm PIC 16/17 mikrodenetleyicileri gibi PIC 16F877 de RISC mimarisini kullanmaktadır. PIC16F87X mikroları birçok esas özelliklere sahiptir. 14 seviyeli, derin küme ve çoklu iç ve dış kesme kaynaklarına sahiptir. 2 aşamalı komut hattı tüm komutların tek bir saykıl' la (çevrimle) işlenmesini sağlamaktadır. Yalnızca bazı özel komutlar 2 saykıl çekerler. Bu komutlar dallanma komutlarıdır. PIC16F87X ailesi dış elemanları azaltacak çok özel özelliklere sahiptir ve böylece maliyet minimuma inmekte, sistemin güvenilirliği artmakta, enerji sarfiyatı azalmaktadır. Bunun yanı sıra tüm PIC'lerde 4 adet osilatör seçeneği mevcuttur. Bunlarda tek pinli RC osilatör, düşük maliyet (4 MHZ) , LP osilatör (Kristal veya seramik rezonatör), enerji sarfiyatını minimize etmekte (asgari akım) (40 KHZ), XT kristal veya seramik rezonatör osilatörü standart hızlı ve HS kristal veya seramik rezonatörlü osilatör çok yüksek hıza sahiptir (20 MHZ). PIC mikrodenetleyicilerinin en büyük özelliği sleep modu özelliğidir. Bu mod sayesinde işlem yapılmadığı durumlarda PIC uyuma moduna geçerek çok düşük akım çeker. Kullanıcı bir kaç iç ve dış kesmelerle PIC' i uyuma modundan çıkarabilmektedir. Yüksek güvenilirlikli Watchdog Timer kendi bünyesindeki çip üstü RC osilatörü ile yazılımı kilitlemeye karşı korumaktadır. PIC16F877 EEPROM program belleği, aynı aygıt paketinin orijinali ve üretimi için kullanılmasına olanak vermektedir. Yeniden programlanabilirliği mikroyu uygulamanın sonundan kaldırmadan kodu güncelleştirmeye izin vermektedir. Bu aygıtın kolayca erişilemediği, fakat prototipinin kod güncelleştirmesi gerekli olduğu durumlarda, birçok uygulamanın geliştirilmesinde yararlıdır. Bunun yanı sıra bu kodun güncelleştirilmesi diğer ayrı uygulamalarda da yararlıdır. [10]

3.3.1.3.1.2 PIC 16F877'nin Genel Özellikleri

- Yüksek hızlı RISC işlemciye sahiptir;
- 35 adet komut mevcuttur;
- Tüm komutlar 1 saykıl çeker, (Dallanma komutları 2 saykıl çeker.);
- 20 Mhz'ye kadar işlem hızına sahiptir;
- 8Kx14 word'lük flash program belleği mevcuttur;
- 368x8 bayt'luk data belleği;
- 256x8 byte'luk EEPROM data belleği;
- PIC16C73B/74B/76/77 ile uyumlu pin yapısı;
- Doğrudan ve dolaylı adresleme;
- Power-on Reset (POR), Power-up Timer (PWRT) , üzerinde bulunan RC osilatör ile çalışan Watchdog Timer (WDT);
- Programlanabilen kod koruma;
- Enerji tasarrufu için uyku (SLEEP) modu;
- Düşük güçlü yüksek hızlı CMOSFLASH/EEPROM teknolojisi;
- Tamamen statik dizayn;
- Devre üzerinde seri programlama;
- 5 V'luk kaynak ile çalışma;
- 2 V ile 5.5 V arasında işlem yapabilme özelliği;
- Düşük güç harcaması;
 - < 2 mA typical @ 5V, 4 MHz
 - < 20 mA typical @ 3V, 32 kHz
 - < 1 mA typical standby

3.3.1.3.1.3 PIC 16F877'nin Belirleyici Özellikleri

- Timer0: 8 bit prescaler'e sahip 8bit zamanlayıcı/sayıcı,
- Timer1: Sleep modunda artış gösterebilen ve harici saat darbesiyle artırılabilen Prescaler' li 16 bit zamanlayıcı/sayıcı,
- Timer2: 8 bit periyot kaydedicili, prescaler ve postscalerli 16bit zamanlayıcı/sayıcı,
- İki adet tutma, karşılaştırma, PWM modülü
- 200ns çözünürlükte 16 bitlik karşılaştırma
- 10 bit çözünürlükte PWM
- 10 bit çok kanallı Analog-Dijital çevirici
- Seri port ve I2C modülleri
- 9 bit adres saptamaya sahip USART/SCI



Şekil 3.69 PIC 16F877'nin bacak yapısı



Şekil 3.70 PIC 16F877'nin fiziksel görünüşü

3.3.1.3.1.4 PIC 16F877 Pin Tanımlamaları

Pin Adı	Pin	Giriş/Çıkış	Standart	Açıklama
OSC1/CLKIN	13	G	ST/CMOS	Kristal/Harici osilator girişi.
OSC2/CLKOUT	14	Ç	-	Kristal/Harici osilator çıkışı.
MCLR/VPP/THV	1	G/Ç	ST	Mikrodenetleyici resetleme pini.
RA0/AN0	2	G/Ç	TTL	PORTA: Giriş/Çıkış olarak tanımlanabilir. Bunun Dışında Analog Giriş olarak ta tanımlanabilir. RA0...5/AN0...5: Analog giriş RA4: TIMER0 için Clock girişi
RA1/AN1	3	G/Ç	TTL	
RA2/AN2/VREF-	4	G/Ç	TTL	
RA3/AN3/VREF+	5	G/Ç	TTL	
RA4/T0CKI	6	G/Ç	ST	
RA5/SS/AN4	7	G/Ç	TTL	
RB0/INT	33	G/Ç	TTL/ST	PORTB: Giriş/Çıkış olarak tanımlanabilir. Tüm girişlerinde yazılımla programlanabilir pull-up dirençleri vardır. Bunların dışında; RB0: Harici kesme ucu. RB3: Düşük seviye programlama girişi. RB6: Seri programlama girişi. RB7: Seri programlama data girişi
RB1	34	G/Ç	TTL	
RB2	35	G/Ç	TTL	
RB3/PGM	36	G/Ç	TTL	
RB4	37	G/Ç	TTL	
RB5	38	G/Ç	TTL	
RB6/PGC	39	G/Ç	TTL/ST	
RB7/PGD	40	G/Ç	TTL/ST	
RC0/T1OSO/T1CKI	15	G/Ç	ST	PORTC: Giriş/Çıkış olarak tanımlanabilir. RC0:TIMER1 osilatör/clock çıkışı. RC1:TIMER1 osilatör girişi veya Capture2-G/Compare2-Ç/PWM2 çıkışı RC2:Capture1-G/Compare1-O/PWM1 çıkışı RC3:SPI ve I2C Mod. Senkron seri clock G/Ç RC4:SPI modunda data giriş I2C modunda G/Ç RC5: SPI modunda data çıkış RC6:USART senkron gönderme veya senkron clock RC7:USART senkron alma veya senkron data görevi
RC1/T1OSI/CCP2	16	G/Ç	ST	
RC2/CCP1	17	G/Ç	ST	
RC3/SCK/SCL	18	G/Ç	ST	
RC4/SDI/SDA	23	G/Ç	ST	
RC5/SDO	24	G/Ç	ST	
RC6/TX/CK	25	G/Ç	ST	
RC7/RX/DT	26	G/Ç	ST	
RD0/PSP0	19	G/Ç	ST/TTL	PORTD: Giriş/Çıkış olarak tanımlanabilir. Bunun dışında mikro işlemci hattında arabirim olarak kullanıldığında paralel slave port olarak kullanılabilir.
RD1/PSP1	20	G/Ç	ST/TTL	
RD2/PSP2	21	G/Ç	ST/TTL	
RD3/PSP3	22	G/Ç	ST/TTL	
RD4/PSP4	27	G/Ç	ST/TTL	
RD5/PSP5	28	G/Ç	ST/TTL	
RD6/PSP6	29	G/Ç	ST/TTL	
RD7/PSP7	30	G/Ç	ST/TTL	
RE0/RD/AN5	8	G/Ç	ST/TTL	PORTE: Giriş/Çıkış olarak tanımlanabilir. Bunun dışında RE0:Paralel Slave porttan okuma kontrolü veya 5. Analog Giriş. RE1: Paralel Slave porttan yazma kontrolü veya 6. Analog Giriş RE2: Paralel Slave porttan seçim kontrolü veya 7. Analog Giriş
RE1/WR/AN6	9	G/Ç	ST/TTL	
RE2/CS/AN7	10	G/Ç	ST/TTL	
VSS	12,31	P	-	Mikrodenetleyici için toprak seviyesi.
VDD	11,32	P	-	Mikrodenetleyici için pozitif kaynak

Tablo 3.8 PIC 16F877'de Pin Tanımlamaları

3.3.1.3.1.5 Giriş/Çıkış Portları (G/Ç Portları)

G/Ç portları giriş/çıkış işlemlerinin dışında bazı çevresel işlemleri de yapacak özelliklere sahiptirler. Çevre birimleri kullanıldığında genel amaçlı giriş/çıkış pini kullanılmaz.

3.3.1.3.1.5.1 PORTA ve TRISA Kaydedicisi

PORTA 6 bit giriş/çıkış olarak yönlendirilebilir porttur. Bu portu yönlendiren yazmaç ise TRISA yazmacıdır. TRISA kaydındaki herhangi bir bit 1 ise buna uygun çıkış sürücüsü yüksek direnç moduna getirilecektir. TRISA kaydındaki herhangi bir bitin 0 olması durumunda ise çıkış mandalı seçilen pinin üzerine getirilir. Analog giriş kullanıldığında TRISA yazmacı RA pininin yönünü kontrol eder.

3.3.1.3.1.5.2 PORTB ve TRISB Kaydedicisi

PORTB 8 bit giriş/çıkış olarak yönlendirilebilir porttur. Bu portu yönlendiren yazmaç ise TRISB yazmacıdır. TRISB kaydındaki herhangi bir bit 1 ise buna uygun çıkış sürücüsü yüksek direnç moduna getirilecektir. TRISB kaydındaki herhangi bir bitin 0 olması durumunda ise çıkış mandalı seçilen pinin üzerine getirilir. Analog giriş kullanıldığında TRISB yazmacı RB pininin yönünü kontrol eder. Her bir PORTB pini iç direnç düşürücü engellere sahiptir. RBPU(OPTION –REG<7) bitinin silinmesiyle aktif yapılır. Düşürücü engeller, port pini çıkış olarak konfigüre edildiği zaman otomatik olarak kapanmaktadır. Ayrıca dört PORTB pini, RB7:RB4 değişim özelliklerinde kesmelere sahiptir. Yalnızca giriş olarak konfigüre edilen pinler kesmenin meydana gelmesine sebep olabilirler. (yani, herhangi bir çıkış olarak şekillendirilen RB7:RB4 pini değişim ilişkisi üzerindeki kesmeden hariç tutulmuştur.)Giriş modundaki pinlerin

değeri PORTB` nin önceki okunmasındaki eski değeri ile karşılaştırılır. Pinlerin “uyuşmayan” kısımları RB port değişim kesmesini üretmek için birlikte OR’lanır.

3.3.1.3.1.5.3 PORTC ve TRISC Kaydedicisi

PORTC 8 bit G/Ç olarak yönlendirilebilir porttur. Bu portu yönlendiren yazmaç ise TRISC yazmacıdır. TRISC Kaydındaki herhangi bir bit 1 ise buna uygun çıkış sürücüsü yüksek direnç moduna getirilecektir. TRISC Kaydındaki herhangi bir bitin 0 olması durumunda ise çıkış mandalı seçilen pinin üzerine getirilir. Analog giriş kullanıldığında TRISC yazmacı RC pininin yönünü kontrol eder.

3.3.1.3.1.5.4 PORTD ve TRISD Kaydedicisi

PORTD 8 bit Schmitt trigger tampon girişli porttur. Bu port PSPMODE (TRISD) denetim biti tarafından kurulur. Giriş tamponu TTL tampondur.

3.3.1.3.1.5.5 PORTE ve TRISE Kaydedicisi

Bu port RE0/RD/AN5, RE1/WR/AN6 ve RE2/CS/AN7 olmak üzere 3 adet pine sahiptir. Giriş veya çıkış portu olarak ayarlanabilir. PORTE kontrolü TRISE tarafından yapılır. PSPMODE(TRISE<4>) denetim biti tarafından ayarlanır.

3.3.1.3.1.6 ADC (Analog Dijital Konvertör) Modülü

PIC 16 F877'de 8 kanallı 10 bit'e kadar çevirme işlemi yapabilen bir analog-dijital çevirici (ADC) modülü bulunmaktadır. PIC 16F877 üzerindeki ADC modülün çalışması şu şekildedir. Analog giriş örnekle ve tut kondansatörünü şarj eder. Örnekle ve tut kondansatörünün çıkışı dönüştürücünün girişine uygulanır. Dönüştürücü, ardışık yaklaştırma yoluyla bu analog düzeyin sayısal sonucunu üretir. Bu A/D dönüşümde, analog giriş sinyali 10 bitlik sayı karşılaştırma ile sonuçlanır. ADC eşsiz bir özelliğe sahiptir. İşlem yapmazken uyuma moduna geçer.

Uyuma modunda ADC'nin saatinde bir iç RC osilatörü üretilmelidir. ADC Modül dört (4) kaydediciye sahiptir.

Bunlar;

1- A/D Yüksek sonuç kaydedicisi (ADRESH)

2- A/D Düşük sonuç kaydedicisi (ADRESL)

3- A/D Kontrol kaydedici 0 (ADCON0)

4- A/D Kontrol kaydedici 1 (ADCON1)

A/D çeviricinin kontrolünü ADCON0 ve ADCON1 kaydedicileri sağlamaktadır.

3.3.1.3.1.6.1 ADCON0 Kaydedicisi

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.71 ADCON0 Kaydedicisinin Bit Yerleşimi

Adcon,0(ADON):A/D çeviriciyi yetkilendirme bitidir. ADON=1 ise A/D çevirici açıktır ve işlem yapılabilir durumdadır. ADON=0 ise A/D çevirici kapalıdır.

Adcon,2(GO/DONE): Eğer ADCON biti de 1 ise A/D çevirici statü biti görevini üstlenir. GO/DONE=1 ise A/D çevirici işlem yapıyor demektir. GO/DONE=0 ise A/D çevirici üzerinde herhangi bir işlem yapılmıyordur.

Adcon,3:5(CHS0:CHS2):A/D çevirici için kanal seçim bitlerini oluşturur. Bu bitlere verilecek değerlerle A/D çevirme için hangi kanalın seçileceği belirlenir. Daha önceden de belirtildiği gibi PIC 16F877’de A/D çevirici için 8 kanal mevcuttur.

000 = kanal 0, (RA0/AN0)

001 = kanal 1, (RA1/AN1)

010 = kanal 2, (RA2/AN2)

011 = kanal 3, (RA3/AN3)

100 = kanal 4, (RA5/AN4)

101 = kanal 5, (RE0/AN5)

110 = kanal 6, (RE1/AN6)

111 = kanal 7, (RE2/AN7)

Adcon,6:7(ADCS0:ADCS1): A/D çevirici için clock frekansı seçim bitleridir. Bu bitlere verilecek değerler ile A/D çevirme işlemi esnasında kullanılacak frekans değeri bize sunulan değerler içerisinde seçilir.

00 = FOSC/2

01 = FOSC/8

10 = FOSC/32

11 = FRC (Harici bir RC osilasyon kaynağından gelen clock darbeleri kullanılır.)

3.3.1.3.1.6.2 ADCON1 Kaydedicisi

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.72 ADCON1 Yazmacının Bit Yerleşimi

Adcon1,0:3(PCFG0:PCFG3): A/D çevirici portunun biçimini düzenlemeyi sağlayan bitlerdir. Yani A/D çevirme işleminde kullanılacak pinlerin nasıl davranacağını belirlememize olanak sağlarlar.

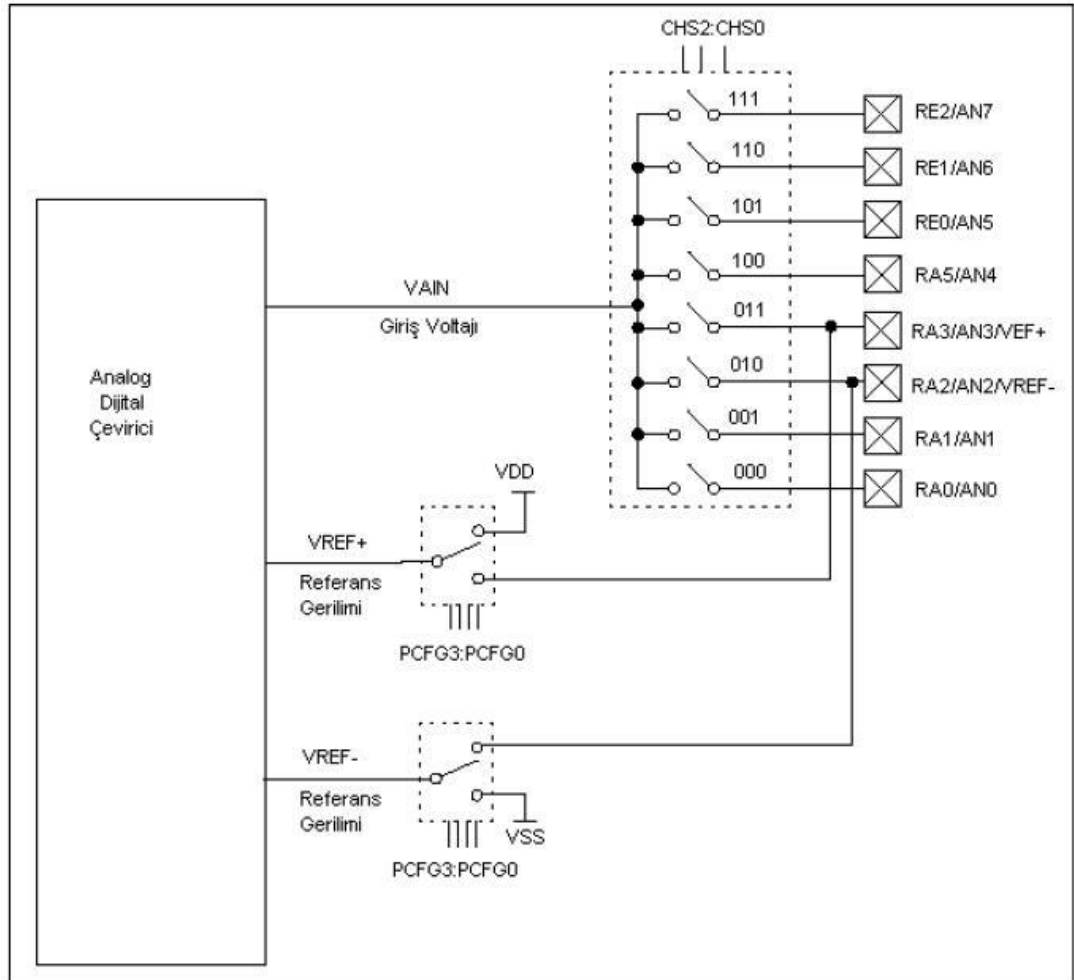
PCFG3: PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	KANAL/ REF
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0111	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

Tablo 3.9 ADCON1 Yazmacı tablosu

Bu tabloda A=Analog giriş D=Dijital giriş/çıkış anlamına gelir.

Adcon1,7(ADFM): A/D çevirme işlemi esnasında meydana gelen verinin biçimini belirlemeye yarayan bittir. ADFM=1 ise ADRESH kaydedicisinin MSB kısmındaki altı biti 0 kabul edilir ve A/D çevirme sonucunda elde edilen veri ADRESH'ın 2 bitlik LSB kısmına ve ADRESL'ye yazılır. ADFM=0 ise ADRESL'nin Lsb kısmındaki 6 biti 0 kabul edilir ve A/D çevirme sonucu elde edilen veri ADRESL'nin son iki bitine ve ADRESH'a yazılır.

ADRESH: ADRESL kaydedicileri A/D dönüşümün 10 bit sonucunu kapsar. A/D dönüşümü bittiği zaman, sonuç A/D sonuç kaydedicisinin içine yüklenir. A/D modülü Şekil 3.73 'de görülmektedir.



Şekil 3.73 A/D Blok Diyagramı

A/D Modülü biçimlendirildikten sonra, dönüştürme işlemi başlamadan önce kanal seçilmiş olmalıdır. Analog giriş kanallarında ilgili TRIS bitleri giriş için seçilmiş olmalıdır. Aşağıdaki adımlar, A/D dönüşüm yapmak için takip edilmelidir.

1- A/D Modülü Konfigürasyonu

Analog pinler, referans voltajları ve digital I/O konfigürasyonu (ADCON1)

- A/D giriş kanalı seçimi (ADCON0)
- A/D dönüşüm saat sekimi (ADCON0)
- A/D Modülünü açma

2- A/D Kesme Konfigürasyonu

- ADIF bitinin temizlenmesi
- ADIE bitinin ayarlanması
- GIE bitinin ayarlanması

3- Gerekli zamanı bekleme işlemi

4- Dönüşümün başlaması

- GO/DONE bitinin ayarlanması (ADCON0)

5- A/D dönüşümünün beklenmesi

6- A/D dönüşüm sonucunu okuma ve kaydetme

7- Diğer dönüşüm için 1. ve 2. kez adımları tekrarlama

3.3.1.3.1.6.3 A/D Dönüşüm Saatinin Seçimi

TAD bit başına A/D dönüşüm zamanı olarak tanımlanır. 10 bit A/D dönüşüm için maksimum 12 TAD gerekir. TAD seçimi için mümkün olan 4 seçenek vardır.

- 2TOSC
- 8TOSC
- 32TOSC
- Dahili RC Osilatörü

A/D Dönüşümün doğru olarak yapılması için, TAD minimum 1,6 seçilmiş olmalıdır.

3.3.1.3.1.6.4 Analog Port Pinlerini Yapılandırma

ADCON1 ve TRIS kaydedicileri kontrol ve işletim port pinleridir. Analog girişlerin TRU bitlerinin karşılıklı olarak ayarlanması gerekir. TRIS biti temizlenmiş ise dijital çıkış seviyesine (VOH veya VOL) dönüştürme yapılmış demektir.

3.3.1.3.1.6.5 A/D Dönüşümü Sırasında Uyuma

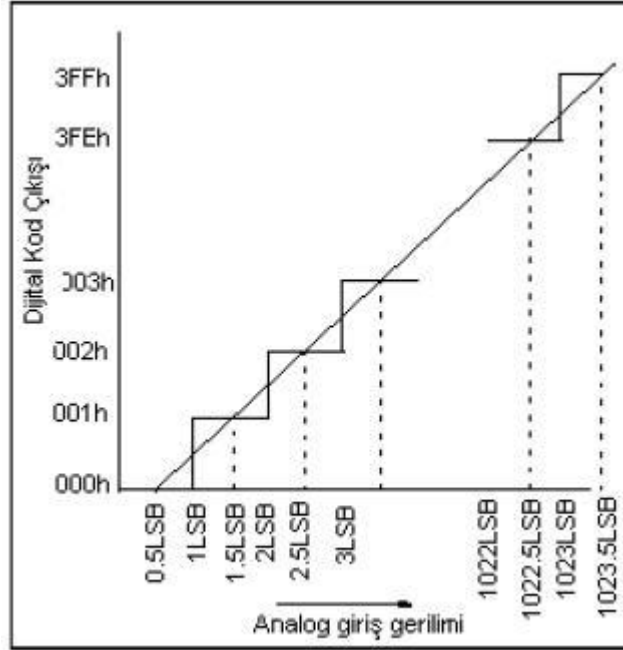
A/D Modül işletim sırasında uyuma modunda olacaktır. Bunun için A/D saat kaynağı ayarlanmalıdır. (ADCS1:ADCS0=11). RC saat kaynağı seçildiği zaman dönüştürme başlamadan önce A/D modül bir saat çevrimi süresi kadar bekler. Bu uyku talimatına izin verir. Dönüşümdeki tüm sayısal anahtarlama gürültüsü elemine edilmiş olur. Dönüştürme işlemi bittiği zaman GO/DONE biti temizlenir ve sonuç adres kaydedicisi içine yüklenir. A/D kesmesi etkinleştirilirse aygıt uykudan uyanır. A/D kesmesi pasifleştirilirse A/D modülü kapalı duruma dönse de bir süre açık kalacaktır.

3.3.1.3.1.6.6 A/D Doğruluk/Hata

Aygıt frekansı RC saatin alçak kullanımı olduğu sistemlerde tercih edilir. Yüksek frekans azaltılırken, TAD aygıt osilatörü türetilmelidir. A/D Dönüştürücü için belirtilen salt doğruluk, miktar ölçme hatası, integral hata, türevsel hata, tam skala hata, sapma hatalarının katkıları toplamını içerir. Herhangi bir kod için ideal geçişe karşı, güncel geçişten maksimum sapma olarak tanımlanır. Verilen bir analog giriş aralığı için sayısal çıkış kodu aynı olur. Bu sayısal kod analog girişten ölçülen miktar kadardır. Analogtan dijitale ölçme işleminde hata tipik olarak $\frac{1}{2}$ LSB kadardır.

3.3.1.3.1.6.7 Transfer Fonksiyonu

A/D Dönüştürücünün transfer fonksiyonu aşağıda gösterildiği gibidir. Analog giriş voltajı/1024 ile bulunur.



Şekil 3.74 A/D Transfer Fonksiyonu

3.3.1.3.1.7 CPU ' nun Spesifik Özellikleri

Mikrodenetleyici 'yi diğer işlemcilerden ayıran şey, gerçek zaman uygulamalarının gereksinimleri ile ilgili özel devreleridir. PIC16F877' te sistem güvenliğini maksimize eden, dış elemanları ayırarak maliyeti minimize eden, güç tasarrufu, çalışma modu ve kod koruma gibi özellikleri taşımaktadır. Bu özellikler;

- OSC seçimi
- Reset

- Güç kaynağı reseti (POR)
- Yüksek güç timer (PWRT)
- Osilatör başlangıç Timer (OST)
- Kesmeler
- Watchdog Timer
- Sleep
- Kod koruma
- ID yerleşimleri
- Devre içi seri programlama

PIC16F877 'te yalnızca konfigürasyon bitleri tarafından kapatılabilen Watchdog Timer mevcuttur. Güvenliği arttırmak için bu kendi RC osilatörünü de çalıştırmaktadır. Yüksek güçte gereken esas gecikmeleri sağlayan 2 Timer mevcuttur. Bunlardan birisi Osilatör Başlangıç Timer 'dır. Bu timer, kristal osilatör durgunlaşınca kadar çipi resette tutar. Diğer timer ise yalnızca nominal yüksek güçte 72 ms sabit gecikme üreten Yüksek Güç Timer' ıdır. Bu güç kaynağı stabilize olurken aygıtı resette tutar. Bu iki çip üzeri Timer ile, uygulamaların çoğu hiçbir reset devrelerini gerektirmemektedir. SLEEP modu çok düşük enerjili alçak güç modunu sunmaktadır. Kullanıcı SLEEP ten çıkmak için dış reset, Watchdog Timer zaman aralığı veya kesmeleri kullanabilir. Bazı osilatör seçenekleri, kısımları uygulamaya yerleştirmek için elde edilmektedir. RC osilatör seçeneği sistem maliyetini, LP kristal seçeneği ise güç sarfiyatını düşürmektedir. Çeşitli seçenekleri seçmek için konfigürasyon bitler seti kullanılmaktadır.

3.3.1.3.1.7.1 Biçimlendirme (Konfigürasyon) Bitleri

Biçimlendirme bitleri çeşitli aygıt işlevlerini seçmek için programlanabilir ('0' olarak okur) yada programlamadan bırakılabilir. ('1' olarak okur) Bu bitler 2007h program bellek yerleşiminde saklanır. 2007h adresi kullanıcı program bellek biriminin ötesindedir ve özel test/biçim bellek birimine (2000h- 3FFFh) aittir. Bu birime yalnızca programlama sürecinde erişilebilir.

CP1	CP0	BKBUG	-	WRT	CPD	LVP	BODEN	CP1	CP0	PWRT	WDTE	FOSC1	FOSC0
Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.75 Biçimlendirme bitlerinin bellekteki bit dizilimi

Bit 0,1 (FOSC0,FOSC1): Osilatör seçme bitleridir. Yapılacak uygulamada hangi tür osilatör kullanılacağını tayin eder. (RC, XT, HS, LP) FOSC0 ve FOSC1 bitlerinin aldığı durumlara göre kullanılacak osilatör türleri Tablo 7.10 'da verilmiştir.

FOSC1	FOSC0	OSC
1	1	RC
1	0	HS
0	1	XT
0	0	LP

Tablo 3.10 Osilatör seçim tablosu

Bit 2 (WDTE): Watchdog timer'in kontrol edildiği bittir. Bu bitin 1 olması WDT 'ı devreye sokarken, 0 olması ile WDT devreden çıkarılır.

Bit 3():Power up timer kontrol bitidir. Bu bitin 1 olması ile PWRT devreden çıkarılır, 0 olması ile ise PWRT devreye sokulur.

Bit 6 (BODEN): Brown out reset özelliğine ait yetkilendirme bitidir. BODEN=1 ise BOR kullanım dışı, BODEN=0 ise BOR etkindir. Bit 7 (LVP): Düşük gerilim programlaması için yetkilendirme bitidir. LVP=1 ise RB3/PGM pinin PGM özelliği devreye girer. LVP=0 olduğunda ise RB3 özelliği kullanılabilir.

Bit 8(CPD): Bellek EEPROM verisi kod koruması için yetkilendirme bitidir. CPD=1 ise kod koruması devre dışıdır. CPD=0 ise kod koruma etkindir ve EEPROM içerisine kayıtlı veriye bir teşebbüste bulunulamaz.

Bit 9(WRT): Flash program belleğine yazma yetkilendirme bitidir. WRT=1 ise korumasız program belleğine EECON kontrolünde veri yazılabilir. WRT=0 ise korumasız program belleğine EECON kontrolünde veri yazılamaz.

Bit 11(DEBUG): Devre üzeri seri programlama yetkilendirme bitidir. DEBUG=1 ise RB6, RB7 normal G/Ç işlemlerinde kullanılır. DEBUG=0 ise RB6, RB7 aygıt içerisine program yerleştirme işleminde kullanılır.

Bit 12:13 (CP0:CP1):Yapılan uygulamanın kopyalamaya veya herhangi bir teşebbüse karşı kod koruma özelliğinin konması işlemini kontrol eden bitlerdir. Bu bitlerin aldığı farklı değerler ile farklı kod koruma seçenekleri söz konusu olmaktadır.

11 = Kod koruma kapalı

10 = 1F00h tan 1FFFh'a kadar kod koruması

01 = 1000h tan 1FFFh'a kadar kod koruması

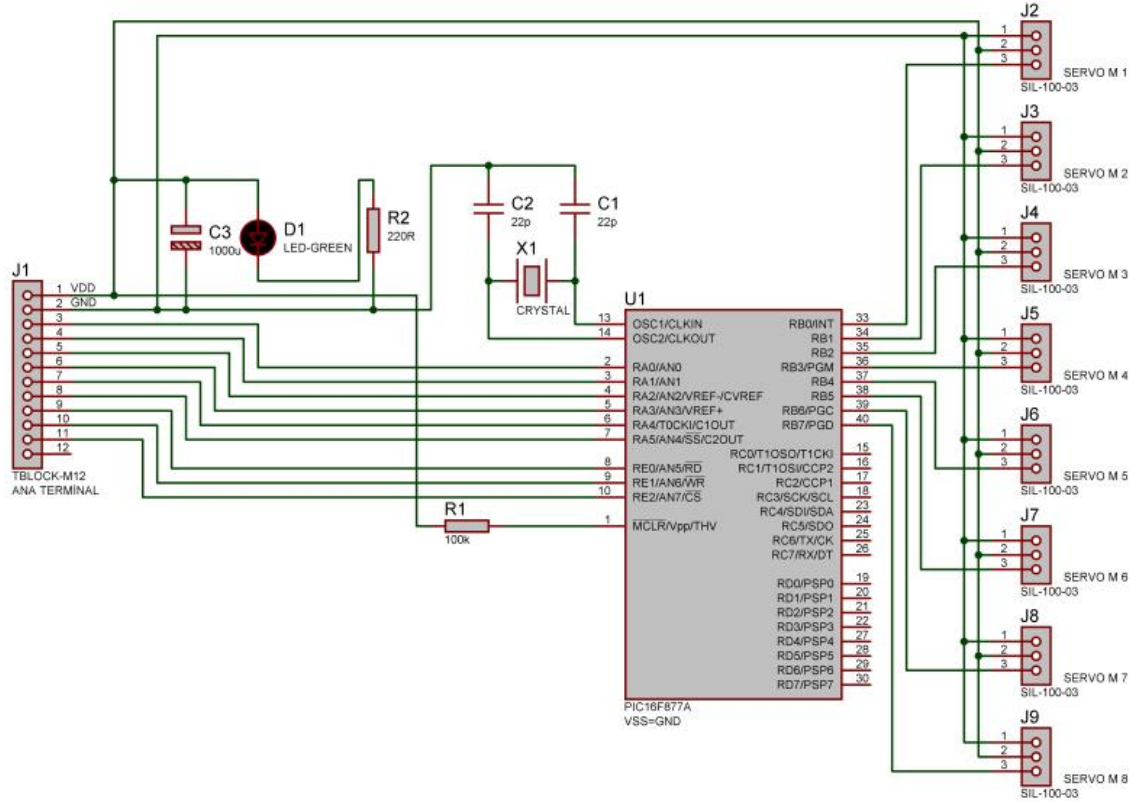
00 = 0000h tan 1FFFh'a kadar kod koruması

3.3.1.3.1.7.2 Osilatör Tipleri

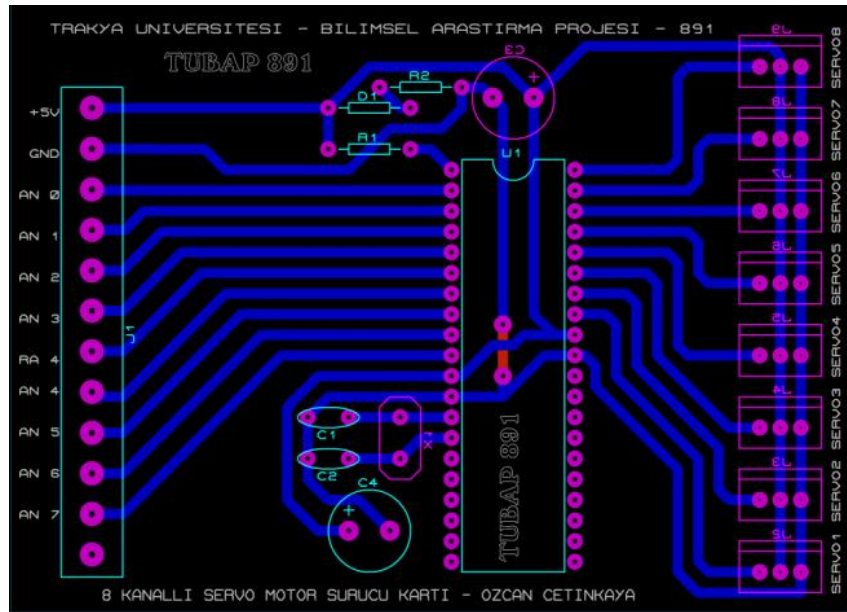
PIC16F877 mikrodnetleyicilerinde 4 çeşit osilatör tipi bulunmaktadır. Kullanıcı bu 4 moddan birini seçerek iki biçimlendirme bitini (FOSC1 ve FO SC2) programlayabilir.

- LP Kristal veya seramik rezonatör-asgari akım - 40Khz
- XT Kristal veya seramik rezonatör- genel amaçlı - 4Mhz
- HS Kristal veya seramik rezonatör- yüksek hız - 20 Mhz
- RC Direnç/ Kapasitör zaman sabitli - düşük maliyet - 4Mhz

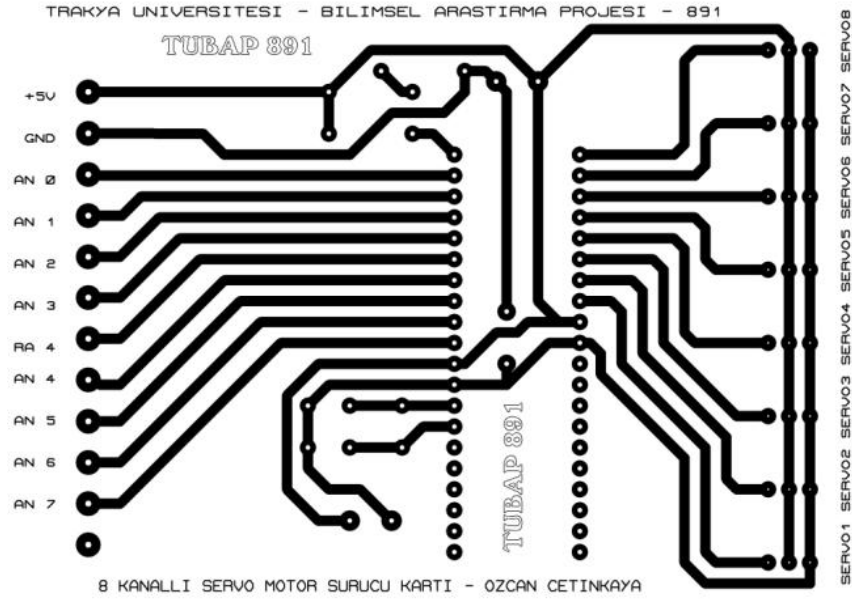
3.3.1.3.2 Kontrol Kartı Devre Şeması ve Baskı devresi



Şekil 3.76 Servo Motor Sürücü Kartı Devre Şeması



Şekil 3.77 Baskı Devreye Elemanların Yerleşimi



Şekil 3.78 Baskı devre

3.3.1.3.3 Mikrodenetleyici Yazılımı

ISIS ile hazırlanan elektronik devrenin simülasyonunun yapılabilmesi ve gerçek hayatta çalıştırabilmesi için Mikrodenetleyiciye bir yazılımın yüklenmesi gerekir. Bu yazılım çalışmamızdaki ihtiyaçları karşılıyor olmalı ki devremiz doğru ve kararlı çalışabilsin. Yazılımı hazırlamaya başlamadan önce hangi programlama dilinin kullanılacağını tespit etmemiz gerekir. Bunun için Assembler, PICBASIC, CCS C ve bunlar gibi birkaç programla dili kullanılabilir. Çalışmamızdaki yazılımı orta seviye programlama dili olan PICBASIC ile hazırladık. Öncelikle mikrodenetleyicimizin potansiyometreden gelen analog bilgiyi kaç bitlik çözünürlükle sayısal (digital) sinyale dönüştüreceğini, Clock Frekans kaynağını, Analog-digital dönüştürücünün örnekleme periyodunu ve elektronik devrede kullandığımız osilatörün frekansını belirtmemiz gerekir. Bunun için sırasıyla aşağıdaki tanımlamalar yapılır.

DEFINE ADC_BITS 8 ; Çözünürlük biti belirlenir

DEFINE ADC_CLOCK 3 ; Clock kaynağı belirlenir (Dâhili Osilatör)

DEFINE ADC_SAMPLEUS 100 ; 100 mikrosaniye aralıkla örnekleme yapılır.

DEFINE OSC 10 ; Osilatör frekansı belirtilir. (10 MHz)

Daha sonra değişkenler tanımlanır. Bu çalışmada 5 giriş ve 5 çıkış değişkeni ve 1 tanede periyodu tanımlayan olmak üzere 9 tane değişkenimiz bulunmaktadır. ADC_BILGI_? Giriş değişkeleri olup potansiyometreden gelen analog bilginin karşılığı olan 8 bitlik değişkenlerdir. SERVO_? Değişkeleri ise çıkış değişkenleri olup her servo motorun mil konumunu belirleyen modülasyon genişliğine ait değişkenlerdir, PER değişkeni de oluşturulacak sinyalin periyodunu tanımlayan değişkendir. Değişkenler WORD tipi (-32768-65536) olarak tanımlanmışlardır.

ADC_BILGI_0 VAR WORD

ADC_BILGI_1 VAR WORD

ADC_BILGI_2 VAR WORD

ADC_BILGI_3 VAR WORD

ADC_BILGI_4 VAR WORD

SERVO_0 VAR WORD

SERVO_1 VAR WORD

SERVO_2 VAR WORD

SERVO_3 VAR WORD

SERVO_4 VAR WORD

PER VAR WORD

Değişkenler tanımlandıktan sonra giriş ve çıkış için kullanacağımız uçların tanımlanması gerekir. Eğer herhangi bir portun TRIS Registerinin ilgili bitine 1 değeri yüklenirse o portun ilgili ucu giriş ucu olarak, 0 değeri yüklenirse çıkış ucu olarak

tanımlanmış olur. O halde PortA ‘nın TRISA registerinin ilk beş bitine 1 değeri, portb ‘nin TRISB registerinin ilk beş bitine de 0 değeri yüklenmelidir.

TRISA=%00011111 ;PortA ‘nın tüm uçları giriş pini olarak tanımlandı

TRISB=%00000000 ;PortB ‘nın tüm uçları çıkış pini olarak tanımlandı. (% ikilik sayı sistemini ifade eder)

Bütün tanımlamaları yaptıktan sonra ADC ‘nin ADCON1 registerinin ayarlanması gerekir.

ADCON1=%00000000 ; ADCON1 registeri çalışmamıza göre ayarlanır.

Burada ilk dört bit hangi uçların Analog giriş için kullanılacağını belirtmek için kullanılır. Bunu 0000 olarak atadık bu 8 Analog girişinde kullanılabileceğini belirtir. Sonraki iki bit kullanılmıyor sıfır olması gerekir, 6. bit Clock frekansını seçmek için kullanılır burada dahili osilatör frekansını kullandık, 7. son bit ise analog girişlerden okunan verinin sola veya sağa dayalı olmasını belirler biz 8 bit çözünürlük kullandığımız için sağa dayalı olmasını tercih ettik.

Şimdi analog-digital dönüşümün yapılp modülasyon genişliğine dönüştürülecek olan programın yazılmasına geldi. Potansiyometreden gelen analog bilgi mikrodenetleyicide 0–255 arasında (8 bit) bir değere dönüştürülür. Bu bilgi Analog giriş olarak tanımlanan uçlardan ADCIN komutuyla okunur. Yazılımda *BAK: IF ADCON0.2=1 THEN BAK* komut satırıyla ADC analog girişlerden okuma tamamlanıncaya kadar beklemeyi sağlar. Analog girişlerden okuma tamamlandığında ise okumayı keserek bir sonraki komuttan işlemler devam eder. ADCIN komutu *ADCIN*, ‘analog giriş portu’ şeklinde kullanılır.

Örneğin, *ADCIN 0,ADC_BILGI_0*; komutuyla 0. Analog giriş pininden okunan analog bilgi *ADC_BILGI_0* değişkenine atanmış olur. Bu bilginin modülasyon genişliği olarak çıkışa aktarılması gerekir. Bunun için Pulsout komutunu kullanılır. Pulsout komutu belirlenen pinden belirlenen süre kadar Lojik–1 sinyali üretir. “*Pulsout pin, bekleme süresi*” şeklinde kullanılır. Burada bekleme süresi ms cinsinden olup mikrodenetleyicide kullanılan osilatör frekansına göre değişmektedir. Pulsout

komutunun 4MHz osilatör frekansında işleme süresi (komut saykılı) $10\mu s$ 'dir. Şimdi kısa bir hesaplama yapalım. Osilatör frekansı her zaman komut frekansının 4 katıdır, Bizim devremizdeki osilatör frekansı 10MHz öyleyse mikrodenetleyicinin komut saykılı $(10 * 4)/10 = 4\mu s$ 'dir. Buradan anlaşılan Pulsout komutunda bekleme süresine yazacağımız her bir değer bize $4\mu s$ 'lik gecikme yaratacaktır. Bize 0.5–2.5ms arasındaki bir gecikme servo motoru sürmek için yeterli olacaktır.

O halde ADC den gelen bilgi 0 olduğunda 0.5ms 'lik, 255 olduğunda ise 2.5ms 'lik modülasyon genişliğini verecek bir formül oluşturalım. Formülümüz aşağıdaki gibi olacaktır.

$$SERVO_? = ((ADC_BILGI_?/2)*4)+125 \quad (3.1)$$

Adc 'den 255 bilgisinin (ADC_BILGI_?=255) geldiğini düşünürsek

$$\left(\left(\frac{255}{2} \right) * 4 + 125 \right) * 4 = 2540\mu s \cong 2500\mu s \quad (3.2)$$

elde edilir ve bu SERVO_? Değişkenine atanır.

Adc 'den 0 bilgisinin (ADC_BILGI_?=0) geldiğini düşünürsek

$$\left(\left(\frac{0}{2} \right) * 4 + 125 \right) * 4 = 500\mu s \quad (3.3)$$

elde edilir ve bu SERVO_? Değişkenine atanır. (Komut saykılı $4\mu s$)

Formül 3.2 'de sonuç 2540 olmasına rağmen gecikmelerden dolayı $40\mu s$ 'lik fazla önemsenmemiştir.

Formül 3.1 'e göre ADC 'den gelen 0–255 arasındaki herhangi bir değer için doğru orantılı olarak 0.5–2.5ms arasında bir modülasyon genişliği sağlanır. Bu çalışmada kullandığımız osilatör 10MHz 'lidir. Eğer farklı bir frekansa sahip osilatör kullanılacaksa 3.1 nolu formül yeniden düzenlenmelidir.

Aşağıdaki tabloda farklı osilatör frekansları için kullanacağımız formüller verilmiştir. (? İşareti servo motor numarasını temsil eder)

Osilatör Frekansı	Bir Komut Saykılı	Kullanılacak Formül
1 MHz	$(10 * 4)/1 = 40 \mu s$	$SERVO_? = ((ADC_BILGI_?/20)*4)+12.5$
4 MHz	$(10 * 4)/4 = 10 \mu s$	$SERVO_? = ((ADC_BILGI_?/5)*4)+50$
10 MHz	$(10 * 4)/10 = 4 \mu s$	$SERVO_? = ((ADC_BILGI_?/2)*4)+125$
20 MHz	$(10 * 4)/20 = 2 \mu s$	$SERVO_? = ((ADC_BILGI_?/1)*4)+250$

Tablo 3.11 Çeşitli osilatör frekanslarında SERVO_? Değişenleri için kullanılacak formül.

Modülasyon genişliği için kullanılan 1. formülde elde edilen sonuç SERVO_? Değişkenine atanır. SERVO_? Değişkeninin PULSOUT komutuyla PORTB ‘nin ilgili uçlarından çıkış olarak üretilmesi aşağıdaki komut satırıyla sağlanır.

PULSOUT PORTB.?,SERVO_? ; burada SERVO_? Değişkenindeki değer in Pulsout komutunun Saykıl süresiyle (bu çalışmada 4µs) çarpımı kadar PORTB ‘nin ?. Ucundan Lojik-1 çıkışı üretilir.

Örneğin, *PULSOUT PORTB.0,SERVO_0* komut satırında ADC ‘den SERVO_0 değişkenine 400 değeri geliyorsa PORTB ‘nin 0. ucundan (400*4µs) 1600 µs Logic-1 çıkışı üretilir ve bu süre sonunda tekrar PortB ‘nin 0. ucu Logic-0 olur.

Motor milinin konumunu koruması veya başka bir açı değeri alabilmesi için PortB ‘nin 0, 1, 2, 3 ve 4 nolu pinlerinden elde edilen modülasyon genişliğinin periyodik olarak devam etmesi gerekir. Servo motorların kontrol girişine uygulanan sinyalin frekansı 50Hz olmalıdır. Böylece uygulanan sinyalin bir periyodu $T = 1/50 = 0,02s = 20ms$ olması gerekir. Mikrodenetleyici yazılımında pulsout komutuyla Portb ‘nin ilgili pinlerinden logic-1 çıkışı üretildikten sonra PAUSE komutuyla bekleme alınarak çıkış sinyallerinin periyodik olarak tekrar etmesi sağlanır. Pause komutunun kullanımı aşağıdaki gibidir.

PAUSE bekleme_süresi; Burada bekleme süresi değeri 1ms ile çarpılarak elde edilen süre kadar bekleme yapar, süre sonunda işlemlere devam edilir.

SERVO_0, SERVO_1, SERVO_2, SERVO_3 ve SERVO_4 modülasyonları maksimum değerler aldığı anda toplamı 1275 değerini alır, buda $1275 * 4 \mu s \cong 5ms$ ‘lik bir

süredir. SERVO_0, SERVO_1, SERVO_2 ve SERVO_3 ve SERVO_4 deęişkenlerinin anlık deęerlerinin toplamını 4 μ s ile çarpıp 1000 'e bölersek servolara uygulanan toplam modülasyon genişliğini ms cinsinden elde ederiz. 20ms 'lik bir periyod oluşturabilmek için PAUSE komutunun bekleme süresine PER deęişkeni yüklenir.

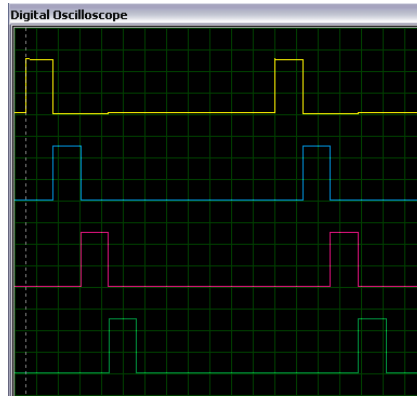
PER deęişkeni Periyod süresinden (20ms) Toplam Modülasyon Genişliği süresi çıkartılarak bulunur.

20ms 'lik periyod elde edilebilmesi için yazılımımızda aşağıdaki komut satırı kullanılır.

$$PER = (20 - (((SERVO_0 + SERVO_1 + SERVO_2 + SERVO_3 + SERVO_4) * 4 / 1000))$$

$$PER = (20 - (((SERVO_0 + SERVO_1 + SERVO_2 + SERVO_3 + SERVO_4) / 250)) \quad (3.4)$$

PAUSE PER ; Böylece Servo motorların kontrol girişlerine uygulanan sinyallerin periyodu 20ms 'de sabit tutulmuş olur.



Şekil 3.79 ISIS simülasyon programında Servo motorların kontrol sinyallerinin osiloskop görüntüsü

16F877/A mikrodenetleyicisine yüklenecek olan programın PicBasicPro Diliyle hazırlanmış halinin tamamı aşağıdaki gibidir.[11]

```
TRISA=%00011111
```

```
TRISB=0
```

```
TRISC=0
```

```
TRISD=0
```

```
'-----
```

```
@ DEVICE pic16F877
```

```
@ DEVICE pic16F877, WDT_on
```

```
@ DEVICE pic16F877, PWRT_ON
```

```
@ DEVICE pic16F877, PROTECT_OFF
```

```
@ DEVICE pic16F877, HS_OSC
```

```
'-----
```

```
DEFINE    ADC_BITS    8
```

```
DEFINE    ADC_CLOCK   3
```

```
DEFINE    ADC_SAMPLEUS 100
```

```
DEFINE    OSC 10
```

```
'-----
```

```
ADCON1=%00000000
```

```
'-----
```

```
ADC_BILGI_0 VAR WORD
```

ADC_BILGI_1 VAR WORD

ADC_BILGI_2 VAR WORD

ADC_BILGI_3 VAR WORD

ADC_BILGI_4 VAR WORD

SERVO_0 VAR WORD

SERVO_1 VAR WORD

SERVO_2 VAR WORD

SERVO_3 VAR WORD

SERVO_4 VAR WORD

PER VAR WORD

BASLA:

ADCIN 0,ADC_BILGI_0

ADCIN 1,ADC_BILGI_1

ADCIN 2,ADC_BILGI_2

ADCIN 3,ADC_BILGI_3

ADCIN 4,ADC_BILGI_4

BAK: IF ADCON0.2=1 THEN BAK

ADC_BILGI_0=ADC_BILGI_0+1

ADC_BILGI_1=ADC_BILGI_1+1

ADC_BILGI_2=ADC_BILGI_2+1

ADC_BILGI_3=ADC_BILGI_3+1

SERVO_0=((ADC_BILGI_0/2)*4)+125

SERVO_1=((ADC_BILGI_1/2)*4)+125

SERVO_2=((ADC_BILGI_2/2)*4)+125

SERVO_3=((ADC_BILGI_3/2)*4)+125

SERVO_4=((ADC_BILGI_4/2)*4)+125

PER=(20-((SERVO_0+SERVO_1+SERVO_2+SERVO_3+SERVO_4)/250))

PULSOUT PORTB.0,SERVO_0

PULSOUT PORTB.1,SERVO_1

PULSOUT PORTB.2,SERVO_2

PULSOUT PORTB.3,SERVO_3

PULSOUT PORTB.4,SERVO_4

LOW PORTB.0

LOW PORTB.1

LOW PORTB.2

LOW PORTB.3

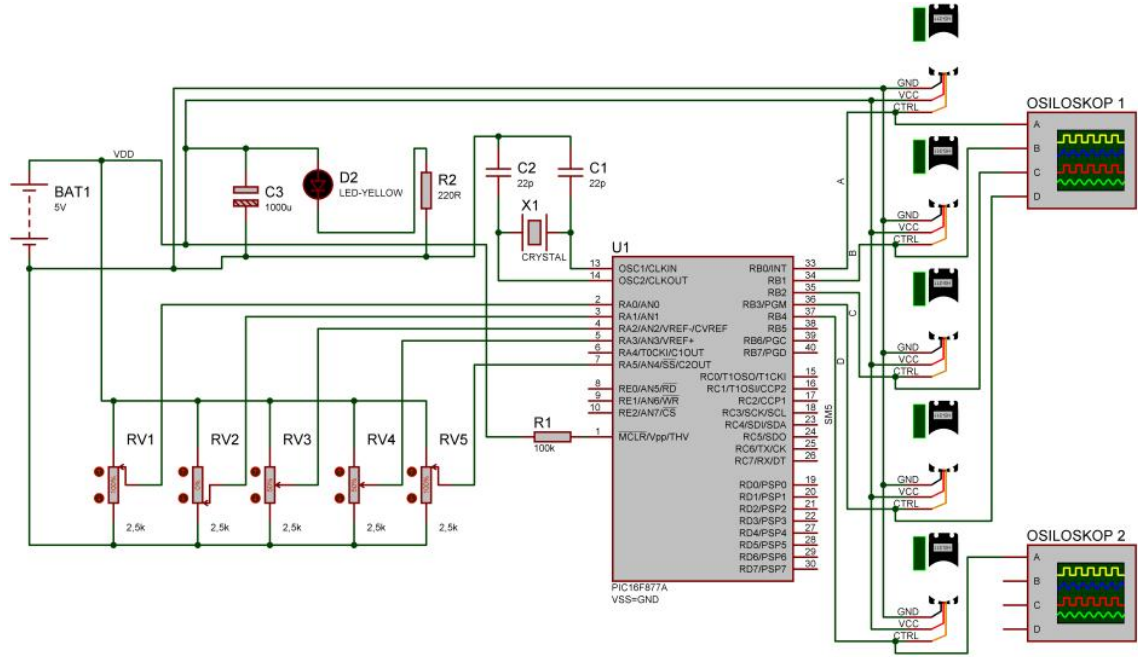
LOW PORTB.4

PAUSE PER

GOTO BASLA

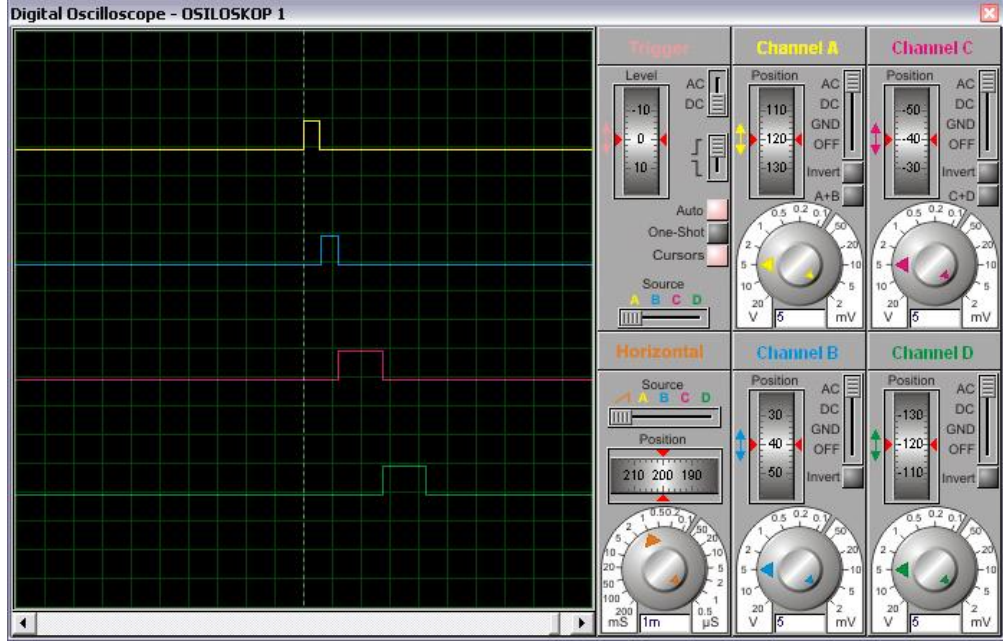
3.3.1.3.4 Kontrol Kartının Bilgisayar Ortamında Test Edilmesi

ISIS proteus programı ile hazırlanan servo motor sürücü kartı devresinin ve mikrodenetleyici için hazırlanan programın eksiksiz ve doğru çalıştığını bilgisayar ortamında sanal ölçme cihazlarıyla test edilmesi gerekir. Bu amaçla yine Proteus ISIS programı kullanılır. Proteus ISIS ile Şekil 3.80 'deki Devre şeması hazırlandıktan sonra mikrodenetleyiciye ait kod ISIS programında sanal olarak mikrodenetleyiciye yüklenir.

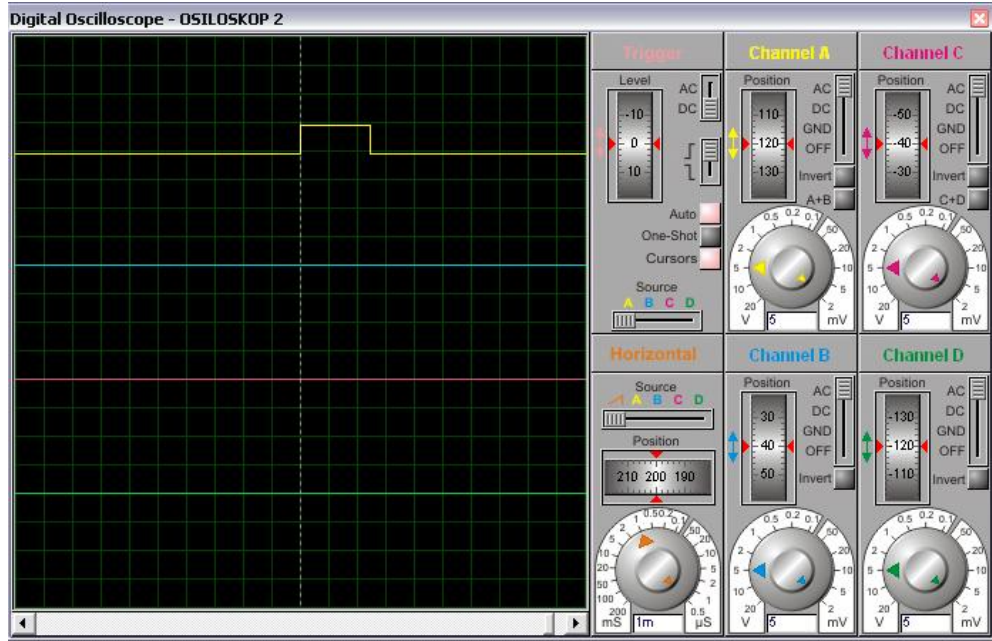


Şekil 3.80 Devre şeması

Daha sonra analog girişleri sağlayacak olan potansiyometreler ile servo motor kontrol sinyalini görüntülemek için Sanal Osiloskop cihazı eklenir. Simülasyon başlatıldığında potansiyometrelerde üretilen analog sinyaller sanal voltmetrelerde 1, 2, 3 ve 4 nolu servo motor kontrol çıkışlarından üretilen sinyaller Şekil 3.81 'de ve 5 nolu servo motor kontrol çıkışı ise Şekil 3.82 'deki sanal osiloskopta görüntülenir.



Şekil 3.81 Osiloskop 1



Şekil 3.82 Osiloskop 2

Böylece devre fiziksel olarak hazırlanmadan daha tasarım aşamasında eksiklikler ve hatalar belirlenerek eksiksiz ve tam doğru çalışan bir devre hazırlanmış olur.

3.3.1.4 Analog-Sayısal Dönüştürücü Veri Giriş ve Çıkış Kartı

Master kolun eklemlerindeki potansiyometrelerden gelen analog bilgiyi sayısal bilgi formatında bilgisayar ortamına taşımak, bilgisayarda işlenen sayısal bilginin analog bilgi olarak servo motor sürücü kartına göndermek amacıyla JAC AIO3320 Analog-Sayısal Dönüştürücü (Veri Toplama) Veri Giriş-Çıkış kartı kullanılmıştır.



Şekil 3.83 JAC AIO3320 Kart ve ekipmanları

3.3.1.4.1 JAC AIO3320 ADC Kartın Özellikleri

PCI tabanlı tak çalıştır fonksiyonu ile 16 adet kart tanımlanabilir. Analog giriş ve çıkış kanalları için 37 pinli ray montajlı JS51026 bağlantı terminali, sayısal giriş ve çıkış kanalları için ADP9201 ray montajlı bağlantı terminali kullanılmıştır.

- Analog Giriş Fonksiyonu:

İzole edilmiş 8 kanallı 12 bit çözünürlüklü analog giriş ve bu girişlerin gerilim aralığı yazılımla değiştirilebilir. Dönüştürme hızı kanal başına 13 μ S 'dir.[12]

Analog Giriş Gerilim Aralığı

Mode 0: 0 ~ +5V

Mode 1: -5 ~ +5V

Mode 2: 0 ~ +10V

Mode 3: -10 ~ +10V

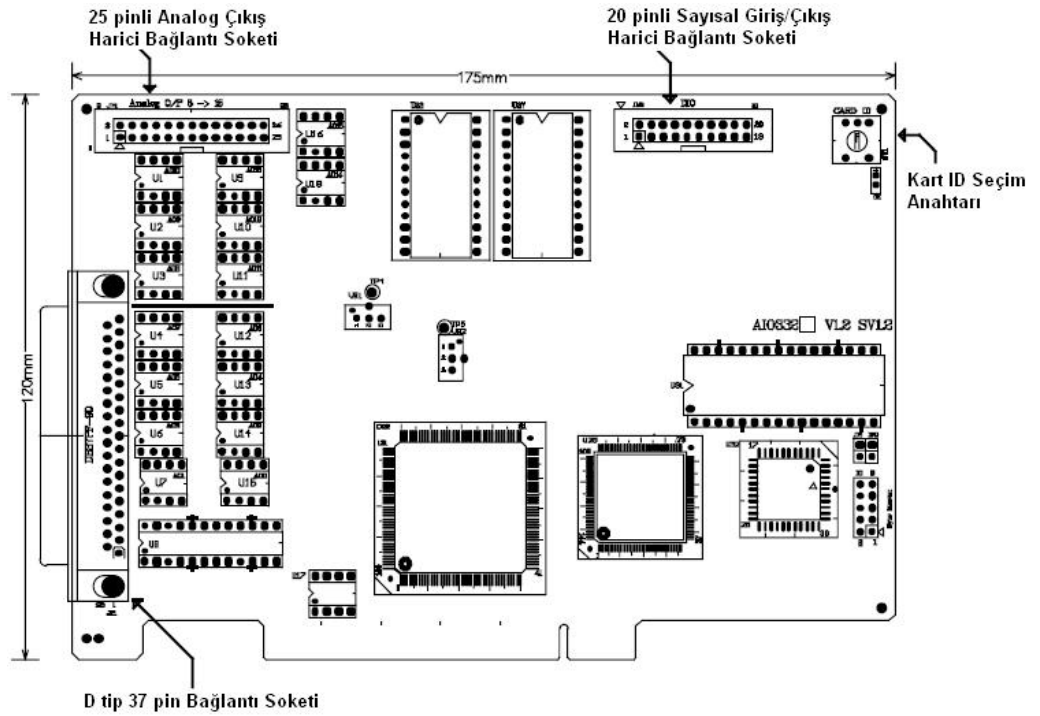
- Analog Çıkış Fonksiyonu:

İzole edilmiş 8 kanallı 16 bit çözünürlüklü analog çıkış. Otomatik tarama oranı 2.1kHz 'dir.

- Sayısal Giriş-Çıkış Fonksiyonu:

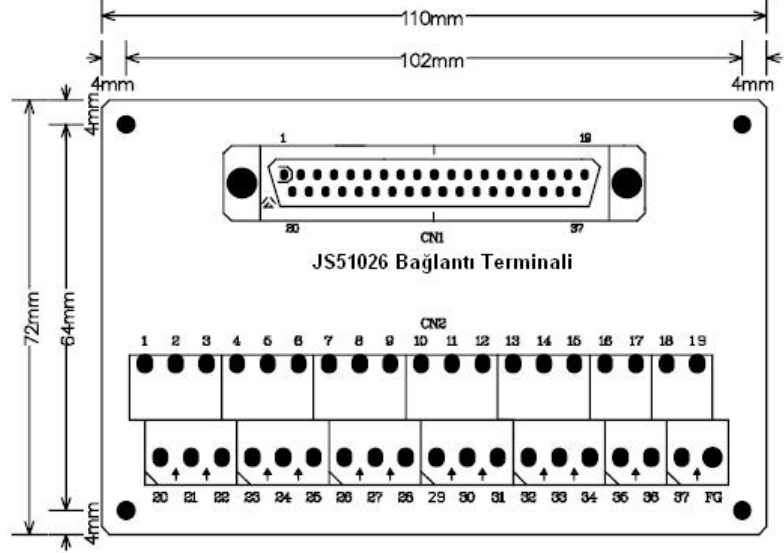
8 adet izole edilmiş Sayısal giriş ve çıkış. Sayısal Giriş ve Çıkış gerilim aralığı 0~30Volt ve çıkış akımı maksimum 500 mA 'dir.

AIO 3320 kartın fiziksel ölçüleri ve Blokları



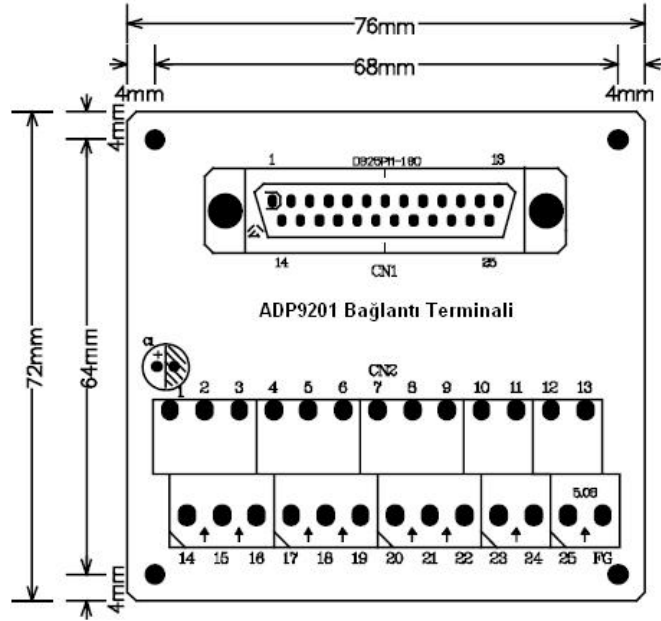
Şekil 3.84 AIO 3320 kartın fiziksel ölçüleri ve Blokları[12]

JS51026 Bağlantı Terminali



Şekil 3.85 JS51026 Bağlantı Terminali[12]

ADP9201 Bağlantı Terminali



Şekil 3.86 ADP9201 Bağlantı Terminali[12]

3.3.1.4.2 AIO3320 için JS51026 Analog Giriş ve Çıkış Bağlantı Terminali

JS51026 bağlantı terminali endüstriyel amaçlı ray montajlı, AIO3320 kartının Analog giriş ve çıkış kanal bağlantılarının yapılabildiği bir terminaldir. 37 adet bağlantı noktası bulunmaktadır. 8 adet Analog Girişler AI0~AI7 portları, 8 adet analog çıkışlar ise DA0~DA7 portları ile tanımlanmıştır. Her kanalın toprak yani ortak uçları ise AGND ile tanımlanmıştır. Ayrıca bağlantı terminali üzerinde +5V 'luk 2 adet çıkış ve herhangi bağlantısı olmayan 4 adet port bulunmaktadır. Bağlantı terminali JS51026 ile Veri toplama kartı AIO3320 arasındaki bağlantı Ekranlanmış (topraklanmış) 37 damarlı dişi-erkek sokete sahip DB37PF kablosu ile yapılmaktadır.

Portların bağlantı terminali üzerindeki numaralarına göre isimleri aşağıdaki tabloda belirtilmiştir. [12]

Pin	Açıklama			Pin	Açıklama
1	AI0: 0. Analog Giriş	AI0	1	20	AGND
2	AI1: 1. Analog Giriş	AI1	2	21	AGND
3	AI2: 2. Analog Giriş	AI2	3	22	AGND
4	AI3: 3. Analog Giriş	AI3	4	23	AGND
5	AI4: 4. Analog Giriş	AI4	5	24	AGND
6	AI5: 5. Analog Giriş	AI5	6	25	AGND
7	AI6: 6. Analog Giriş	AI6	7	26	AGND
8	AI7: 7. Analog Giriş	AI7	8	27	AGND
9	DA0: 0. Analog Çıkış	DA0	9	28	AGND
10	DA1: 1. Analog Çıkış	DA1	10	29	AGND
11	DA2: 0. Analog Çıkış	DA2	11	30	AGND
12	DA3: 3. Analog Çıkış	DA3	12	31	AGND
13	DA4: 4. Analog Çıkış	DA4	13	32	AGND
14	DA5: 5. Analog Çıkış	DA5	14	33	AGND
15	DA6: 6. Analog Çıkış	DA6	15	34	AGND
16	DA7: 7. Analog Çıkış	DA7	16	35	+5Vda
17	+5Vda: 5V Çıkış	+5Vda	17	36	N.C.
18	Bağlantı Yok	N.C.	18	37	N.C.
19	Bağlantı Yok	N.C.	19		

Tablo 3.12 JS51026 bağlantı terminali pin tablosu



Şekil 3.87 JS51026 bağlantı terminali



Şekil 3.88 DB37PF kablosu

3.3.1.4.3 AIO3320 için ADP9201 Analog Giriş ve Çıkış Bağlantı Terminali

ADP9201 bağlantı terminali endüstriyel amaçlı ray montajlı, AIO3320 kartının Sayısal Giriş ve Çıkış kanal bağlantılarının yapılabildiği bir terminaldir. 30 Adet bağlantı noktası bulunmaktadır. 8 bitlik Sayısal Girişler IN0~IN7 portları, 8 bitlik Sayısal Çıkışlar ise OUT0~OUT7 portları ile tanımlanmıştır. Bunların haricinde Sayısal girişleri aktif yapan 0V portları ile Sayısal çıkışların Ortak ucu olan COM1, COM2, COM3 ve COM4 portları bulunmaktadır. Ayrıca bağlantı noktasının +24V 'luk harici beslemesi için kullanılan Vin ve 0V portları bulunmaktadır. ADP9201 bağlantı terminali ile AIO3320 kartı arasındaki bağlantı 20 pinli yassı kablo kullanılmaktadır.

Portların bağlantı terminali üzerindeki numaralarına göre isimleri aşağıdaki tabloda belirtilmiştir. [12]

Pin	Açıklama		Pin	Açıklama
1	EXT_IN0: 0. Giriş		2	EXT_OUT0: 0. Çıkış
3	EXT_IN1: 1. Giriş		4	EXT_OUT1: 1. Çıkış
5	EXT_IN2: 2. Giriş		6	EXT_OUT2: 2. Çıkış
7	EXT_IN3: 3. Giriş		8	EXT_OUT3: 3. Çıkış
9	EXT_IN4: 4. Giriş		10	EXT_OUT4: 4. Çıkış
11	EXT_IN5: 5. Giriş		12	EXT_OUT5: 5. Çıkış
13	EXT_IN6: 6. Giriş		14	EXT_OUT6: 6. Çıkış
15	EXT_IN7: 7. Giriş		16	EXT_OUT7: 7. Çıkış
17	DGND: Sayısal Toprak		18	DGND: Sayısal Toprak
19	+24Ve: Harici Güç Kaynağı Girişi		20	+24Ve: Harici Güç Kaynağı Girişi

Tablo 3.13 ADP9201 bağlantı terminali pin tablosu



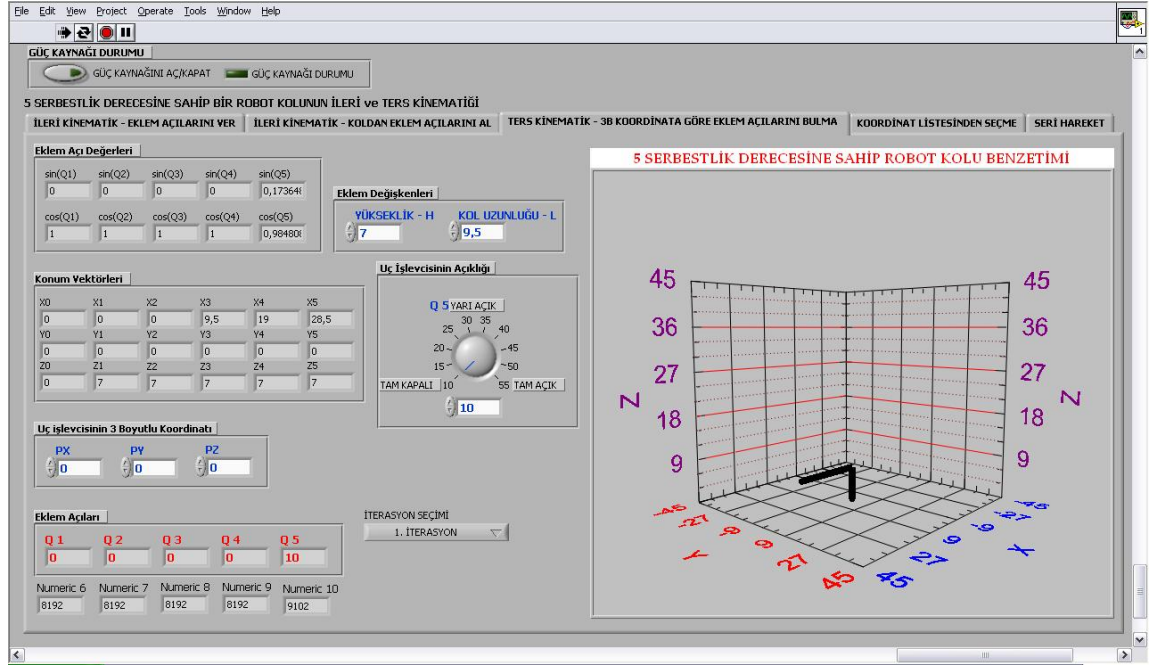
Şekil 3.89 ADP9201 bağlantı terminali



Şekil 3.90 ADP9201 20 pinli kablo

3.3.2 Projede Kullanılan Kontrol Yazılımı

Bu çalışmada AIO3320 veri toplama kartı yardımıyla master kolun eklemlerinden alınan analog bilginin sayısal bilgiye dönüştürülüp çeşitli işlemlerden geçirilerek Slave kolun eklemlerindeki servo motorların kontrolünün yapılabilmesi için kullanıcı arayüzlü bir yazılım hazırlanmıştır. Bu yazılım Grafikselleştirme Programlama Dili olan LabVIEW ile hazırlanmıştır. Eklem açılarına göre her eklem için konum vektörleri ile uç işlevcinin konum vektörü yani ileri kinematik hesabı yazılım altında çalıştırılan MATLAB scripti ile bulunmaktadır. Uç işlevcinin verilen koordinatlarına göre eklem açılarının olması gereken değeri yani ters kinematik hesabı yine program altında çalışan MATLAB scripti ile bulunmaktadır. Yazılımın kullanıcı tarafından beş farklı şekilde kullanım imkanı vardır. Bunlardan birincisi master koldan alınan analog bilgi bilgisayarda işlenerek slave kol kontrol edilir, ikincisi yazılımdan verilen eklem açılarına göre slave kol kontrol edilir, üçüncüsü ise uç işlevcinin koordinatları x,y,z şeklinde verilerek slave kol kontrol edilir dördüncüsü ise uç işlevcinin koordinatları bir koordinat listesinden seçilerek slave kol kontrol edilir. Ve sonuncu olarak slave kol önceden belirlenmiş eklem açılarına göre belirlenen periyotla seri hareketler yapar. Eklem açılarına göre slave kolun grafiği yazılımda anlık olarak gösterilmektedir. Kontrol programına ait kullanıcı arayüzü Şekil 3.91 'da görülmektedir.



Şekil 3.91 Kontrol programına ait kullanıcı arayüzü

3.3.2.1 LabVIEW Grafiksel Programlama Dili

1976 yılında ABD 'nin Texas eyaletinde Austin şehrinde kurulan National Instruments firması ilk çalışmalarına GPIB (General Purpose Interface Bus-IEEE 488) kartlar ile başlamıştır. Daha sonraki yıllarda MAC makineler için DAQ (Data Acquisition/Veri Toplama) kartı üreten şirket 1983 yılında programlama için gereken zamanı azaltacak yollar aramaya başladılar. Nisan 1985 tarihinde Ar-Ge sorumlusu olan Jack Mc, Jeff Kodosky ve sirkelin CEO 'su olan Dr. Jim Truchard liderliğinde ve Texas Üniversitesi kapsamındaki 11 kişilik ekip tarafından, sezgisel kullanıcı arayüzünü içeren ön paneli, yenilikçi blok diyagramı programlama yöntem bilimi ile birleştirilerek verimli, yazılım tabanlı bir grafiksel ölçme sistemi olan LabVIEW sanal enstrümantasyon (ölçme) kavramı geliştirilmiştir.

Bu çalışmalar sonrasında ilk prototip LabVIEW adı altında Macintosh üzerinde yaratıldı. 512KB 'lık hafıza aşıldığında ekip projeyi durdurmak zorunda kaldı. Ocak 1986 yılında Macintosh Plus (1MB) piyasaya çıkması ile proje daha fazla

geliştirilebilme imkânı buldu. DOS tabanlı PC'lerin 620KB engeli yüzünden LabVIEW prototipi yalnızca Macintosh uyumlu olarak kaldı. 1986 'da Macintosh ölçme uygulamalarında sık kullanılmadığı halde diğer işletim sistemleri LabVIEW teknolojisini destekleyene kadar Macintosh teknolojisi için LabVIEW sürüm 1 adı altında en uygun seçenek oldu. Ocak 1990 'da LabVIEW nesneye dayalı programlama teknikleri ile en baştan tekrar tasarlandı. Önemli olarak LabVIEW 2 'nin derleyicisi VI 'ların yürütme hızını C programlama dilinde yaratılmış programlarla kıyaslanabilir bir düzeye getirdi. Yeni MacOS gibi grafiksel işletim sistemlerinin piyasaya çıkması ile National Instruments LabVIEW teknoloji'sini Eylül 1992 'de LabVIEW 2.5 adı altında PC 'ler ve diğer iş istasyonu platformlarına taşıdı. "LabVIEW for Windows" ve "LabVIEW for Sun" tabanlı yeni taşınabilir mimariler tanıtıldı.

Temmuz 1993 de LabVIEW 3 sürümü, başka platformlarda da yürütülebilir özellikte hazırlandı. Nisan 1996 'da LabVIEW' sürüm 4 tanıtıldı ve bu kullanıcılara çok kolay kullanım özellikleri getirildi. Kontroller, fonksiyonlar ve araçlar, isteğe bağlı kayan panolara yerleştirildi. Menüler tekrar organize edildi ve ayrıntılı çevrim içi yardım eklendi. Mart 1998 'de sistemin bir programın ayrı ayrı parçalarını aynı anda çalıştırabilme, geri alma ve bağlanırlılık özellikleriyle LabVIEW 5.0.1 piyasaya çıktı.

2002 başlarında kontrol uygulamaları, test ve görüntüleme için grafiksel geliştirme ortamı sağlayan LabVIEW 6'nın piyasaya çıkarıldığı duyuruldu. 2003 başlarında Express VI özelliği eklenmiş olan LabVIEW 7 Express piyasaya sunuldu. 38 adet express VI içeren "LabVIEW Express" ile I/O, analiz ve sunum ihtiyaçları için LabVIEW blok diyagramları geliştirilmek çok daha kolay hale getirildi. Bu yolla basit bir veri toplama sisteminden ileri düzey tümleşik yazılım geliştirme işlemi için kolay üretkenlik sağlanır hale gelindi.

NI, LabVIEW 7 Express 'den sonra 2004 yılında genişletilmiş DAQ asistan ve Real-time NI-DAQmx içeren LabVIEW 7.1 'i LabVIEW kullanıcılarının hizmetine sundu. LabVIEW 7.1 'de hiç programlama bilgisi gerektirmeyen "easy-to-configure" dialogları kullanarak yüksek hızlı dijital I/O kartları için analog ve dijital uyarıcı sinyaller kolay ve hızlı bir şekilde oluşturulabilir ve değiştirilebilir. Bu sürüm en son network protokolleri ile daha fazla cihaza ve akıllı sensörlere bağlanma imkanı veren yeni fonksiyonları ve zamanlanmış döngüleri içermektedir.

2005 sonlarına doğru yeniden düzenlenmiş açılış arayüzü, menüleri ve paletleri ile LabVIEW 8.0 kullanıma sunuldu. Bu sürümünde kaydetme, değişkenlerin kullanımı, proje ve uygulama geliştirme, web uygulamaları, güvenlik, VI ve fonksiyonlar gibi birçok konuda LabVIEW 'in yetenekleri arttırıldı. 2006 'da LabVIEW 8.2 sürümüne nesne yönelimli programlama, kullanıcı arayüzü geliştirme, kod ile DLL 'lere bağlanabilme, .NET web uygulama servisleri, yüksek hızlı dosya g/ç, MathWorks firmasının MATLAB dil yazım kuralları ile M-File yazabilme için yeni araçlar eklendi. LabVIEW 8.2.1 sürümü ile Windows Vista işletim sistemi için hazır hale getirildi.

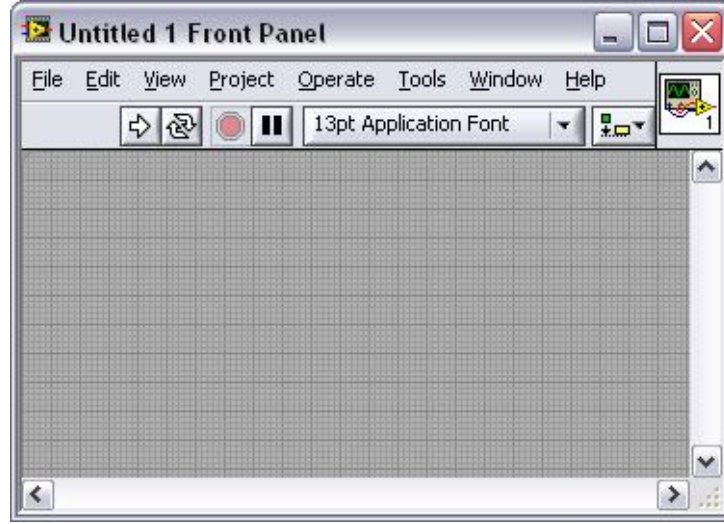
2007 ortalarında LabVIEW 8.5 sürümünün dağıtımına başladı. Bu son sürümde LabVIEW proje klasörlerinden fiziksel klasörlere bağlantı ve diskteki uygulamaları yönetmek için yeni araçlar eklendi. Test, kontrol ve tasarım problemlerinin çözümü için yeni araçlarla "LabVIEW Statechart Modüle" geliştirildi. Çok çekirdekli yapıyı destekleyerek performans arttırıldı ve binlerce endüstriyel aygıt desteklendi.

LabVIEW 20 yılı aşkın bir süredir gelişmekte olmasına rağmen, teknolojiadaki sınırlamalardan dolayı dünyada ve ülkemizde beklenen hızda yaygınlaşmamıştır. 90 'lı yılların sonuna doğru teknolojik gelişmeler LabVIEW 'i gerek GPL (grafiksel programlama dili) olarak gerekse ölçme ve enstrümantasyon alanında yeterli seviyeye getirmiştir. Programlamaya getirdiği kolaylıklarla birlikte veri toplama için sağladığı tümleşik devreler sayesinde, çok hassas ölçümler, pratik bir şekilde gerçekleştirilebilmektedir. 2000li yılların başından itibaren de hem endüstride üretimin çeşitli alanlarında hem de üniversitelerde akademik seviyede hızla yaygınlaşmaya başlamıştır. Bugün ülkemizde birçok üniversitede grafiksel programlama yapısı ve özellikle LabVIEW lisans seviyesinde seçmeli ders olarak okutulmaktadır.[13]

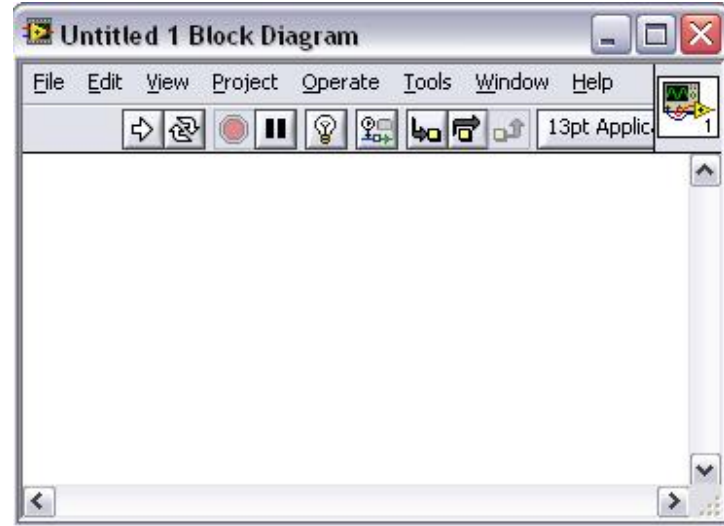
3.3.2.1.1 Ön Panel ve Blok Diyagramı

LabVIEW grafiksel programlama dilinin Kullanıcı ara yüzü Ön Panel (Front Panel) ve Blok Diyagramı (Block Diagram) olmak üzere iki ana kısımdan oluşmaktadır. Ön Panel veya Front Panel olarak adlandırılan arayüz kullanıcının programı doğrudan

doğruya kullandığı her türlü veri girişinin yapılabildiği ve veri çıkışının izlenebildiği çeşitli özelliklere sahip Giriş ve Gösterge Formlarının bulunduğu kısımdır. Şekil 3.92 Blok diyagram ise ön panelden ve/veya donanımsal veri kaynağından alından verilerin çeşitli fonksiyonlarla işlenip çıkış verilerinin yine ön paneldeki göstergelere ve/veya donanımsal veri kaynaklarına gönderildiği veri işleme fonksiyonların bulunduğu, kullanıcının program çalışırken göremediği kısımdır. Şekil 3.93



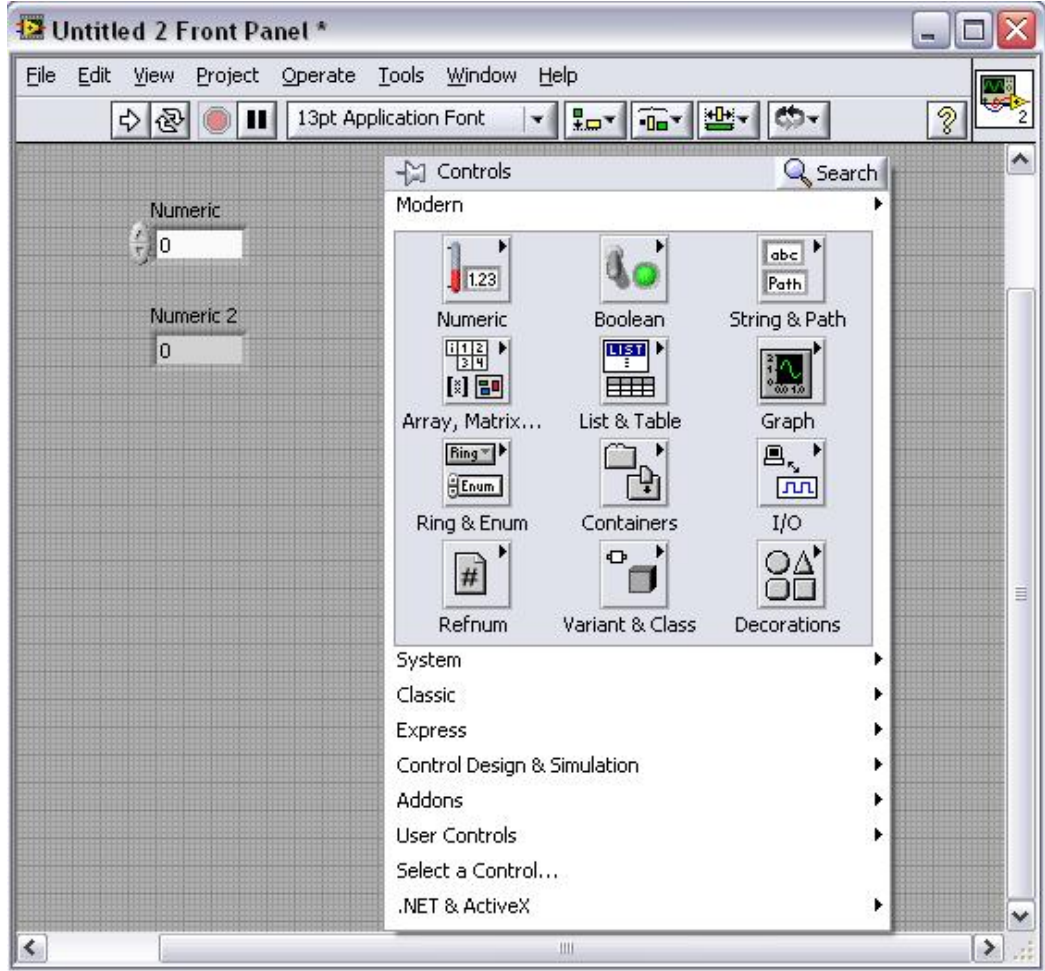
Şekil 3.92 Ön panel



Şekil 3.93 Blok diyagram

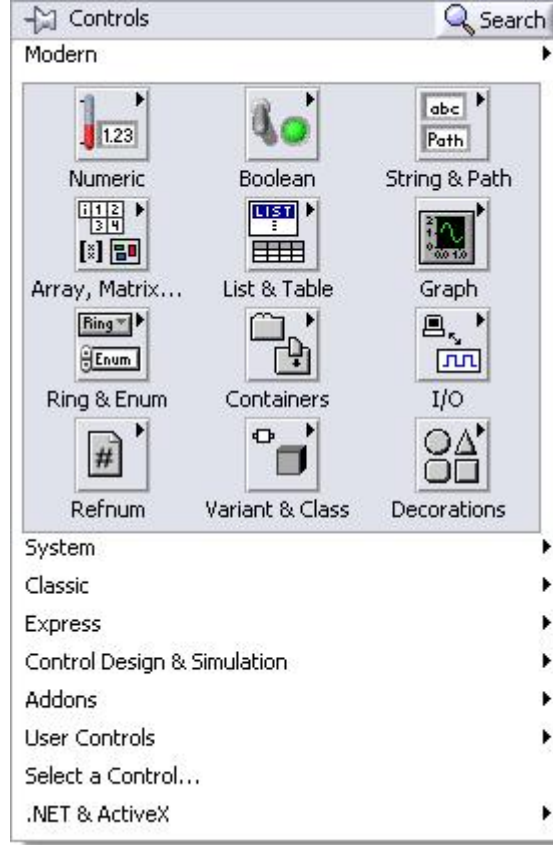
Ön panel veya orijinal adıyla Front panel program arayüzünün hazırlandığı, programın özelliğine göre üzerine Veri Giriş veya Çıkış nesnelerinin eklenmesinden

oluşmaktadır. Ön panelde Veri Girişinin yapıldığı Nesnelere Kontrol (Control), Veri çıkışının yapıldığı Nesnelere ise Gösterge (Indicator) adı verilir. Şekil 3.94



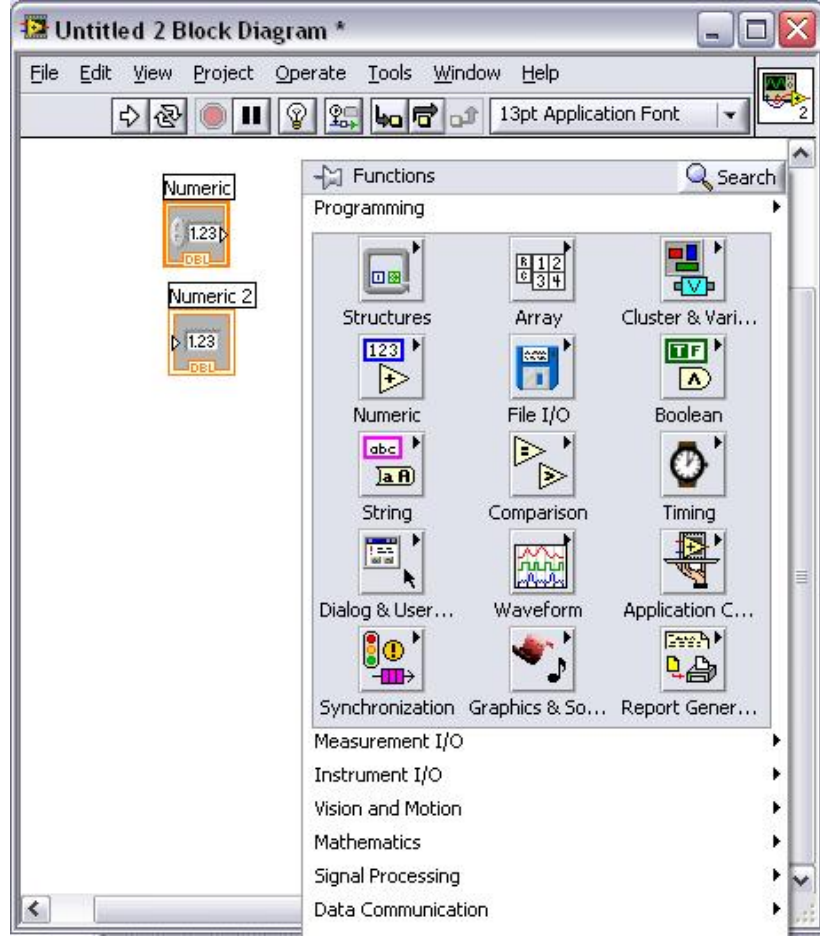
Şekil 3.94 Kontrol ve Gösterge nesnesi

Kontrol ve Gösterge Nesneleri Ön Panel 'de Kontrol Paleti içinde yer alır. Kontrol paletinde çeşitli veri tiplerine göre Kontrol ve Gösterge Nesneleri bulunmaktadır. Gerekli olan nesnelere Kontrol paletindeki Kategorilerden seçilerek ön panele eklenir. Kontrol paletindeki Nesnelere özelliklerine ve görünüşlerine göre Modern, Klasik, Sistem, Ekspres ve eklenen modüllere göre değişik kategorilerden oluşmaktadır. Şekil 3.95



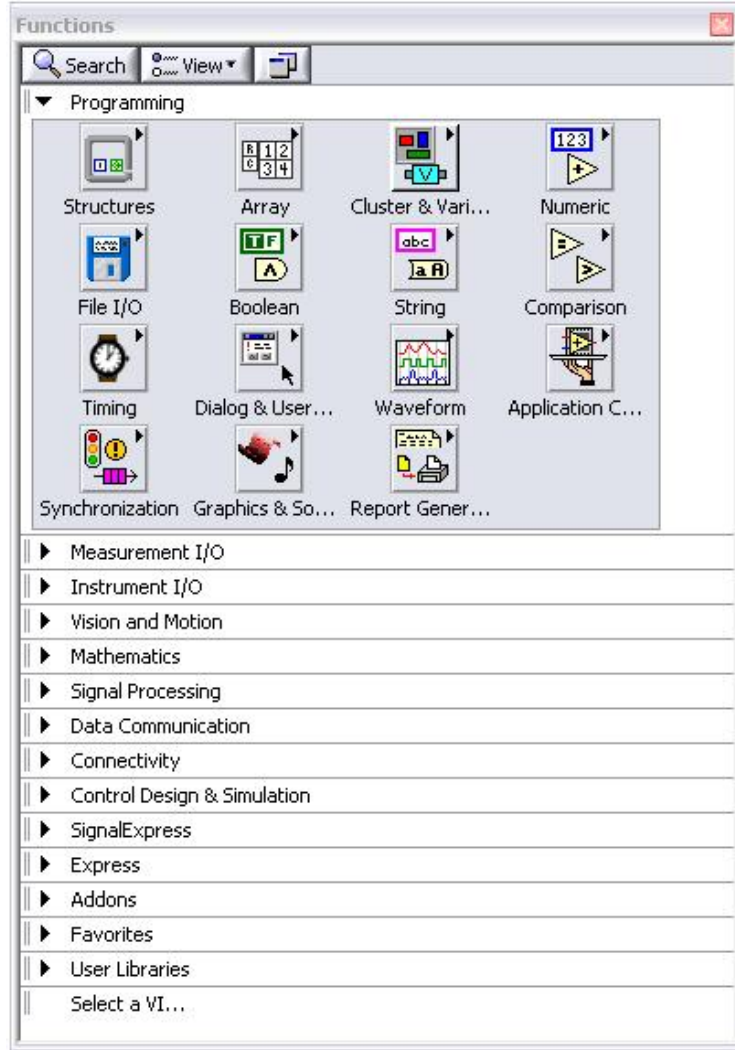
Şekil 3.95 Kontrol paleti

Blok diyagram diğer adıyla Block Diagram Ön panele eklenen Kontrol veya Gösterge nesnelerinin karşılığı olan Terminaller ve işlemleri gerçekleştiren çeşitli fonksiyon nesnelerinden oluşmaktadır. Terminaller ön panele eklenen nesnelerle bağlantılıdır. Terminaller nesnelere olduğu gibi kontrol ve gösterge terminalleri olarak isimlendirilir. Terminallerin sağında veya solunda sağ tarafı işaret eden bir ok bulunur. Bu ok terminalin sol kenarında ise Gösterge, Sağ kenarında ise Kontrol nesnesidir. Şekil 3.96



Şekil 3.96 Kontrol ve Gösterge terminalleri

Fonksiyonlar ise LabVIEW 'in temel işlem elemanlarıdır. Açık Sarı zemin üzerine siyah font rengine sahip nesnelere sahiptir. Fonksiyonlar ön panel veya blok diyagrama sahip olmamakla birlikte bağlantı noktalarına sahiptirler. Fonksiyonlar Blok Diyagramındaki Fonksiyon Paletinde Programming, Instruments I/O, Mathematics, Connectivity, ve SignalExpress gibi yapmış olduğu işlemlere göre kategorilere ayrılmış olarak bulunur. Şekil 3.97



Şekil 3.97 Fonksiyon paleti

Blok diyagramında ön paneldeki Kontrol, Gösterge terminalleri ve fonksiyonların dışında çeşitli işlemleri yerine getiren yapılar bulunmaktadır.[14]

SubVIs: Diğer VI 'ların blok diyagramında kullanılan VI 'lardır. (Alt_Virtual Instrument)

Structures: İşlem sırasını yöneten mantık elemanlarıdır.

Formula Nodes: Blok diyagrama doğrudan formül girilebilmesi için kullanılan boyutlanabilen elemanlardır.

Expression Nodes: Tek değişkenli deyimlerin hesaplanmasında kullanılan yapılarıdır.

Property Nodes: Bir türün özelliklerinin ayarlanması veya bulunması için kullanılan yapılardır.

Invoke Nodes: Bir türün metotlarının hesaplanması için kullanılan yapılardır.

Code Interface Nodes (CINs): Metin tabanlı programlama dillerinden kod çağırarak için kullanılan yapılardır.

Call by Reference Nodes: Dinamik olarak yüklenen VI 'ları çağırarak için kullanılan yapılardır.

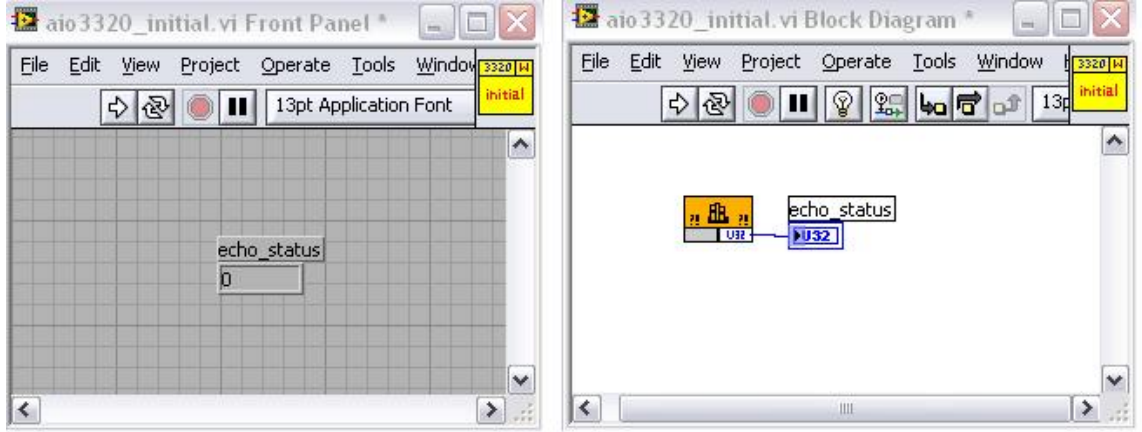
Call Library Nodes: Standart kütüphaneleri veya DLL çağırarak için kullanılan yapılardır.

3.3.2.2 Analog-Sayısal Veri Toplama Kartı Sürücülerini

JAC AIO3320 Analog-Sayısal veri toplama kartının LabVIEW programlama diliyle kontrol edilebilmesi için üretici firma tarafından Analog-Sayısal Dönüştürücü Bloğu, Sayısal-Analog Dönüştürücü Bloğu, Başlangıç Bloğu, Sayısal Giriş-Çıkış Bloğu ve Güvenlik Bloğu için sürücü adı verilen modüller bulunmaktadır.

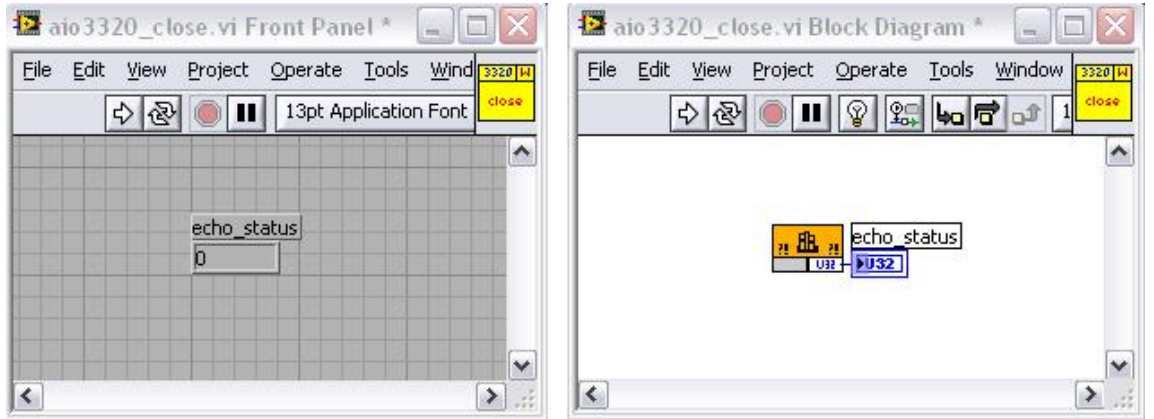
3.3.2.2.1 Veri Toplama Kartının LabVIEW ile Açılıp Kapatılması

AIO3320 kartının Veri toplama ve dönüşüm yapabilmesi için öncelikle inital.vi sürücüsünün kullanılması gerekir. LabVIEW programı içine inital.vi sürücüsünün blok diyagramındaki inital fonksiyonun eklenmesi gerekir, eğer inital fonksiyonu yazılıma eklenmezse kart çalışmayacağı için veri okuma ve yazma işlemi yapamaz. Aşağıda inital.vi sürücüsüne ait Ön panel ve Blok Diyagramı görülmektedir.



Şekil 3.98 inital.vi sürücüsüne ait Ön panel ve Blok Diyagramı[13]

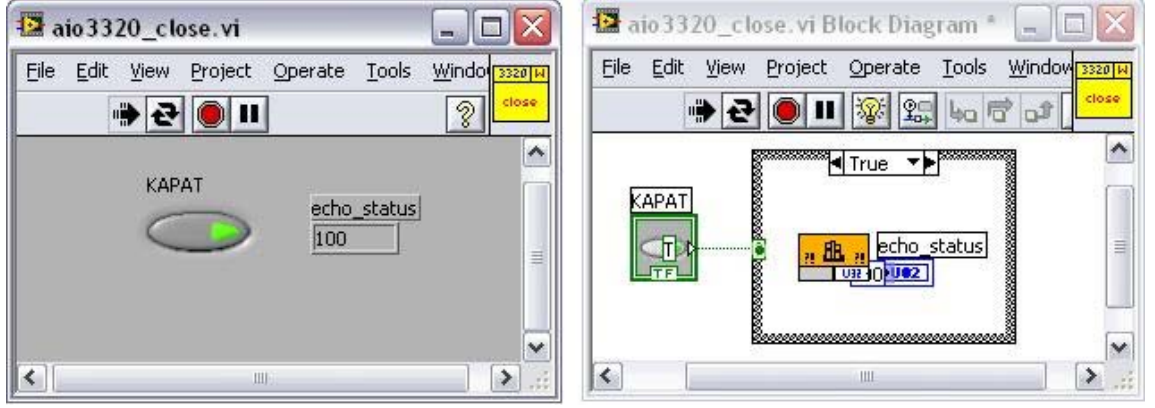
Kartımızın yazılım aracılığıyla kapatılması için close.vi sürücüsünün kullanılması gerekir. Close.vi sürücüsünün blok diyagramındaki close fonksiyonu doğrudan yazılım içine konulmamalı bir Karar yapısıyla birlikte çalıştırılmalıdır. Aksi halde close fonksiyonu doğrudan yazılım içine konulduğunda kart çalışmaya başlamadan kapanacaktır, buda kartın veri okuma ve yazma işlemi yapmasını engeller. Aşağıda close.vi sürücüsüne ait Ön panel ve Blok Diyagramı görülmektedir.



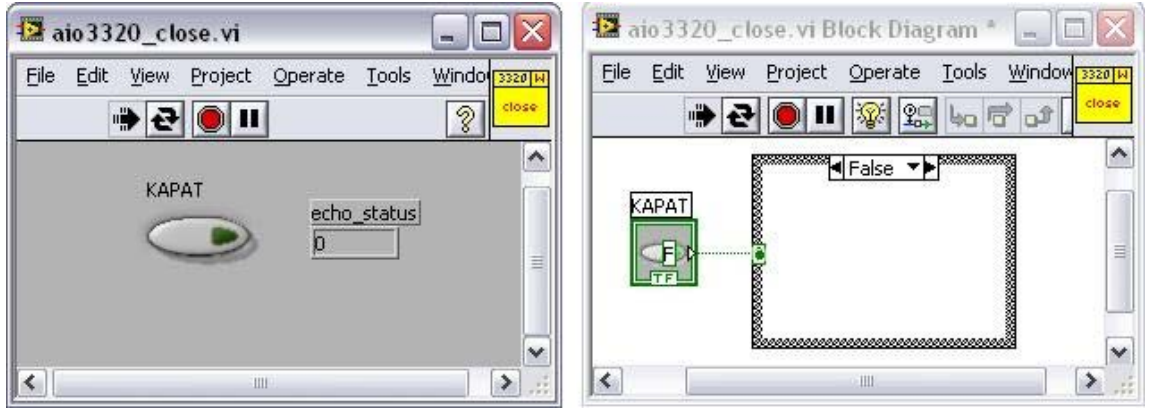
Şekil 3.99 close.vi sürücüsüne ait Ön panel ve Blok Diyagramı[13]

Burada da close fonksiyonunun bir karar yapısıyla kullanılması görülmektedir. Burada Kapat butonuna basılmadığı sürece Case Structure yapısına False sinyali (Şekil 3.101) gönderildiğinden kart kapanmayacaktır, Kapat butonuna basıldığında ise Case Structure[14] yapısına True sinyali (Şekil 3.100) gönderileceğinden kart kapanacaktır.

Aşağıda Kapat butonunun True ve False Durumlarına göre blok diyagramının durumu görülmektedir.

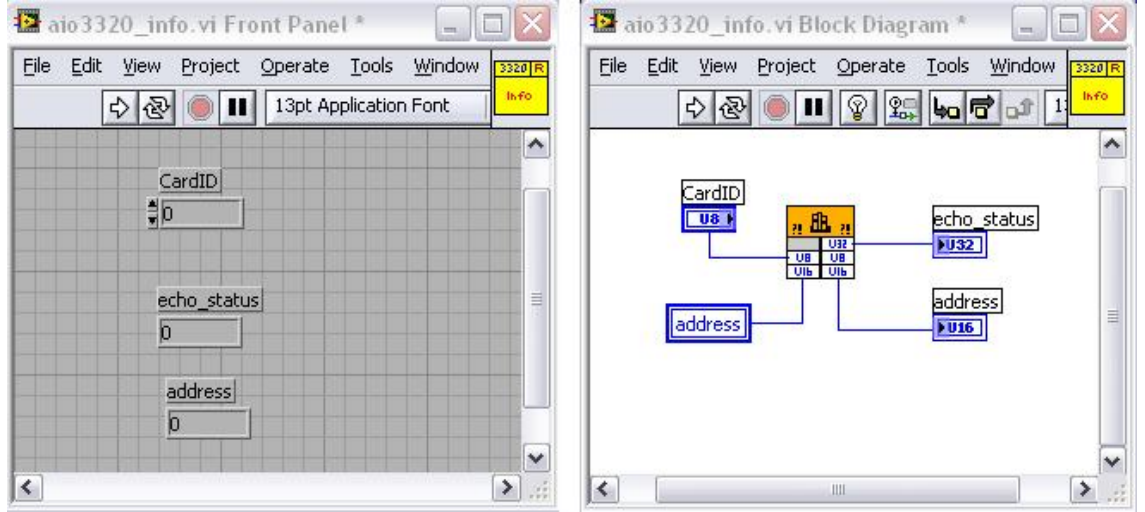


Şekil 3.100 Case Structure yapısına True sinyali gönderilmesi



Şekil 3.101 Case Structure yapısına False sinyali gönderilmesi

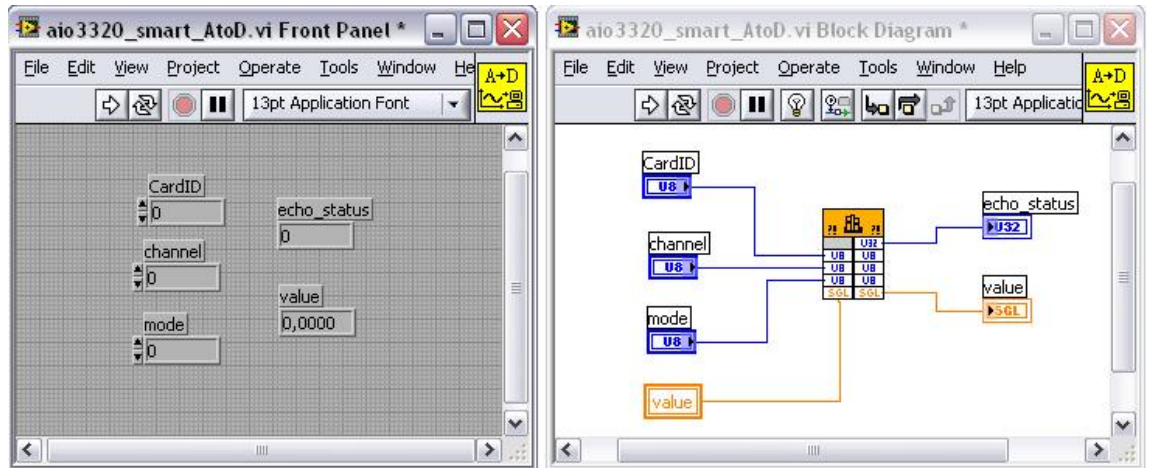
Yazılımla çalıştırılan yada kapatılan kartın anlık durumunu göstermek için info.vi sürücüsü kullanılır. Bu sürücüye ait blok diyagramı direk olarak yazılımın içine kopyalanır. Aşağıda info.vi sürücüsüne ait Ön panel ve Blok Diyagramı görülmektedir.



Şekil 3.102 info.vi sürücüsüne ait Ön panel ve Blok Diyagramı[13]

3.3.2.2.2 Analog Verinin Sayısal Veriye LabVIEW ile Dönüştürülmesi

Yazılım aracılığıyla AIO3320 kartın analog girişlerinden Analog sinyallerin okunup sayısal veriye dönüştürülebilmesi için smart_AtoD.vi sürücüsü kullanılır. Aşağıda smart_AtoD.vi sürücüsüne ait ön panel ve blok diyagramı görülmektedir.

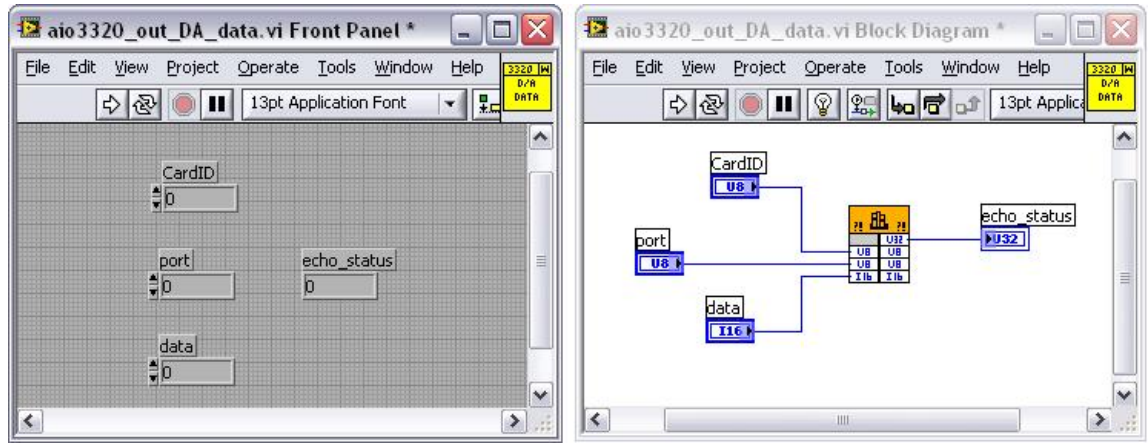


Şekil 3.103 smart_AtoD.vi sürücüsüne ait Ön panel ve Blok diyagramı[13]

Analog verinin okunup sayısal veriye dönüştürülmesi için smart_AtoD.vi blok diyagramında bazı ayarların yapılması gerekir. smart_AtoD.vi blok diyagramı incelendiğinde CardID, channel ve mode olmak üzere üç tane control elemanı, Value ve echo_status olmak üzere iki tane indicator görülmektedir. CardID Kontrolü kartımızın id numarası, channel Kontrolü hangi analog kanaldan okuma yapılacağı ve mode Kontrolü ile analog verinin giriş aralığı seçilir. Echo_status indicatoru hata kodunu, Value indicatoru ise analog bilginin sayısal veri karşılığını verir.

3.3.2.2.3 Sayısal Verinin Analog Veriye LabVIEW ile Dönüştürülmesi

AIO3320 kartla Yazılım aracılığıyla Sayısal verinin analog çıkışından analog veri olarak üretilmesi için Out_DA_data.vi sürücüsü kullanılır. Aşağıda Out_DA_data.vi sürücüsüne ait ön panel ve blok diyagramı görülmektedir.



Şekil 3.104 Out_DA_data.vi sürücüsüne ait Ön panel ve blok diyagramı[13]

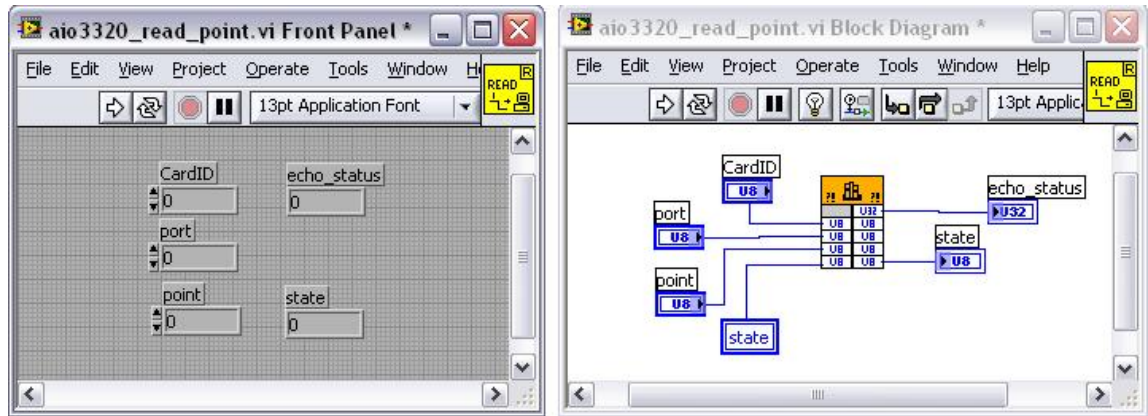
Sayısal verinin analog veriye dönüştürülmesi için smart_AtoD.vi blok diyagramında bazı ayarların yapılması gerekir. Out_DA_data.vi blok diyagramı incelendiğinde CardID, port ve data olmak üzere üç tane Kontrol nesnesi, echo_status olmak üzere bir tane Gösterge nesnesi görülmektedir. CardID Kontrolü kartımızın id numarası, port Kontrolüne hangi analog kanaldan çıkış yapılacağı ve data Kontrolü ile

analog bilgiye dönüştürülmek istenen sayısal veri yazılır. Echo_status göstergesi hata kodunu verir.

3.3.2.2.4 Sayısal Giriş ve Çıkış Sinyallerinin LabVIEW ile Kontrolü

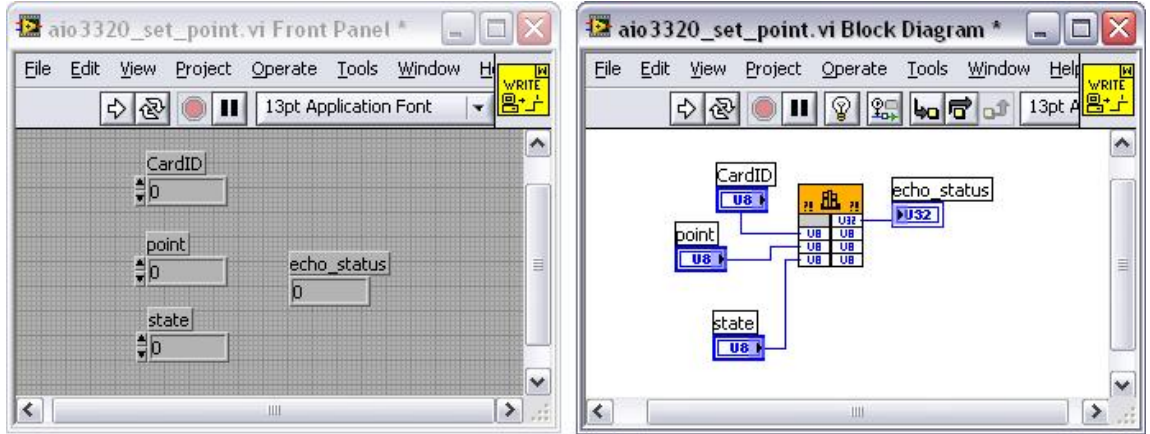
AIO3320 kartın sayısal giriş ve çıkışları ayrı ayrı kontrol edilir. Sayısal girişler read_point.vi sürücüsüyle okunur, sayısal çıkışlar ise set_point.vi sürücüsüyle kontrol edilir. Sayısal giriş ve çıkışların kontrol edilebilmesi için sürücülerin blok diyagramlarındaki fonksiyonlar yazılıma kopyalanmalıdır.

Read_point.vi sürücüsünün blok diyagramı incelendiğinde CardID, port ve point olmak üzere üç tane Kontrol, echo_status ve state olmak üzere iki adet gösterge nesnesi bulunmaktadır. cardID kartımızın id numarası, port kontrolü sayısal giriş için 0, sayısal çıkış için 1 olmalı, point kontrolü sayısal girişin kontrol edilecek olan biti (giriş kanalı) ve echo_status kartı hata kodunu bize verir.



Şekil 3.105 Read_point.vi sürücüsünün Ön panel ve Blok Diyagramı[13]

Set_point.vi sürücüsünün blok diyagramı incelendiğinde CardID, point ve state olmak üzere üç tane kontrol nesnesi ve echo_status olmak üzere bir adet gösterge nesnesi bulunmaktadır. cardID kartımızın id numarası, point kontrolü sayısal çıkışın kontrol edilecek olan biti (çıkış kanalı), state sayısal çıkışın alması gereken değer Lojik-0 veya Lojik-1 değeri ve echo_status kartı hata kodunu bize verir.



Şekil 3.106 Set_point.vi sürücüsünün Ön panel ve Blok diyagramı[13]

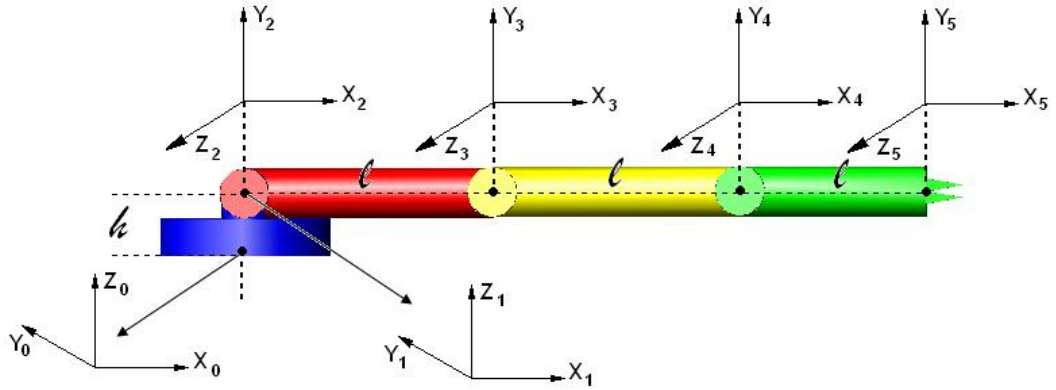
3.3.2.3 MATLAB Script ile İleri ve Ters Kinematik Hesaplamalar

Master koldan alınan eklem açılarına veya yazılımla oluşturulan slave kolun eklem açılarına göre konum vektörlerinin bulunması gerekir. Bu amaçla yazılımda bize ileri ve ters kinematik hesaplar gerekir. İleri ve ters kinematik hesaplamaların LabVIEW 'in kendi fonksiyonları ile yapılması uzun ve karmaşık olacağından yine LabVIEW 'in altında çalışan MATLAB scripti ile yapılır. MATLAB scripti programın blok diyagramına fonksiyon paletinden eklenir.

LabVIEW altında çalışan MATLAB scripti ileri ve ters kinematik hesapları yapabilmek için arka planda MATLAB programını çalıştırarak scripte yazılan MATLAB komut dizisini çalıştırarak LABVIEW programından aldığı eklem bilgilerine karşılık ileri kinematik, uç işlevcisinin koordinat bilgilerine göre ters kinematik sonuçlarını üretir. İleri ve ters kinematik hesaplamaların yapılabilmesi için öncelikle Slave kolun matematiksel modeli çıkartılıp ileri ve ters kinematik eşitlikleri bulunmalıdır. Bu nedenle öncelikle ileri ve ters kinematik modelleri ve konum vektörleri bulunur.

3.3.2.3.1 Master ve Slave Kolların İleri Kinematik Modeli ve Konum Vektörleri

İleri kinematik modelinin çıkarılması için öncelikle eklem değişkenleri ve sabitleri belirlendikten sonra koordinat sistemleri eklemlere yerleştirilir. Bunların nasıl yapılacağı İleri Kinematik bölümünde ayrıntılı olarak açıklanmıştır.



Şekil 3.107 Dört serbestlik dereceli robot kolun matematiksel modeli

Koordinat sistemleri eklemlere yerleştirildikten sonra D-H değişkenleri bulunur ve tabloya yazılır. Robotun hareket etmesiyle değişmeyen parametreler a_{i-1} uzuv uzunlukları ve α_{i-1} eksen açılarıdır. Değişen parametreler ise eklem döner ise θ_{i-1} eklem açısı eğer eklem prizmatik ise d_{i-1} eklem kaçıklığıdır.

Eksen No	D-H Değişkenleri				Eklem Değişkeni
	α_{i-1}	a_{i-1}	d_i	θ_i	
i	α_{i-1}	a_{i-1}	d_i	θ_i	d_i veya θ_i
1	$\alpha_0 = 0$	$a_0 = 0$	$d_1 = h$	θ_1	θ_1
2	$\alpha_1 = 90$	$a_1 = 0$	$d_2 = 0$	θ_2	θ_2
3	$\alpha_2 = 0$	$a_2 = l$	$d_3 = 0$	θ_3	θ_3
4	$\alpha_3 = 0$	$a_3 = l$	$d_4 = 0$	θ_4	θ_4
5	$\alpha_4 = 0$	$a_4 = l$	$d_5 = 0$	θ_5	θ_5

Tablo 3.14 D-H Değişkenleri

Beş serbestlik dereceli Slave kolun Homojen Dönüşüm Matrisi

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Öncelikle Homojen Dönüşüm Matrisinden her eklem Homojen dönüşüm matrisleri bulunur. Daha sonra her eklem Homojen Dönüşüm Matrisi birbiriyle çarpılarak 0_5T Homojen Dönüşüm Matrisi bulunur. 0_5T Homojen dönüşüm Matrisinin Konum vektörü Slave kolun uç işlevcisinin üç boyutlu koordinat düzlemindeki Px, Py ve Pz konumunu verir.

$${}^0_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$${}^0_5T = {}^0_1T * {}^1_2T * {}^2_3T * {}^3_4T * {}^4_5T$$

Birinci eklem için dönüşüm matrisi;

$${}^0_1T = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & a_0 \\ \sin \theta_1 \cos \alpha_0 & \cos \theta_1 \cos \alpha_0 & -\sin \alpha_0 & -\sin \alpha_0 d_1 \\ \sin \theta_1 \sin \alpha_0 & \cos \theta_1 \sin \alpha_0 & \cos \alpha_0 & \cos \alpha_0 d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 \cos 0 & \cos \theta_1 \cos 0 & -\sin 0 & -\sin 0.h \\ \sin \theta_1 \sin 0 & \cos \theta_1 \sin 0 & \cos 0 & \cos 0.h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7) \\ = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

İkinci eklem için dönüşüm matrisi;

$$\begin{aligned}
 {}^1_2T &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_1 \\ \sin \theta_2 \cos \alpha_1 & \cos \theta_2 \cos \alpha_1 & -\sin \alpha_1 & -\sin \alpha_1 d_2 \\ \sin \theta_2 \sin \alpha_1 & \cos \theta_2 \sin \alpha_1 & \cos \alpha_1 & \cos \alpha_1 d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ \sin \theta_2 \cos 90 & \cos \theta_2 \cos 90 & -\sin 90 & -\sin 90 \cdot 0 \\ \sin \theta_2 \sin 90 & \cos \theta_2 \sin 90 & \cos 90 & \cos 90 \cdot 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.8}$$

Üçüncü eklem için dönüşüm matrisi;

$$\begin{aligned}
 {}^2_3T &= \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_2 \\ \sin \theta_3 \cos \alpha_2 & \cos \theta_3 \cos \alpha_2 & -\sin \alpha_2 & -\sin \alpha_2 d_3 \\ \sin \theta_3 \sin \alpha_2 & \cos \theta_3 \sin \alpha_2 & \cos \alpha_2 & \cos \alpha_2 d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l \\ \sin \theta_3 \cos 0 & \cos \theta_3 \cos 0 & -\sin 0 & -\sin 0 \cdot 0 \\ \sin \theta_3 \sin 0 & \cos \theta_3 \sin 0 & \cos 0 & \cos 0 \cdot 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.9}$$

Dördüncü eklem için dönüşüm matrisi;

$$\begin{aligned}
 {}^3_4T &= \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & a_3 \\ \sin \theta_4 \cos \alpha_3 & \cos \theta_4 \cos \alpha_3 & -\sin \alpha_3 & -\sin \alpha_3 d_4 \\ \sin \theta_4 \sin \alpha_3 & \cos \theta_4 \sin \alpha_3 & \cos \alpha_3 & \cos \alpha_3 d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & l \\ \sin \theta_4 \cos 0 & \cos \theta_4 \cos 0 & -\sin 0 & -\sin 0 \cdot 0 \\ \sin \theta_4 \sin 0 & \cos \theta_4 \sin 0 & \cos 0 & \cos 0 \cdot 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & l \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.10}$$

Beşinci eklem için dönüşüm matrisi;

$$\begin{aligned}
 {}^4_5T &= \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & a_4 \\ \sin \theta_5 \cos \alpha_4 & \cos \theta_5 \cos \alpha_4 & -\sin \alpha_4 & -\sin \alpha_4 d_5 \\ \sin \theta_5 \sin \alpha_4 & \cos \theta_5 \sin \alpha_4 & \cos \alpha_4 & \cos \alpha_4 d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & l \\ \sin \theta_5 \cos 0 & \cos \theta_5 \cos 0 & -\sin 0 & -\sin 0 \cdot 0 \\ \sin \theta_5 \sin 0 & \cos \theta_5 \sin 0 & \cos 0 & \cos 0 \cdot 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & l \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.11}$$

Elde edilen beş ekleme ait dönüşüm matrisleri birbirleriyle çarpılarak Slave Kola ait dönüşüm matrisi elde edilir.

$${}^0_5T = {}^0_1T * {}^1_2T * {}^2_3T * {}^3_4T * {}^4_5T$$

$$\begin{aligned}
{}^0_2T = {}^0_1T * {}^1_2T &= \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos \theta_1 \cos \theta_2 & -\cos \theta_1 \sin \theta_2 & \sin \theta_1 & 0 \\ \sin \theta_1 \cos \theta_2 & -\sin \theta_1 \sin \theta_2 & -\cos \theta_1 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3.12}$$

$${}^0_3T = {}^0_2T * {}^2_3T = \begin{bmatrix} \cos \theta_1 \cos \theta_2 & -\cos \theta_1 \sin \theta_2 & \sin \theta_1 & 0 \\ \sin \theta_1 \cos \theta_2 & -\sin \theta_1 \sin \theta_2 & -\cos \theta_1 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

$${}^0_3T = \begin{bmatrix} \cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3 & -\cos \theta_1 \cos \theta_2 \sin \theta_3 - \cos \theta_1 \sin \theta_2 \cos \theta_3 & \sin \theta_1 & l \cos \theta_1 \cos \theta_2 \\ \sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3 & -\sin \theta_1 \cos \theta_2 \sin \theta_3 - \sin \theta_1 \sin \theta_2 \cos \theta_3 & -\cos \theta_1 & l \sin \theta_1 \cos \theta_2 \\ \sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3 & -\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3 & 0 & l \sin \theta_2 + h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned}
{}^0_4T = {}^0_3T * {}^3_4T &= \begin{bmatrix} \cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3 & -\cos \theta_1 \cos \theta_2 \sin \theta_3 - \cos \theta_1 \sin \theta_2 \cos \theta_3 & \sin \theta_1 & l \cos \theta_1 \cos \theta_2 \\ \sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3 & -\sin \theta_1 \cos \theta_2 \sin \theta_3 - \sin \theta_1 \sin \theta_2 \cos \theta_3 & -\cos \theta_1 & l \sin \theta_1 \cos \theta_2 \\ \sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3 & -\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3 & 0 & l \sin \theta_2 + h \\ 0 & 0 & 0 & 1 \end{bmatrix} * \\
&= \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & l \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos \theta_4 (\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) - \sin \theta_4 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \sin \theta_2 \cos \theta_3) \\ \cos \theta_4 (\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) - \sin \theta_4 (\sin \theta_1 \cos \theta_2 \sin \theta_3 + \sin \theta_1 \sin \theta_2 \cos \theta_3) \\ \cos \theta_4 (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + \sin \theta_4 (-\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3) \\ 0 \end{bmatrix} \\
&\quad \begin{matrix} -\sin \theta_4 (\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) - \cos \theta_4 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \sin \theta_2 \cos \theta_3) & \sin \theta_4 \\ -\sin \theta_4 (\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) - \cos \theta_4 (\sin \theta_1 \cos \theta_2 \sin \theta_3 + \sin \theta_1 \sin \theta_2 \cos \theta_3) & -\cos \theta_4 \\ -\sin \theta_4 (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + \cos \theta_4 (-\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3) & 0 \\ 0 & 0 \end{matrix} \tag{3.14}
\end{aligned}$$

$$\begin{bmatrix} l(\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) + l \cos \theta_1 \cos \theta_2 \\ l(\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) + l \sin \theta_1 \cos \theta_2 \\ l(\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + l \sin \theta_2 + h \\ 1 \end{bmatrix}$$

$${}^0T_5 = {}^0T_4 * {}^4T_5 =$$

$$\begin{bmatrix} \cos \theta_4 (\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) - \sin \theta_4 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \sin \theta_2 \cos \theta_3) \\ \cos \theta_4 (\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) - \sin \theta_4 (\sin \theta_1 \cos \theta_2 \sin \theta_3 + \sin \theta_1 \sin \theta_2 \cos \theta_3) \\ \cos \theta_4 (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + \sin \theta_4 (-\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3) \\ 0 \end{bmatrix}$$

$$\begin{array}{r} -\sin \theta_4 (\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) - \cos \theta_4 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \sin \theta_2 \cos \theta_3) \quad \sin \theta_1 \\ -\sin \theta_4 (\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) - \cos \theta_4 (\sin \theta_1 \cos \theta_2 \sin \theta_3 + \sin \theta_1 \sin \theta_2 \cos \theta_3) \quad -\cos \theta_1 \\ -\sin \theta_4 (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + \cos \theta_4 (-\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3) \quad 0 \\ 0 \quad 0 \end{array}$$

$$\begin{array}{l} l(\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) + l \cos \theta_1 \cos \theta_2 \\ l(\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) + l \sin \theta_1 \cos \theta_2 \\ l(\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + l \sin \theta_2 + h \\ 1 \end{array} \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & l \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \dots & \dots & \sin \theta_1 \\ \dots & \dots & -\cos \theta_1 \\ \dots & \dots & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.15)$$

$$\begin{array}{l} l(c\theta_4(c\theta_1c\theta_2c\theta_3 - c\theta_1s\theta_2s\theta_3) - s\theta_4(c\theta_1c\theta_2s\theta_3 + c\theta_1s\theta_2c\theta_3)) + l(c\theta_1c\theta_2c\theta_3 - c\theta_1s\theta_2s\theta_3) + l.c\theta_1c\theta_2 \\ l(c\theta_4(s\theta_1c\theta_2c\theta_3 - s\theta_1s\theta_2s\theta_3) - s\theta_4(s\theta_1c\theta_2s\theta_3 + s\theta_1s\theta_2c\theta_3)) + l.(s\theta_1c\theta_2c\theta_3 - s\theta_1s\theta_2s\theta_3) + l.s\theta_1c\theta_2 \\ l(c\theta_4(s\theta_2c\theta_3 + c\theta_2s\theta_3) + s\theta_4(-s\theta_2s\theta_3 + c\theta_2c\theta_3)) + l(s\theta_2c\theta_3 + c\theta_2s\theta_3) + l.s\theta_2 + h \\ 1 \end{array}$$

$$c=\cos, s=\sin$$

Kolon istasyona yerleştirildiği konum vektörü

$$\begin{array}{l} P_{X0} = 0 \\ P_{Y0} = 0 \\ P_{Z0} = 0 \end{array} \quad (3.16)$$

1 nolu eklem konum vektörü

$$\begin{array}{l} P_{X1} = 0 \\ P_{Y1} = 0 \\ P_{Z1} = h \end{array} \quad (3.17)$$

2 nolu eklemnin konum vektörü

$$\begin{aligned} P_{X2} &= 0 \\ P_{Y2} &= 0 \\ P_{Z2} &= h \end{aligned} \quad (3.18)$$

3 nolu eklemnin konum vektörü

$$\begin{aligned} P_{X3} &= l \cdot \cos \theta_1 \cos \theta_2 \\ P_{Y3} &= l \cdot \sin \theta_1 \cos \theta_2 \\ P_{Z3} &= l \cdot \sin \theta_2 + h \end{aligned} \quad (3.19)$$

4 nolu eklemnin konum vektörü

$$\begin{aligned} P_{X4} &= l \cdot (\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) + l \cdot \cos \theta_1 \cos \theta_2 \\ P_{Y4} &= l \cdot (\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) + l \cdot \sin \theta_1 \cos \theta_2 \\ P_{Z4} &= l \cdot (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + l \cdot \sin \theta_2 + h \end{aligned} \quad (3.20)$$

5 nolu eklemnin (uç işlevcisinin) konum vektörü

$$\begin{aligned} P_{X5} &= l \cdot (\cos \theta_4 (\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) - \\ &\quad \sin \theta_4 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \sin \theta_2 \cos \theta_3)) \\ &\quad + l \cdot (\cos \theta_1 \cos \theta_2 \cos \theta_3 - \cos \theta_1 \sin \theta_2 \sin \theta_3) + l \cdot \cos \theta_1 \cos \theta_2 \\ P_{Y5} &= l \cdot (\cos \theta_4 (\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) - \\ &\quad \sin \theta_4 (\sin \theta_1 \cos \theta_2 \sin \theta_3 + \sin \theta_1 \sin \theta_2 \cos \theta_3)) \\ &\quad + l \cdot (\sin \theta_1 \cos \theta_2 \cos \theta_3 - \sin \theta_1 \sin \theta_2 \sin \theta_3) + l \cdot \sin \theta_1 \cos \theta_2 \\ P_{Z5} &= l \cdot (\cos \theta_4 (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + \\ &\quad \sin \theta_4 (-\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3)) \\ &\quad + l \cdot (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + l \cdot \sin \theta_2 + h \end{aligned} \quad (3.21)$$

3.3.2.3.2 Master ve Slave Kolların Ters Kinematik Modeli ve Eklem Açılı

İleri yön kinematiğine ait ${}^0_5T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T$ dönüşüm matrisinin her iki tarafını ${}^0_1T^{-1}$ ile çarpalım.

$${}^0_1T^{-1} {}^0_5T = {}^0_1T^{-1} {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T \quad (3.22)$$

Bilindiği gibi ${}^0_1T^{-1} {}^0_1T = I$ olduğundan denklem aşağıdaki gibi olur.

$${}^1_5T = {}^1_2T {}^2_3T {}^3_4T {}^4_5T \quad (3.23)$$

Öncelikle her dönüşüm matrisinin tersi bulunur. Dönüşüm matrisinin tersinin bulunması Ters Kinematik bölümünde ayrıntılı olarak anlatılmıştır.

0_1T Dönüşüm matrisinin tersi ${}^0_1T^{-1}$ matrisine eşittir.

$${}^0_1T^{-1} = \begin{bmatrix} {}^0_1R^T & -{}^0_1R^T * {}^0P_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.24)$$

$${}^0_1R = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^0_1R^T = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.25)$$

$$-{}^0_1R^T {}^0P_1 = \begin{bmatrix} -\cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & -\cos \theta_1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -h \end{bmatrix} \quad (3.26)$$

$${}^0_1T^{-1} = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.27)$$

1_2T Dönüşüm matrisinin tersi ${}^1_2T^{-1}$ matrisine eşittir.

$${}^1_2T^{-1} = \begin{bmatrix} {}^1_2R^T & -{}^1_2R^T * {}^1P_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.28)$$

$${}^1_2R = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 \\ 0 & 0 & -1 \\ \sin \theta_2 & \cos \theta_2 & 0 \end{bmatrix} \quad {}^1_2R^T = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 \\ -\sin \theta_2 & 0 & \cos \theta_2 \\ 0 & -1 & 0 \end{bmatrix} \quad (3.29)$$

$$-{}^1_2R^T * {}^1P_2 = \begin{bmatrix} -\cos \theta_2 & 0 & -\sin \theta_2 \\ \sin \theta_2 & 0 & -\cos \theta_2 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.30)$$

$${}^1_2T^{-1} = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.31)$$

2_3T Dönüşüm matrisinin tersi ${}^2_3T^{-1}$ matrisine eşittir.

$${}^2_3T^{-1} = \begin{bmatrix} {}^2_3R^T & -{}^2_3R^T * {}^2P_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.32)$$

$${}^2_3R = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^2_3R^T = \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 \\ -\sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.33)$$

$$-{}^2_3R^T * {}^2P_3 = \begin{bmatrix} -\cos \theta_3 & -\sin \theta_3 & 0 \\ \sin \theta_3 & -\cos \theta_3 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -l \cdot \cos \theta_3 \\ l \cdot \sin \theta_3 \\ 0 \end{bmatrix} \quad (3.34)$$

$${}^2_3T^{-1} = \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 & -l \cdot \cos \theta_3 \\ -\sin \theta_3 & \cos \theta_3 & 0 & l \cdot \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.35)$$

3_4T Dönüşüm matrisinin tersi ${}^3_4T^{-1}$ matrisine eşittir.

$${}^0_4T^{-1} = \begin{bmatrix} {}^0_4R^T & -{}^0_4R^T * {}^0P_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.36)$$

$${}^3_4R = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 \\ \sin \theta_4 & \cos \theta_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad {}^2_3R^T = \begin{bmatrix} \cos \theta_4 & \sin \theta_4 & 0 \\ -\sin \theta_4 & \cos \theta_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.37)$$

$$-{}^3_4R^T {}^3P_4 = \begin{bmatrix} -\cos \theta_4 & -\sin \theta_4 & 0 \\ \sin \theta_4 & -\cos \theta_4 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -l \cos \theta_4 \\ l \sin \theta_4 \\ 0 \end{bmatrix} \quad (3.38)$$

$${}^3_4T^{-1} = \begin{bmatrix} \cos \theta_4 & \sin \theta_4 & 0 & -l \cos \theta_4 \\ -\sin \theta_4 & \cos \theta_4 & 0 & l \sin \theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.39)$$

${}^0_1T^{-1}$, ${}^0_2T^{-1}$, ${}^0_3T^{-1}$ ve ${}^0_4T^{-1}$ matrislerini teker teker 0_5T matrisiyle çarparak sırasıyla 1_5T , 2_5T , 3_5T ve 4_5T matrisleri elde edilir.

$[{}^0_1T]^{-1} {}^0_5T$ Çarpımını bulduk. Şimdide ${}^1_2T {}^2_3T {}^3_4T {}^4_5T$ çarpımını bulalım ve iki sonucu birbirine eşitleyelim.

$$[{}^0_1T]^{-1} {}^0_5T = {}^1_5T \quad (3.40)$$

$${}^0_1T^{-1} * {}^0_5T = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.41)$$

$$= \begin{bmatrix} \dots & \dots & \dots & p_x \cos \theta_1 + p_y \sin \theta_1 \\ \dots & \dots & \dots & -p_x \sin \theta_1 + p_y \cos \theta_1 \\ \dots & \dots & \dots & p_z - h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

${}^0_2T^{-1} {}^0_5T$ Çarpımını bulduk. Şimdide ${}^2_3T {}^3_4T {}^4_5T$ çarpımını bulalım ve iki sonucu birbirine eşitleyelim.

$${}^0_1T^{-1} [{}^1_2T]^{-1} {}^0_5T = {}^2_3T \quad (3.42)$$

$$\begin{aligned}
{}^0_1T^{-1} * {}^1_2T^{-1} * {}^0_5T &= \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 & 0 & \cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2 & 0 \\ -\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2 & 0 & -\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2 & 0 \\ 0 & -1 & 0 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \dots & \dots & \dots & p_x(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) + p_z(\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) \\ \dots & \dots & \dots & p_x(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) + p_z(-\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) \\ \dots & \dots & \dots & -p_y - h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.43)
\end{aligned}$$

${}^0_3T^{-1} {}^0_5T$ Çarpımını bulduk. Şimdide ${}^3_4T {}^4_5T$ çarpımını bulalım ve iki sonucu birbirine eşitleyelim.

$$\begin{aligned}
& \left[{}^0_1T \right]^{-1} * \left[{}^1_2T \right]^{-1} * \left[{}^2_3T \right]^{-1} * {}^0_5T = {}^3_5T \\
& = \begin{bmatrix} \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 & 0 & \cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2 & 0 \\ -\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2 & 0 & -\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2 & 0 \\ 0 & -1 & 0 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 & -l \cos \theta_3 \\ -\sin \theta_3 & \cos \theta_3 & 0 & l \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& \qquad \qquad \qquad * \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} \\
& = \begin{bmatrix} \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) & \sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \\ \cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) & \sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) \\ \sin \theta_3 & -\cos \theta_3 \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} \cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2 & -l \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \\ -\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2 & -l \cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) \\ 0 & -l \sin \theta_3 - h \\ 0 & 1 \end{bmatrix} * \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} \\
& = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ 0 & 0 & 0 \end{bmatrix} \qquad \qquad \qquad (3.44) \\
& \left. \begin{aligned} & p_x (c\theta_3 (c\theta_1 c\theta_2 - s\theta_1 s\theta_2)) + p_y (s\theta_3 (c\theta_1 c\theta_2 - s\theta_1 s\theta_2)) + p_z (c\theta_1 s\theta_2 + s\theta_1 c\theta_2) - l c\theta_3 (c\theta_1 c\theta_2 - s\theta_1 s\theta_2) \\ & p_x (c\theta_3 (-s\theta_1 c\theta_2 - c\theta_1 s\theta_2)) + p_y (s\theta_3 (-s\theta_1 c\theta_2 - c\theta_1 s\theta_2)) + p_z (-s\theta_1 s\theta_2 + c\theta_1 c\theta_2) - l c\theta_3 (-s\theta_1 c\theta_2 - c\theta_1 s\theta_2) \\ & p_x (s\theta_3) + p_y (-c\theta_3) + 0 - l s\theta_3 - h \\ & 1 \end{aligned} \right\}
\end{aligned}$$

$\left[{}^0_4T \right]^{-1} {}^0_5T$ çarpımıyla 4_5T çarpımını birbirine eşitleyelim.

$$\begin{aligned}
& \left[{}^0T_1 \right]^{-1} * \left[{}^1T_2 \right]^{-1} * \left[{}^2T_3 \right]^{-1} * \left[{}^3T_4 \right]^{-1} * {}^0T_5 = {}^4T_5 \\
& = \begin{bmatrix} \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) & \sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \\ \cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) & \sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) \\ \sin \theta_3 & -\cos \theta_3 \\ 0 & 0 \\ \cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2 & -l \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \\ -\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2 & -l \cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) \\ 0 & -l \sin \theta_3 - h \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos \theta_4 & \sin \theta_4 & 0 & -l \cos \theta_4 \\ -\sin \theta_4 & \cos \theta_4 & 0 & l \sin \theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& \quad * \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} \\
& = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ 0 & 0 & 0 \end{bmatrix} \\
& \quad \left. \begin{aligned} & p_x (\cos \theta_4 \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) - \sin \theta_4 (\sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2))) + \\ & p_y (\sin \theta_4 (\cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \cos \theta_4 (\sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2))) + \\ & p_z (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) - l \cos \theta_4 (\cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \\ & l \sin \theta_4 (\sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) - l \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \\ \\ & p_x (\cos \theta_4 (\cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) - \sin \theta_4 (\sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2))) + \\ & p_y (\sin \theta_4 (\cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \cos \theta_4 (\sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2))) + \\ & p_z (-\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) - l \cos \theta_4 (\cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \\ & l \sin \theta_4 (\sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) - l \cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) \\ \\ & p_x (\sin \theta_3 \cos \theta_4 + \cos \theta_3 \sin \theta_4) + \\ & p_y (\sin \theta_3 \sin \theta_4 - \cos \theta_3 \cos \theta_4) + 0 \\ & -l \cos \theta_4 \sin \theta_3 - l \sin \theta_4 \cos \theta_3 - l \sin \theta_3 - h \\ & \quad \quad \quad 1 \end{aligned} \right\} \quad (3.45)
\end{aligned}$$

$${}^0_5T = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix} \dots \left[\begin{array}{l} l(\cos\theta_4(\cos\theta_1\cos\theta_2\cos\theta_3 - \cos\theta_1\sin\theta_2\sin\theta_3) - \\ \sin\theta_4(\cos\theta_1\cos\theta_2\sin\theta_3 + \cos\theta_1\sin\theta_2\cos\theta_3)) + \\ l(\cos\theta_1\cos\theta_2\cos\theta_3 - \cos\theta_1\sin\theta_2\sin\theta_3) + l.\cos\theta_1\cos\theta_2 \\ \\ l(\cos\theta_4(\sin\theta_1\cos\theta_2\cos\theta_3 - \sin\theta_1\sin\theta_2\sin\theta_3) - \\ \sin\theta_4(\sin\theta_1\cos\theta_2\sin\theta_3 + \sin\theta_1\sin\theta_2\cos\theta_3)) + \\ \dots \\ l(\sin\theta_1\cos\theta_2\cos\theta_3 - \sin\theta_1\sin\theta_2\sin\theta_3) + l.\sin\theta_1\cos\theta_2 \\ \\ l(\cos\theta_4(\sin\theta_2\cos\theta_3 + \cos\theta_2\sin\theta_3) + \\ \sin\theta_4(-\sin\theta_2\sin\theta_3 + \cos\theta_2\cos\theta_3)) + \\ l(\sin\theta_2\cos\theta_3 + \cos\theta_2\sin\theta_3) + l.\sin\theta_2 + h \\ 1 \end{array} \right] \quad (3.46)$$

$${}^1_5T = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix} \dots \left[\begin{array}{l} l(\cos\theta_4(\cos\theta_2\cos\theta_3 - \sin\theta_2\sin\theta_3) - \\ \sin\theta_4(\cos\theta_2\sin\theta_3 + \sin\theta_2\cos\theta_3)) + \\ l(\cos\theta_2\cos\theta_3 - \sin\theta_2\sin\theta_3) + l.\cos\theta_2 \\ \\ 0 \\ \\ \dots \\ l(\cos\theta_4(\sin\theta_2\cos\theta_3 + \cos\theta_2\sin\theta_3) + \\ \sin\theta_4(-\sin\theta_2\sin\theta_3 + \cos\theta_2\cos\theta_3)) + \\ l(\sin\theta_2\cos\theta_3 + \cos\theta_2\sin\theta_3) + l.\sin\theta_2 + h \\ 1 \end{array} \right] \quad (3.47)$$

$${}^2_5T = \begin{bmatrix} \dots & \dots & \dots & l(\cos\theta_4\cos\theta_3 - \sin\theta_4\sin\theta_3) + l.\cos\theta_3 + l \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & l(\cos\theta_4\sin\theta_3 + \sin\theta_4\cos\theta_3) + l.\sin\theta_3 + h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.48)$$

$${}^3_5T = \begin{bmatrix} \dots & \dots & \dots & l.\cos\theta_4 + l + l \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & l.\sin\theta_4 + h \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots & l.\cos\theta_4 + 2l \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & l.\sin\theta_4 + h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.49)$$

$${}^4_5T = \begin{bmatrix} \dots & \dots & \dots & l + l + l \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & h \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots & 3l \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.50)$$

$$\begin{aligned}
\begin{bmatrix} 0 \\ 1 \end{bmatrix} T^{-1} \begin{bmatrix} 0 \\ 5 \end{bmatrix} T = \begin{bmatrix} \dots & \dots & \dots & p_x \cos \theta_1 + p_y \sin \theta_1 \\ \dots & \dots & \dots & p_x - \sin \theta_1 + p_y \cos \theta_1 \\ \dots & \dots & \dots & p_z - h \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
= \begin{bmatrix} \dots & \dots & \dots & \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix} \\
\left. \begin{aligned} & l(\cos \theta_4 (\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) - \\ & \sin \theta_4 (\cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3)) + \\ & l(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) + l \cos \theta_2 \\ & \dots & 0 \\ & l(\cos \theta_4 (\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + \\ & \sin \theta_4 (-\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3)) + \\ & l(\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + l \sin \theta_2 + h \\ & \dots & 1 \end{aligned} \right\} \quad (3.51)
\end{aligned}$$

$$\begin{aligned}
\begin{bmatrix} 0 \\ 1 \end{bmatrix} T^{-1} \begin{bmatrix} 1 \\ 2 \end{bmatrix} T^{-1} \begin{bmatrix} 0 \\ 5 \end{bmatrix} T = \begin{bmatrix} \dots & \dots & \dots & p_x (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) + p_z (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) \\ \dots & \dots & \dots & p_x (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) + p_z (-\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) \\ \dots & \dots & \dots & -p_y - h \\ 0 & 0 & 0 & 1 \end{bmatrix} = \quad (3.52) \\
\left[\begin{aligned} & \dots & \dots & \dots & l(\cos \theta_4 \cos \theta_3 - \sin \theta_4 \sin \theta_3) + l \cos \theta_3 + l \\ & \dots & \dots & \dots & 0 \\ & \dots & \dots & \dots & l(\cos \theta_4 \sin \theta_3 + \sin \theta_4 \cos \theta_3) + l \sin \theta_3 + h \\ 0 & 0 & 0 & & 1 \end{aligned} \right]
\end{aligned}$$

$$\begin{aligned}
& \begin{bmatrix} 0 \\ 1 \end{bmatrix} T^{-1} * \begin{bmatrix} 1 \\ 2 \end{bmatrix} T^{-1} * \begin{bmatrix} 2 \\ 3 \end{bmatrix} T^{-1} * {}^0_5 T = {}^3_5 T = \\
& = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ 0 & 0 & 0 \end{bmatrix} \\
& \qquad \left. \begin{aligned} & p_x (\cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \\ & p_y (\sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \\ & p_z (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) - l \cdot \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \\ & p_x (\cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \\ & p_y (\sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \\ & p_z (-\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) - l \cdot \cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) \\ & p_x (\sin \theta_3) + p_y (-\cos \theta_3) + 0 - l \cdot \sin \theta_3 - h \\ & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad 1 \end{aligned} \right\} \\
& = \begin{bmatrix} \dots & \dots & \dots & l \cdot \cos \theta_4 + 2l \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & l \cdot \sin \theta_4 + h \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.53}
\end{aligned}$$

$$\begin{aligned}
& \begin{bmatrix} 0 \\ 1 \end{bmatrix} T^{-1} * \begin{bmatrix} 1 \\ 2 \end{bmatrix} T^{-1} * \begin{bmatrix} 2 \\ 3 \end{bmatrix} T^{-1} * \begin{bmatrix} 3 \\ 4 \end{bmatrix} T^{-1} * {}^0_5 T = {}^4_5 T = \\
& = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ 0 & 0 & 0 \end{bmatrix} \\
& \qquad \left. \begin{aligned} & p_x (\cos \theta_4 \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) - \sin \theta_4 (\sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2))) + \\ & p_y (\sin \theta_4 (\cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \cos \theta_4 (\sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2))) + \\ & p_z (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) - l \cdot \cos \theta_4 (\cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \\ & l \cdot \sin \theta_4 (\sin \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) - l \cdot \cos \theta_3 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) \\ \\ & p_x (\cos \theta_4 (\cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) - \sin \theta_4 (\sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2))) + \\ & p_y (\sin \theta_4 (\cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \cos \theta_4 (\sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2))) + \\ & p_z (-\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) - l \cdot \cos \theta_4 (\cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \\ & l \cdot \sin \theta_4 (\sin \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) - l \cdot \cos \theta_3 (-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) \\ \\ & p_x (\sin \theta_3 \cos \theta_4 + \cos \theta_3 \sin \theta_4) + \\ & p_y (\sin \theta_3 \sin \theta_4 - \cos \theta_3 \cos \theta_4) + 0 \\ & - l \cdot \cos \theta_4 \sin \theta_3 - l \cdot \sin \theta_4 \cos \theta_3 - l \cdot \sin \theta_3 - h \\ & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad 1 \end{aligned} \right\} \\
& = \begin{bmatrix} \dots & \dots & \dots & 3l \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.54}
\end{aligned}$$

İşlemlerinden sonra aşağıdaki ters kinematik eşitlikler elde edilir.

$$\begin{aligned}
1. \quad p_x \cos \theta_1 + p_y \sin \theta_1 &= l(\cos \theta_4(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) - \\
&\quad \sin \theta_4(\cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3)) + \\
&\quad l(\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3) + l \cos \theta_2 \\
2. \quad -p_x \sin \theta_1 + p_y \cos \theta_1 &= 0
\end{aligned} \tag{3.55}$$

$$\begin{aligned}
3. \quad p_z - h &= l(\cos \theta_4(\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + \\
&\quad \sin \theta_4(-\sin \theta_2 \sin \theta_3 + \cos \theta_2 \cos \theta_3)) + \\
&\quad l(\sin \theta_2 \cos \theta_3 + \cos \theta_2 \sin \theta_3) + l \sin \theta_2 + h \\
4. \quad p_x(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) + p_z(\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) &= \\
&\quad l(\cos \theta_4 \cos \theta_3 - \sin \theta_4 \sin \theta_3) + l \cos \theta_3 + l \\
5. \quad p_x(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) + p_z(-\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) &= 0
\end{aligned} \tag{3.56}$$

$$6. \quad -p_y - h = l(\cos \theta_4 \sin \theta_3 + \sin \theta_4 \cos \theta_3) + l \sin \theta_3 + h$$

$$\begin{aligned}
7. \quad p_x(\cos \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \\
p_y(\sin \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \\
p_z(\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) - \\
l \cos \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) = l \cos \theta_4 + 2l \\
8. \quad p_x(\cos \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \\
p_y(\sin \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \\
p_z(-\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) - \\
l \cos \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) = 0
\end{aligned} \tag{3.57}$$

$$9. \quad p_x(\sin \theta_3) + p_y(-\cos \theta_3) + 0 - l \sin \theta_3 - h = l \sin \theta_4 + h$$

$$\begin{aligned}
10. \quad p_x(\cos \theta_4 \cos \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) - \sin \theta_4(\sin \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2))) + \\
p_y(\sin \theta_4(\cos \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \cos \theta_4(\sin \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2))) + \\
p_z(\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2) - l \cos \theta_4(\cos \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) + \\
l \sin \theta_4(\sin \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2)) - l \cos \theta_3(\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2) = 3l \\
11. \quad p_x(\cos \theta_4(\cos \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) - \sin \theta_4(\sin \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2))) + \\
p_y(\sin \theta_4(\cos \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \cos \theta_4(\sin \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2))) + \\
p_z(-\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) - l \cos \theta_4(\cos \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) + \\
l \sin \theta_4(\sin \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2)) - l \cos \theta_3(-\sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2) = 0
\end{aligned} \tag{3.58}$$

$$\begin{aligned}
12. \quad p_x(\sin \theta_3 \cos \theta_4 + \cos \theta_3 \sin \theta_4) + p_y(\sin \theta_3 \sin \theta_4 - \cos \theta_3 \cos \theta_4) \\
- l \cos \theta_4 \sin \theta_3 - l \sin \theta_4 \cos \theta_3 - l \sin \theta_3 - h = h
\end{aligned}$$

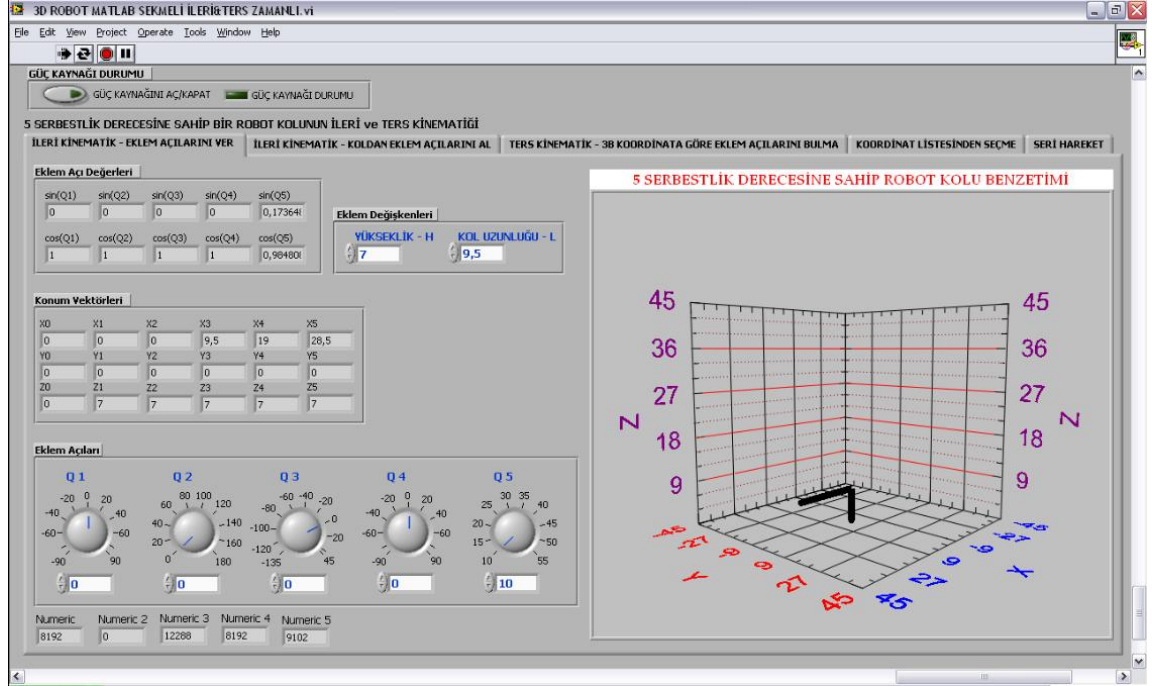
3.3.2.4 Kontrol Programı ve Sekmeler

Kontrol programı Windows Xp işletim sistemi altında çalışan LabVIEW Grafiksel programlama diliyle hazırlanmıştır. Sistemimizdeki Slave kol Hazırlanan bu Kontrol programıyla bilgisayar ortamında üretilen eklem açılarına, Master koldan alınan eklem açılarına ve yine bilgisayar ortamında verilen uç işlevcisi koordinatlarına göre hareket yeteneği kazanır. Kontrol programı ile Slave kol üç temel işleve sahip olsa da bu işlevlerin türevleri olan beş tane sekme bulunmaktadır.

Bunlar;

- Bilgisayar ortamında üretilen eklem açılarına göre Slave Kolun hareket ettirilmesi.
- Master koldan okunan eklem açılarına göre Slave kolun hareket ettirilmesi.
- Bilgisayar ortamında üretilen üç boyutlu uç işlevcisi koordinatına göre Slave kolun hareket ettirilmesi.
- Bilgisayar ortamında üretilen üç boyutlu uç işlevcisi koordinat listesine göre Slave kolun hareket ettirilmesi.
- Slave kolun önceden programlanmış bir hareket dizisine göre periyodik olarak hareket ettirilmesi.

Aşağıda Şekil 3.108 'de Kontrol programının kullanıcı arayüzüne ait genel görünüş bulunmaktadır.



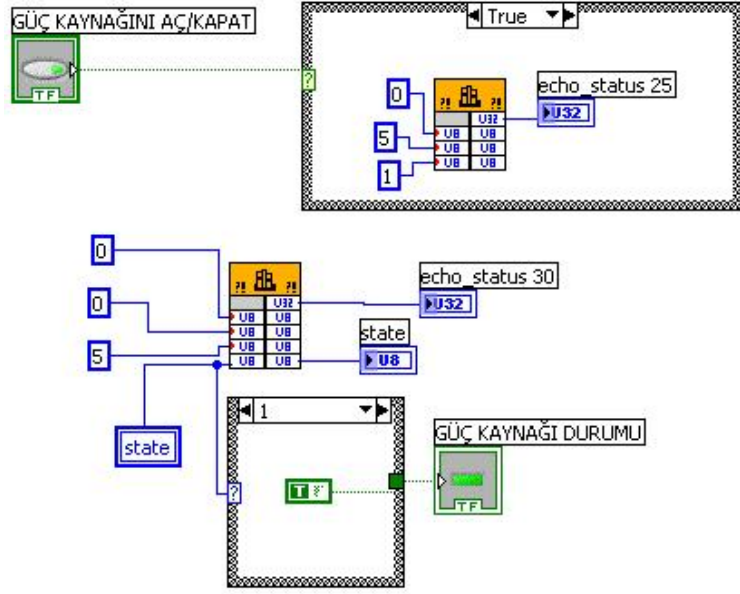
Şekil 3.108 Kontrol programının kullanıcı arayüzü

Kontrol programının bilgisayar ortamında çalışması ve durdurulması için Şekil 3.109 'deki 1 nolu balonun işaret ettiği soldan sağa doğru sırasıyla Run, Run Continuously, Abort Execution, Pause butonları kullanılır. Slave koldaki Servo motorlar ile Master koldaki potansiyometreleri besleyen güç kaynağını açmak ve kapatmak için Şekil 3.109 'deki 2 nolu balonun işaret ettiği buton kullanılmaktadır.[11] Ayrıca Güç kaynağının çalışma durumunu gösteren bir adet Sanal LED göstere bulunmaktadır.

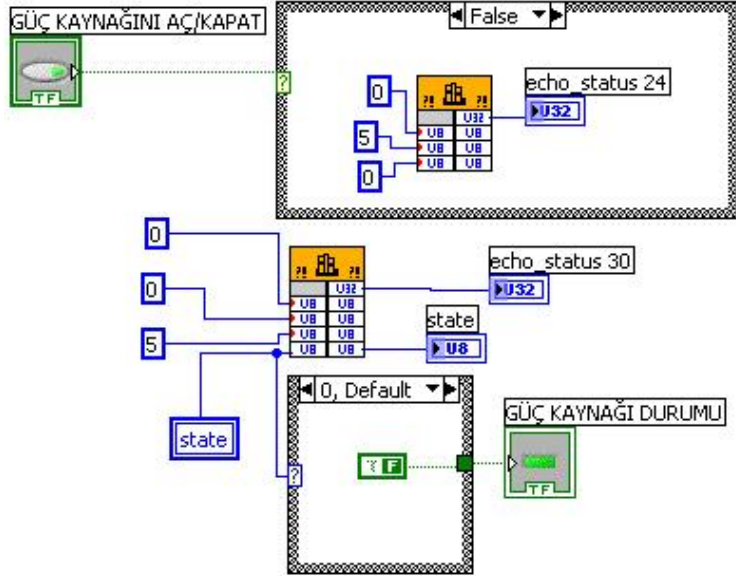


Şekil 3.109 Simülasyon başlatma ve durdurma butonları

Güç Kaynağını Aç/Kapat Kontrol nesnesiyle Güç Kaynağı Durumu göstere nesnesinin blok diyagramı Şekil 3.110 'da Güç kaynağı açıkken, Şekil 3.109 'da ise kapalıyken durumunu göstermektedir.



Şekil 3.110 Güç Kaynağı açık durumda iken blok diyagram durumu

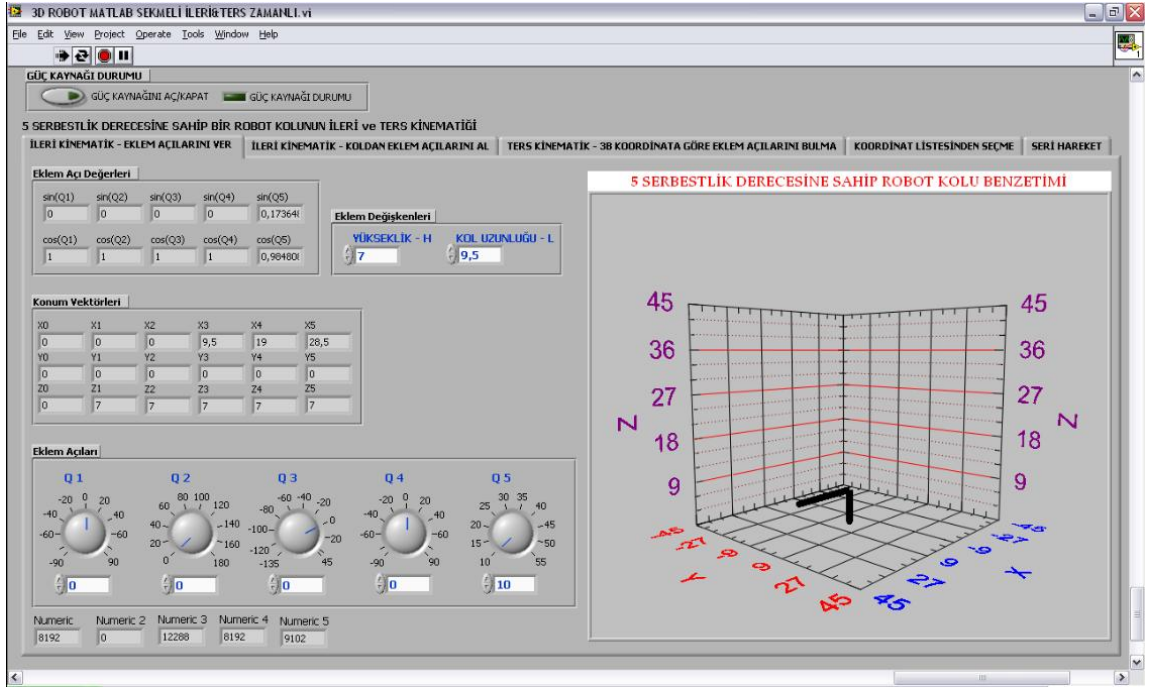


Şekil 3.111 Güç Kaynağı kapalı durumda iken blok diyagram durumu

3.3.2.4.1 İleri Kinematik – Eklem Açılarını Ver

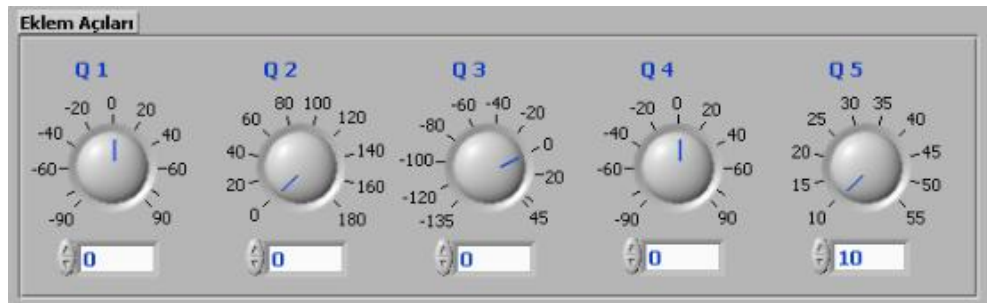
Kontrol programının birinci sekmesi İleri Kinematik – Eklem Açılarını Ver ‘dir.

Şekil 3.112



Şekil 3.112 İleri Kinematik – Eklem Açılarını Ver

Bu sekmede Şekil 3.113 ‘deki Q1, Q2, Q3, Q4 ve Q5 sanal potansiyometreleriyle bilgisayar ortamında üretilen eklem açlarına ve eklem değişkenlerine göre slave kolun eklemlerine hareket veren servo motorları sürmek için gerekli sinyal JAC3320 ADC kartıyla üretilir.



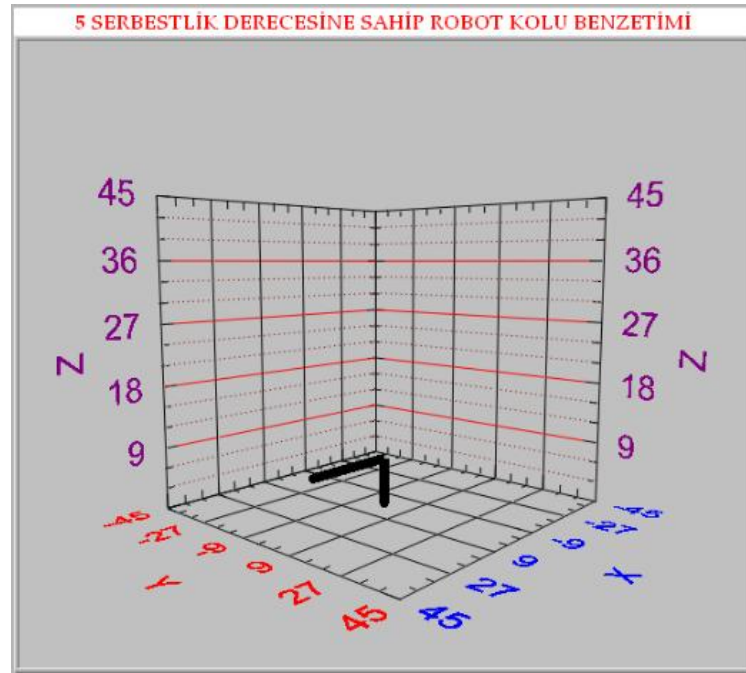
Şekil 3.113 Sanal potansiyometreleri

JAC3320 ile servo motorları sürmek için üretilen Analog sinyal Servo Motor Sürücü Kartıyla PWM sinyal formatına dönüştürülerek servo motorlar kontrol edilir. Ayrıca Q1, Q2, Q3, Q4 ve Q5 sanal potansiyometreleriyle bilgisayar ortamında üretilen eklemlerin açılımlarına göre İleri yön kinematiği hesaplanır ve eklemlere ait konum vektörleri bulunur ve sekmede Şekil 3.114 'de görüntülenir.

Konum Vektörleri					
X0	X1	X2	X3	X4	X5
0	0	0	9,5	19	28,5
Y0	Y1	Y2	Y3	Y4	Y5
0	0	0	0	0	0
Z0	Z1	Z2	Z3	Z4	Z5
0	7	7	7	7	7

Şekil 3.114 Eklemlere ait konum vektörleri

Bu konum vektörlerine göre Slave kolun üç boyutlu koordinat sisteminde aldığı konumunun gerçek zamanlı çizilen grafiği 3D Curve nesnesiyle bu sekmede Şekil 3.115 'de görüntülenir.



Şekil 3.115 Slave kolun anlık üç boyutlu grafiği

Slave ve master kola ait eklem deęişkenleri; Ana çerçeveye olan yükseklięi H ve kol uzunluęu L sabittir. Bu nedenle programda eklem deęişkenleri olarak Şekil 3.116 'de görüntülenmektedir.

YÜKSEKLİK - H	KOL UZUNLUęU - L
7	9,5

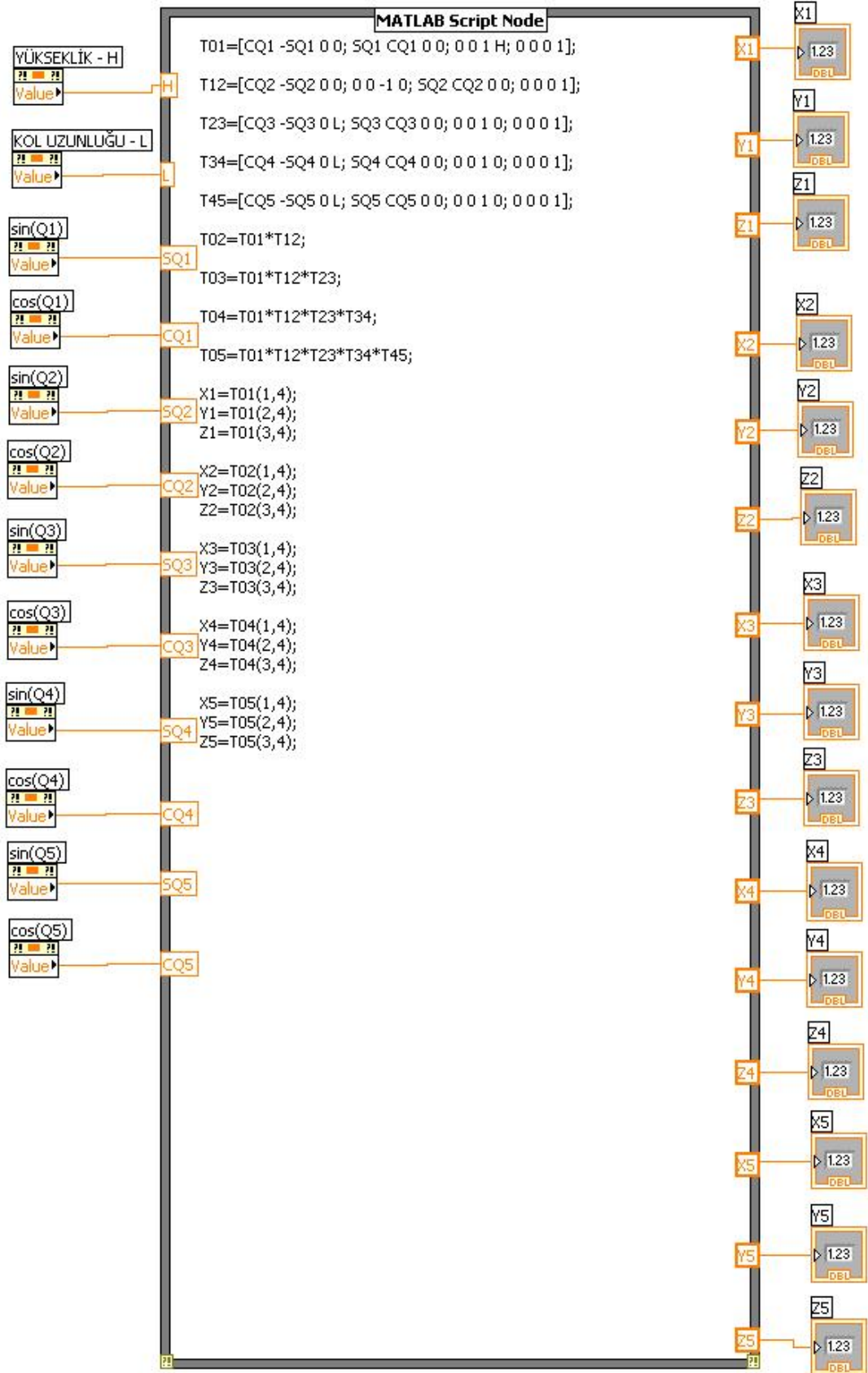
Şekil 3.116 Eklem Deęişkenleri

Ayrıca bu sekmede Q_1 , Q_2 , Q_3 , Q_4 ve Q_5 sanal potansiyometreleriyle bilgisayar ortamında üretilen eklem açılarının Sinüs ve Kosinüs karşılıkları bilgi amaçlı olarak programda Şekil 3.117 'de görüntülenmektedir.

sin(Q1)	sin(Q2)	sin(Q3)	sin(Q4)	sin(Q5)
0	0	0	0	0,17364
cos(Q1)	cos(Q2)	cos(Q3)	cos(Q4)	cos(Q5)
1	1	1	1	0,98480

Şekil 3.117 Eklem açılarının Sin ve Cos karşılıkları

Bu sekmede ileri kinematik matematiksel hesaplarının yapılabilmesi için LabVIEW blok diyagramına eklenebilen MATLAB scripti (Şekil 3.118) kullanılmıştır.



Şekil 3.118 İleri kinematik MATLAB scripti

MATLAB scriptine aşağıdaki MATLAB komutları yazılarak slave kolun ileri kinematik hesapları arka planda MATLAB programı ile hesaplanarak program içine alınır.

$T01=[CQ1 \ -SQ1 \ 0 \ 0; \ SQ1 \ CQ1 \ 0 \ 0; \ 0 \ 0 \ 1 \ H; \ 0 \ 0 \ 0 \ 1];$ T01 Matrisi oluşturulur.

$T12=[CQ2 \ -SQ2 \ 0 \ 0; \ 0 \ 0 \ -1 \ 0; \ SQ2 \ CQ2 \ 0 \ 0; \ 0 \ 0 \ 0 \ 1];$ T12 Matrisi oluşturulur.

$T23=[CQ3 \ -SQ3 \ 0 \ L; \ SQ3 \ CQ3 \ 0 \ 0; \ 0 \ 0 \ 1 \ 0; \ 0 \ 0 \ 0 \ 1];$ T23 Matrisi oluşturulur.

$T34=[CQ4 \ -SQ4 \ 0 \ L; \ SQ4 \ CQ4 \ 0 \ 0; \ 0 \ 0 \ 1 \ 0; \ 0 \ 0 \ 0 \ 1];$ T34 Matrisi oluşturulur.

$T45=[CQ5 \ -SQ5 \ 0 \ L; \ SQ5 \ CQ5 \ 0 \ 0; \ 0 \ 0 \ 1 \ 0; \ 0 \ 0 \ 0 \ 1];$ T45 Matrisi oluşturulur.

$T02=T01*T12;$ T02 Homojen Dönüşüm Matrisi bulunur.

$T03=T01*T12*T23;$ T03 Homojen Dönüşüm Matrisi bulunur.

$T04=T01*T12*T23*T34;$ T04 Homojen Dönüşüm Matrisi bulunur.

$T05=T01*T12*T23*T34*T45;$ T05 Homojen Dönüşüm Matrisi bulunur.

$X1=T01(1,4);$ T01 Homojen Dönüşüm Matrisine ait konum vektörünün X bileşeni X1 'e atanır

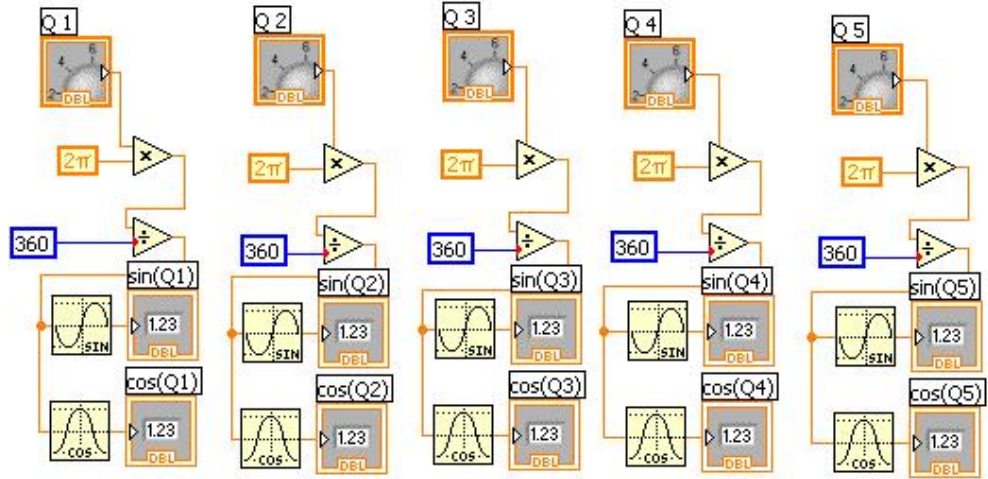
$Y1=T01(2,4);$ T01 Homojen Dönüşüm Matrisine ait konum vektörünün Y bileşeni Y1 'e atanır

$Z1=T01(3,4);$ T01 Homojen Dönüşüm Matrisine ait konum vektörünün Z bileşeni Z1 'e atanır

$X2=T02(1,4);$ T02 Homojen Dönüşüm Matrisine ait konum vektörünün X bileşeni X2 'e atanır

$Y2=T02(2,4);$ Y2 'e atanır	T02 Homojen Dönüşüm Matrisine ait konum vektörünün Y bileşeni
$Z2=T02(3,4);$ Z2 'e atanır	T02 Homojen Dönüşüm Matrisine ait konum vektörünün Z bileşeni
$X3=T03(1,4);$ X3 'e atanır	T03 Homojen Dönüşüm Matrisine ait konum vektörünün X bileşeni
$Y3=T03(2,4);$ Y3 'e atanır	T03 Homojen Dönüşüm Matrisine ait konum vektörünün Y bileşeni
$Z3=T03(3,4);$ Z3 'e atanır	T03 Homojen Dönüşüm Matrisine ait konum vektörünün Z bileşeni
$X4=T04(1,4);$ X4 'e atanır	T04 Homojen Dönüşüm Matrisine ait konum vektörünün X bileşeni
$Y4=T04(2,4);$ X4 'e atanır	T04 Homojen Dönüşüm Matrisine ait konum vektörünün Y bileşeni
$Z4=T04(3,4);$ X4 'e atanır	T04 Homojen Dönüşüm Matrisine ait konum vektörünün Z bileşeni
$X5=T05(1,4);$ X5 'e atanır	T05 Homojen Dönüşüm Matrisine ait konum vektörünün X bileşeni
$Y5=T05(2,4);$ X5 'e atanır	T05 Homojen Dönüşüm Matrisine ait konum vektörünün Y bileşeni
$Z5=T05(3,4);$ X5 'e atanır	T05 Homojen Dönüşüm Matrisine ait konum vektörünün Z bileşeni

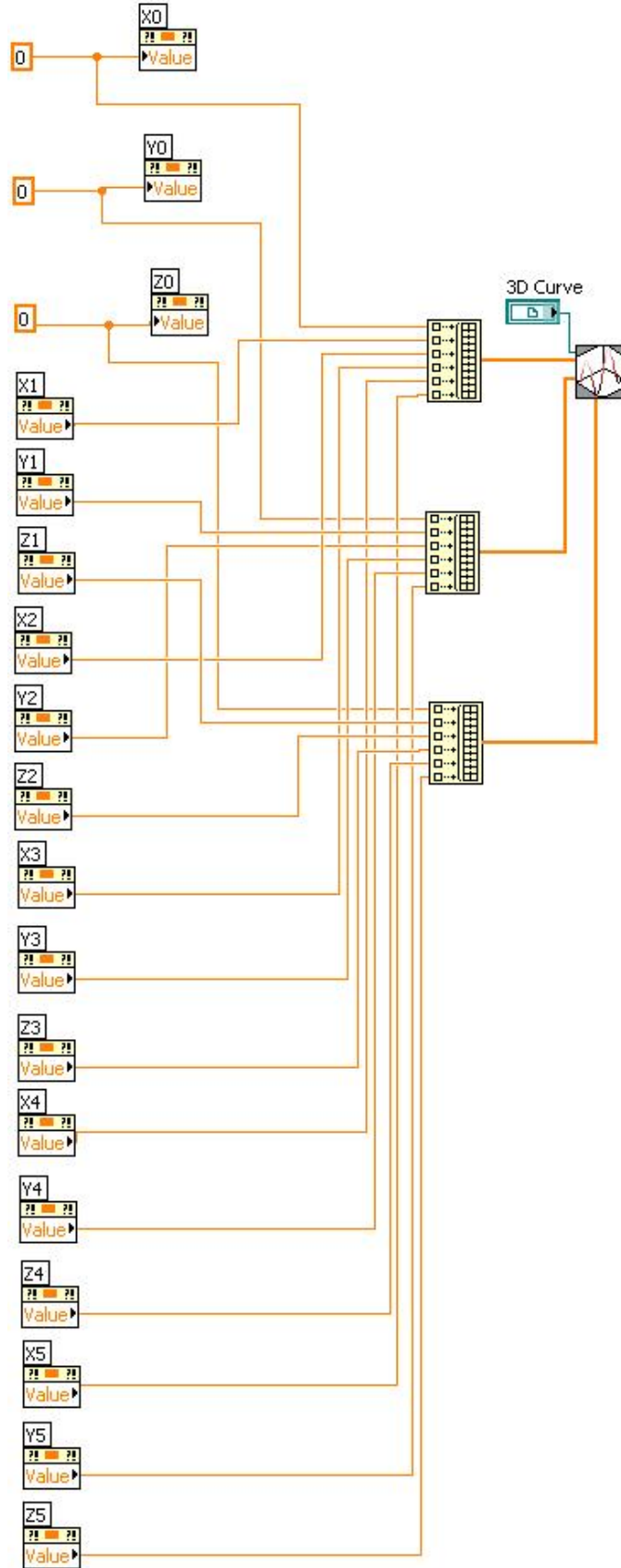
Bu komutlarda kullanılan H ve L eklem sabitleri ve SQ1,CQ1, SQ2,CQ2, SQ3,CQ3, SQ4,CQ4, SQ5,CQ5 giriş değişkenleri olup blok diyagramda aşağıdaki Şekil 3.119 'deki gibi blok diyagramda yer almaktadır.



Şekil 3.119 Giriş değişkenleri Blok diyagramı

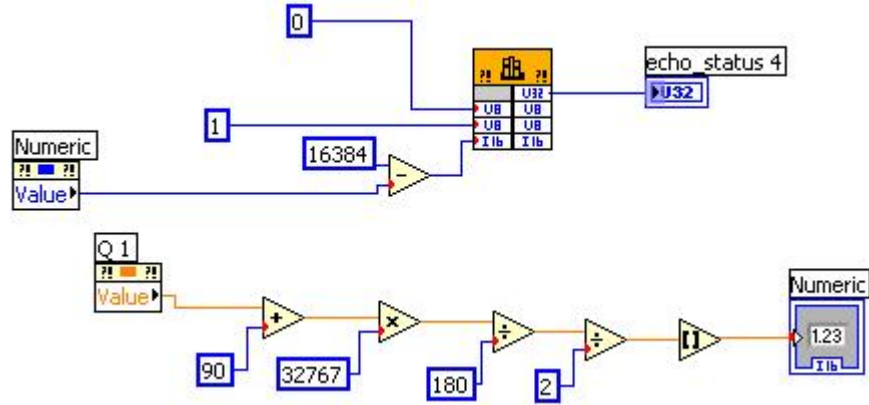
Q1, Q2, Q3, Q4 ve Q5 sanal potansiyometrelerinden alınan açı değerleri tek tek sin ve cos fonksiyonları yardımıyla SQ1,CQ1, SQ2,CQ2, SQ3,CQ3, SQ4,CQ4, SQ5,CQ5 giriş değişkenlerine dönüştürülür. Burada sin ve cos fonksiyonları derece cinsinden değil de radyan cinsinden açı değeri kullandığı için Q1, Q2, Q3, Q4 ve Q5 sanal potansiyometrelerinden alınan açı değerleri radyana dönüştürülerek kullanılır. Bakınız Şekil 3.119 'a.

MATLAB scriptine giren H ve L sabitleriyle, SQ1,CQ1, SQ2,CQ2, SQ3,CQ3, SQ4,CQ4, SQ5,CQ5 değişkenlerinden gelen bilgiye göre X0, Y0, Z0, X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, X4, Y4, Z4, X5, Y5, Z5 konum vektörleri bileşenleri elde edilir. Elde edilen bu konum vektörleri ile sanal potansiyometrelerden alınan eklem açı bilgisine göre Slave kolun anlık grafiği 3D CURVE nesnesiyle programda görüntülenmektedir. Aşağıdaki Şekil 3.120 'da 3D Curve nesnesinin blok diyagramdaki kullanımı görülmektedir.



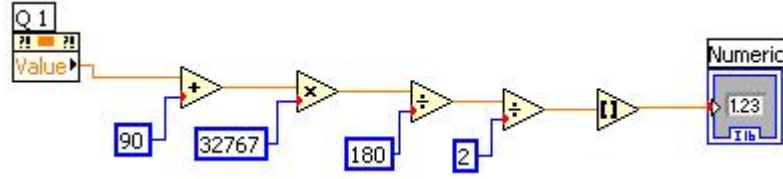
Şekil 3.120 3D Curve nesnesinin blok diyagramı

Q1, Q2, Q3, Q4 ve Q5 sanal potansiyometrelerinden alınan açı değerlerini servo motorların konum kontrolü için gerekli sinyali JAC3320 ADC kartıyla üretebilmek için JAC3320 'ye ait ve daha önce kullanımının açıklandığı Out_DA_data.vi blok diyagramındaki sürücü kullanılır. Out_DA_data.vi sürücüsünün 1. eklem için düzenlenmiş hali aşağıdaki şekilde görülmektedir.



Şekil 3.121 Out_DA_data.vi sürücüsünün 1. eklem için düzenlenmiş hali

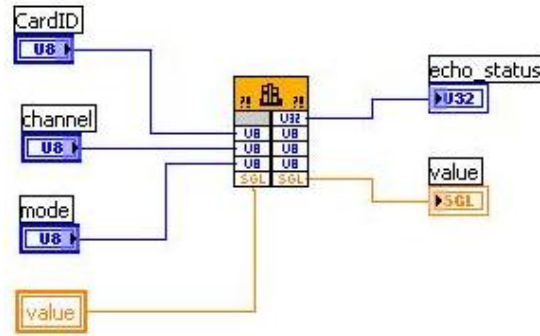
Burada Q1 sanal potansiyometresinden alınan -90 ile +90 aralığındaki açı değeri öncelikle 0-5V aralığındaki analog değere dönüştürülmeden önce sayısal bilgiye dönüştürülmelidir. Çünkü Out_DA_data.vi blok diyagramındaki sürücü sayısal bilgiyi analog bilgiye dönüştürmektedir. Bu nedenle -90 ile +90 arasındaki açı değeri sayısal olarak 0 ile +5V aralığındaki analog değere karşılık gelen 16384 ($32767/2$) sayısal değerine dönüştürülmelidir. Bu amaçla aşağıdaki Şekil 3.122 'deki blok diyagram kullanılarak Q1 potansiyometresinin -90 ile +90 Aralığındaki açı değeri 0 ile 16384 ($32767/2$) aralığındaki sayısal değere doğrusal olarak dönüştürülerek Numeric gösterge nesnesine atanır. Ve bu sayısal değer JAC3320 ADC kartından 0 ile +5V aralığındaki bir analog sinyale dönüştürülür. Bu analog sinyalde servo kontrol sürücü kartıyla 50Hz frekanslı modülasyon genişliği 0.5ms ile 2.5ms arasında değişen bir kontrol sinyaline dönüştürülür. Şekil 3.121 'deki Out_DA_data.vi blok diyagramında Numeric nesnesinin 16384 değerinden çıkarılması servo motor dönüş yönünün ters çevrilmesi için yapılmıştır. Q2, Q3, Q4 ve Q5 eklemlerinde de aynı yöntem kullanılarak ait olduğu eklemlerdeki servo motorlar sürülür.



Şekil 3.122 1 nolu eklem açısının Sayısal değere dönüştürülmesi

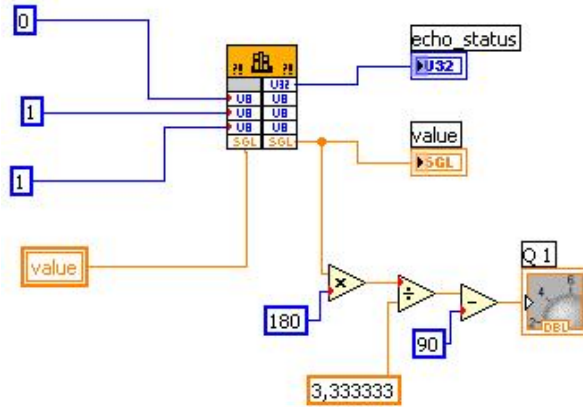
3.3.2.4.2 İleri Kinematik – Koldan Eklem Açılarını Al

Kontrol programının ikinci sekmesi olan “İleri Kinematik – Koldan Eklem Açılarını Al” sekmesinin bir önceki sekme olan “İleri Kinematik – Eklem Açılarını Ver” sekmesinden tek farkı eklem açılarının kontrol programındaki sanal potansiyometrelerle üretilmeyip Master kolun eklemlerine yerleştirilmiş olan gerçek potansiyometrelerden alınarak Slave kola hareket verilmesidir. Bu amaçla master kolun eklemlerine yerleştirilen potansiyometrelerden eklem açıları daha önce kullanımının açıklandığı smart_AtoD.vi blok diyagramındaki (Şekil 3.123) sürücü kullanılır.



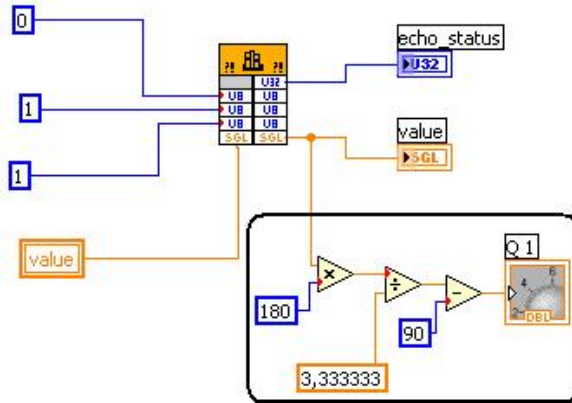
Şekil 3.123 smart_AtoD.vi Blok diyagramı

smart_AtoD.vi blok diyagramındaki sürücünün master kolun 1 nolu eklem açısını okumak için düzenlenmiş hali aşağıdaki Şekil 3.124 ‘deki gibidir.



Şekil 3.124 smart_AtoD.vi blok diyagramının 1 nolu eklem açısını okumak için düzenlenmiş hali

Burada JAC3320 ADC kartın 1, 2, 3, 4 ve 5 nolu kanalından okunan analog bilgi Q1, Q2, Q3, Q4 ve Q5 gösterge nesnelere atanır. Q1, Q2, Q3, Q4 ve Q5 nesnelere bir öncesi sekmede kontrol nesnesi iken bu sekmede gösterge nesnesidir. Q1, Q2, Q3, Q4 ve Q5 nesnelere Master kolun eklem açılarını göstereceği için okunan analog gerilim bilgisinin -90 ile +90 arasındaki bir açı değerine dönüştürülmesi gerekir. Bu amaçla öncelikle kartın kanaldan okuduğu analog bilgi Value değişkenine atanır ve Value değişkenine atanan 0 ile 3,3333 Volt arasında değişen gerilim bilgisi -90 ile +90 arasında değişen bir açı bilgisine doğrusal olarak aşağıdaki Şekil 3.125 'de dikdörtgen içine alınmış blok diyagram ile dönüştürülür.



Şekil 3.125 Gerilimin açı bilgisine dönüştürülmesi

Master kolun eklemeleri ve buna bağlı olarak eklemelerdeki potansiyometrelerin 180 derecelik açı sınırlaması bulunur. Ancak potansiyometrelerin fırça açısı 270 derecedir. Bu nedenle potansiyometrelere 5V uygulandığında 180 derecelik açı

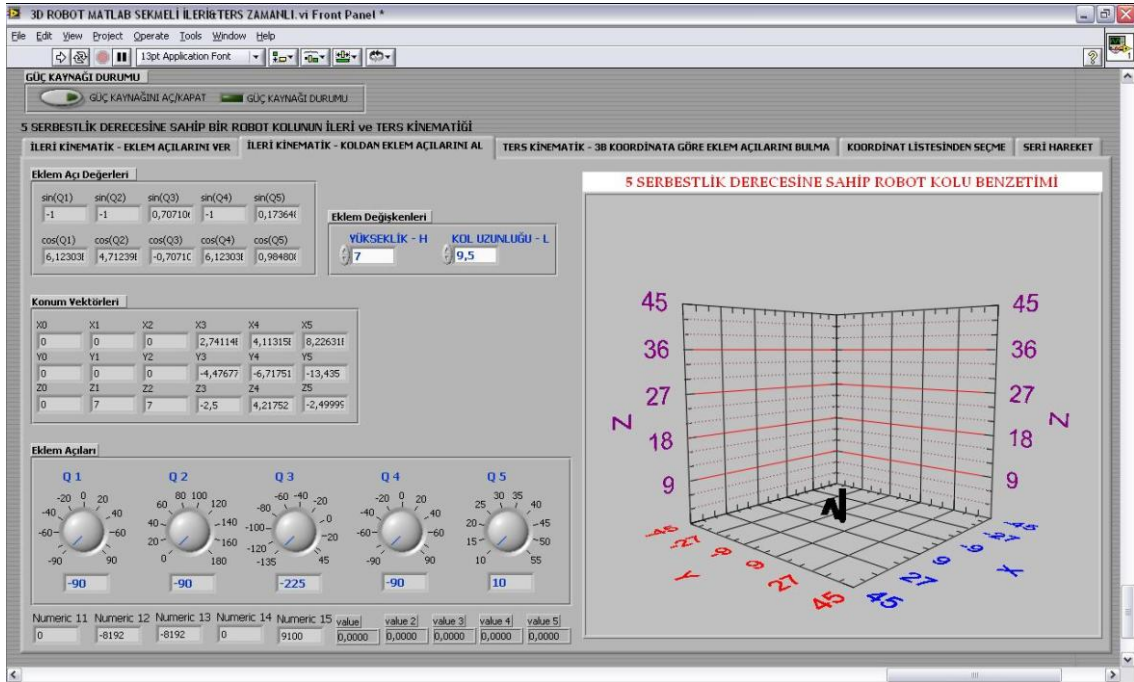
sınırlamasından dolayı eklemlerden alınan gerilim aralığı 0 ile 3.3333V aralığında olacaktır. Bu nedenle Q1, Q2, Q3 ve Q4 gösterge nesnelerinin skalaları 180 derece için 3.3333V ile kalibre edilmiştir. Sadece uç işlevcisinin açıklığını ayarlayan potansiyometrenin açı sınırlaması olmadığı için Q5 nesnesi 45 derece için 5V ile kalibre edilmiştir.



Şekil 3.126 Potansiyometredeki açı sınırlaması

Master kolun eklem açılarından elde edilen analog bilgi Q1, Q2, Q3, Q4 ve Q5 eklem açılarına dönüştürüldükten sonra ileri kinematik hesaplamaları ve slave kolun 3 boyutlu grafiği “İleri Kinematik – Eklem Açılarını Ver” sekmesindeki gibi yapılır.

İleri Kinematik – Eklem Açılarını Ver sekmesine ait kullanıcı ara yüzü Şekil 3.127 ‘da gösterilmiştir.

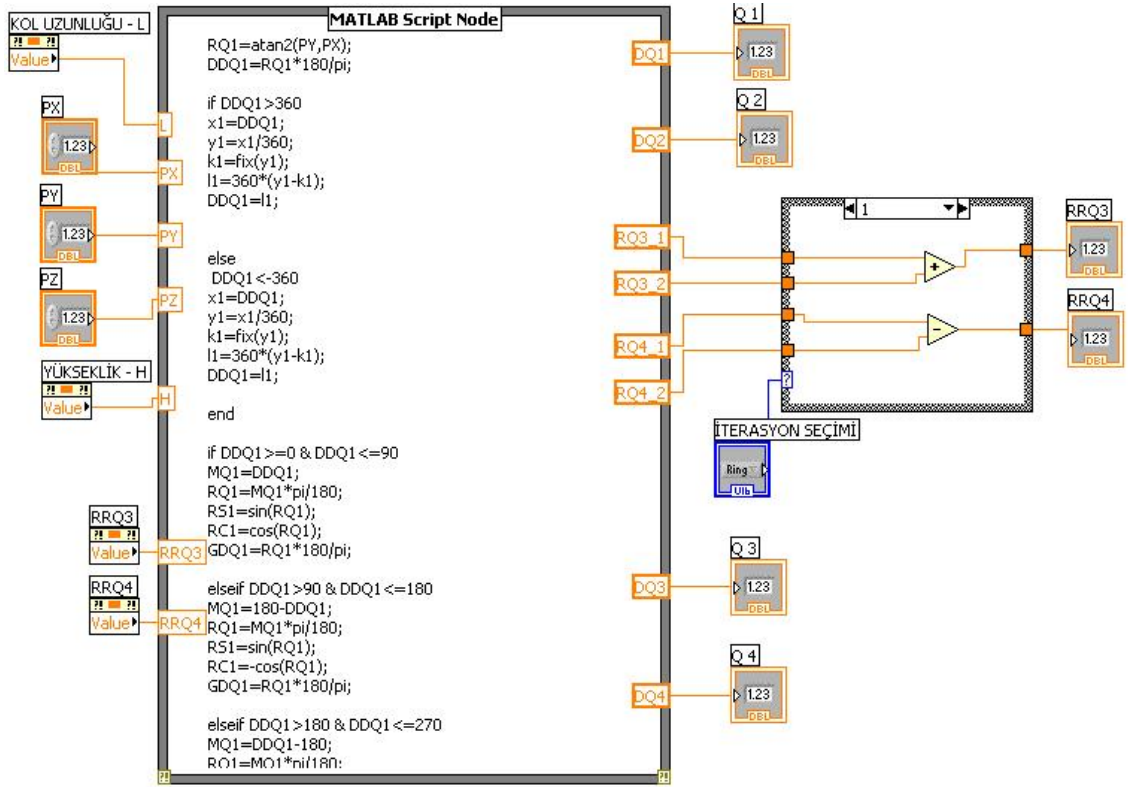


Şekil 3.127 İleri Kinematik – Eklem Açılarını Ver sekmesine ait kullanıcı arayüzü

3.3.2.4.3 Ters Kinematik – 3B Koordinata Göre Eklem Açılarını Bulma

Kontrol programının bu sekmesinde slave kolun uç işlevcisinin 3 boyutlu koordinat düzleminde bulunması istenilen noktanın koordinatları verilerek slave kolun muhtemel eklem açıları bulunur. Slave kolun ters kinematik modelinden elde edilen eşitlikler yardımıyla uç işlevcisinin bulunması istenilen noktanın koordinatına göre eklem açıları bulunur.

Ters kinematik eşitlikleri yardımıyla eklem açılarının bulunabilmesi için yine ileri kinematik hesaplamalarında olduğu gibi Şekil 3.128 'de görülen MATLAB scripti kullanılmıştır.



Şekil 3.128 Ters kinematik MATLAB scripti ve Blok diyagramı

MATLAB scripti içerisine aşağıdaki Ters kinematik eşitliklerden elde edilen komutlar yazılmıştır.

Bu komutlarla öncelikle elde edilen açı değerlerinin -360 dereceden küçük ve 360 dereceden büyük olma durumu kontrol edilir eğer açı değeri -360° ile $+360^\circ$ arasında değilse öncelikle açı -360° ile $+360^\circ$ aralığına getirilir. Daha sonra -360° ile $+360^\circ$ aralığındaki açı -90° ile $+90^\circ$ aralığına dönüştürülür. Böylece Slave kol eklemleri -90° ile $+90^\circ$ arasında hareketi sınırlandırılmış olur. Aksi halde slave kol kendine zarar verir.

```
RQ1=atan2(PY,PX); Q1 eklem açısı radyan cinsinden bulunur
```

```
DDQ1=RQ1*180/pi;
```

```
if DDQ1>360
```

```
  x1=DDQ1;
```

```
  y1=x1/360;
```

```
  k1=fix(y1);
```

```
  l1=360*(y1-k1);
```

```
  DDQ1=l1;
```

```
else
```

```
DDQ1<-360
```

```
  x1=DDQ1;
```

```
  y1=x1/360;
```

```
  k1=fix(y1);
```

```
  l1=360*(y1-k1);
```

```
  DDQ1=l1;
```

```
end
```

```
if DDQ1>=0 & DDQ1<=90
```

$MQ1=DDQ1;$

$RQ1=MQ1*\pi/180;$

$RS1=\sin(RQ1);$

$RC1=\cos(RQ1);$

$GDQ1=RQ1*180/\pi;$

elseif $DDQ1>90$ & $DDQ1\leq 180$

$MQ1=180-DDQ1;$

$RQ1=MQ1*\pi/180;$

$RS1=\sin(RQ1);$

$RC1=-\cos(RQ1);$

$GDQ1=RQ1*180/\pi;$

elseif $DDQ1>180$ & $DDQ1\leq 270$

$MQ1=DDQ1-180;$

$RQ1=MQ1*\pi/180;$

$RS1=-\sin(RQ1);$

$RC1=-\cos(RQ1);$

$GDQ1=RQ1*180/\pi;$

elseif $DDQ1>270$ & $DDQ1\leq 360$

$MQ1=360-DDQ1;$

$RQ1=MQ1*\pi/180;$

$RS1=-\sin(RQ1);$

```

RC1=cos(RQ1);

GDQ1=RQ1*180/pi;

elseif DDQ1>=-90 & DDQ1<0

MQ1=-DDQ1;

RQ1=MQ1*pi/180;

RS1=-sin(RQ1);

RC1=cos(RQ1);

GDQ1=-RQ1*180/pi;

elseif DDQ1>=-180 & DDQ1<-90

MQ1=180+DDQ1;

RQ1=MQ1*pi/180;

RS1=-sin(RQ1);

RC1=-cos(RQ1);

GDQ1=-RQ1*180/pi;

elseif DDQ1>=-270 & DDQ1<-180

MQ1=-DDQ1-180;

RQ1=MQ1*pi/180;

RS1=sin(RQ1);

RC1=-cos(RQ1);

GDQ1=-RQ1*180/pi;

else

```

DDQ1>=-360 & DDQ1<=-270

MQ1=360+DDQ1;

RQ1=MQ1*pi/180;

RS1=sin(RQ1);

RC1=cos(RQ1);

GDQ1=-RQ1*180/pi;

end

DQ1=MQ1;

S1=RS1;

C1=RC1;

RQ2=atan2((-S1*PX+C1*PZ),(C1*PX+S1*PZ)); Q2 eklem açısı bulunur

DDQ2=RQ2*180/pi;

if DDQ2>360

x2=DDQ2;

y2=x2/360;

k2=fix(y2);

l2=360*(y2-k2);

DDQ2=l2;

else

DDQ2<=-360

x2=DDQ2;

```
y2=x2/360;  
  
k2=fix(y2);  
  
l2=360*(y2-k2);  
  
DDQ2=l2;  
  
end  
  
if DDQ2>=0 & DDQ2<=90  
  
MQ2=DDQ2;  
  
RQ2=MQ2*pi/180;  
  
RS2=sin(RQ2);  
  
RC2=cos(RQ2);  
  
GDQ2=RQ2*180/pi;  
  
elseif DDQ2>90 & DDQ2<=180  
  
MQ2=180-DDQ2;  
  
RQ2=MQ2*pi/180;  
  
RS2=sin(RQ2);  
  
RC2=-cos(RQ2);  
  
GDQ2=RQ2*180/pi;  
  
elseif DDQ2>180 & DDQ2<=270  
  
MQ2=DDQ2-180;  
  
RQ2=MQ2*pi/180;  
  
RS2=-sin(RQ2);
```

$$RC2 = -\cos(RQ2);$$

$$GDQ2 = RQ2 * 180 / \pi;$$

$$\text{elseif } DDQ2 > 270 \ \& \ DDQ2 \leq 360$$

$$MQ2 = 360 - DDQ2;$$

$$RQ2 = MQ2 * \pi / 180;$$

$$RS2 = -\sin(RQ2);$$

$$RC2 = \cos(RQ2);$$

$$GDQ2 = RQ2 * 180 / \pi;$$

$$\text{elseif } DDQ2 \geq -90 \ \& \ DDQ2 < 0$$

$$MQ2 = -DDQ2;$$

$$RQ2 = MQ2 * \pi / 180;$$

$$RS2 = -\sin(RQ2);$$

$$RC2 = \cos(RQ2);$$

$$GDQ2 = -RQ2 * 180 / \pi;$$

$$\text{elseif } DDQ2 \geq -180 \ \& \ DDQ2 < -90$$

$$MQ2 = 180 + DDQ2;$$

$$RQ2 = MQ2 * \pi / 180;$$

$$RS2 = -\sin(RQ2);$$

$$RC2 = -\cos(RQ2);$$

$$GDQ2 = -RQ2 * 180 / \pi;$$

```
elseif DDQ2>=-270 & DDQ2<-180
```

```
MQ2=-DDQ2-180;
```

```
RQ2=MQ2*pi/180;
```

```
RS2=sin(RQ2);
```

```
RC2=-cos(RQ2);
```

```
GDQ2=-RQ2*180/pi;
```

```
else
```

```
DDQ2>=-360 & DDQ2<-270
```

```
MQ2=360+DDQ2;
```

```
RQ2=MQ2*pi/180;
```

```
RS2=sin(RQ2);
```

```
RC2=cos(RQ2);
```

```
GDQ2=-RQ2*180/pi;
```

```
end
```

```
DQ2=MQ2;
```

```
S2=RS2;
```

```
C2=RC2;
```

```
A1=PX*(-S1*C2-C1*S2)+L*(S1*C2+C1*S2);
```

```
B1=PY*(-S1*C2-C1*S2);
```

```
C11=-PZ*(-S1*S2+C1*C2);
```

```
RQ3_1=(pi/2)-atan2(B1,A1); Q3 eklem açısının birinci bileşeni bulunur
```

$$DQ3_1=RQ3_1*180/\pi;$$

$RQ3_2=(\pi/2)-\text{atan2}(\sqrt{A1^2+B1^2-C11^2}),C11)$; Q3 eklem açısının ikinci bileşeni bulunur

$$DQ3_2=RQ3_2*180/\pi;$$

$$RRQ3=RQ3_1+RQ3_2;$$

$$DDQ3=(RRQ3*180/\pi);$$

if $DDQ3 > 360$

$$x3=DDQ3;$$

$$y3=x3/360;$$

$$k3=\text{fix}(y3);$$

$$l3=360*(y3-k3);$$

$$DDQ3=l3;$$

else

$$DDQ3 < -360$$

$$x3=DDQ3;$$

$$y3=x3/360;$$

$$k3=\text{fix}(y3);$$

$$l3=360*(y3-k3);$$

$$DDQ3=l3;$$

end

if $DDQ3 \geq 0$ & $DDQ3 \leq 90$

MQ3=DDQ3;

RQ3=MQ3*pi/180;

RS3=sin(RQ3);

RC3=cos(RQ3);

GDQ3=RQ3*180/pi;

elseif DDQ3>90 & DDQ3<=180

MQ3=180-DDQ3;

RQ3=MQ3*pi/180;

RS3=sin(RQ3);

RC3=-cos(RQ3);

GDQ3=RQ3*180/pi;

elseif DDQ3>180 & DDQ3<=370

MQ3=DDQ3-180;

RQ3=MQ3*pi/180;

RS3=-sin(RQ3);

RC3=-cos(RQ3);

GDQ3=RQ3*180/pi;

elseif DDQ3>370 & DDQ3<=360

MQ3=360-DDQ3;

RQ3=MQ3*pi/180;

RS3=-sin(RQ3);

```

RC3=cos(RQ3);

GDQ3=RQ3*180/pi;

elseif DDQ3>=-90 & DDQ3<0

MQ3=-DDQ3;

RQ3=MQ3*pi/180;

RS3=-sin(RQ3);

RC3=cos(RQ3);

GDQ3=-RQ3*180/pi;

elseif DDQ3>=-180 & DDQ3<-90

MQ3=180+DDQ3;

RQ3=MQ3*pi/180;

RSi=-sin(RQ3);

RCi=-cos(RQ3);

GDQ3=-RQ3*180/pi;

elseif DDQ3>=-370 & DDQ3<-180

MQ3=-DDQ3-180;

RQ3=MQ3*pi/180;

RS3=sin(RQ3);

RC3=-cos(RQ3);

GDQ3=-RQ3*180/pi;

else

```

$$DDQ3 \geq -360 \ \& \ DDQ3 < -370$$

$$MQ3 = 360 + DDQ3;$$

$$RQ3 = MQ3 * \pi / 180;$$

$$RS3 = \sin(RQ3);$$

$$RC3 = \cos(RQ3);$$

$$GDQ3 = -RQ3 * 180 / \pi;$$

end

$$DQ3 = MQ3;$$

$$S3 = RS3;$$

$$C3 = RC3;$$

$$A3 = (C3 * PX * (-S1 * C2 - C1 * S2) + PY * S3 * (-S1 * C2 - C1 * S2) - L * C3 * (-S1 * C2 - C1 * S2));$$

$$B3 = (-S3 * PX * (-S1 * C2 - C1 * S2) + PY * C3 * (-S1 * C2 - C1 * S2) + L * S3 * (-S1 * C2 - C1 * S2));$$

$$C33 = -(PZ * (-S1 * S2 + C1 * C2) - L * C3 * (-S1 * C2 - C1 * S2));$$

$$RQ4_1 = (\pi/2) - \text{atan2}(B3, A3); \text{ Q4 eklem açısının birinci bileşeni bulunur}$$

$$DQ4_1 = RQ4_1 * 180 / \pi;$$

$$RQ4_2 = (\pi/2) - \text{atan2}(\sqrt{A3^2 + B3^2 - C33^2}, C33); \text{ Q4 eklem açısının ikinci bileşeni bulunur}$$

$$DQ4_2 = RQ4_2 * 180 / \pi;$$

$$RRQ4 = RQ4_1 + RQ4_2;$$

$$DDQ4 = (RRQ4 * 180 / \pi);$$

```
if DDQ4>360

x4=DDQ4;

y4=x4/360;

k4=fix(y4);

l4=360*(y4-k4);

DDQ4=l4;

else

DDQ4<-360

x4=DDQ4;

y4=x4/360;

k4=fix(y4);

l4=360*(y4-k4);

DDQ4=l4;

end

if DDQ4>=0 & DDQ4<=90

MQ4=DDQ4;

RQ4=MQ4*pi/180;

RS4=sin(RQ4);

RC4=cos(RQ4);

GDQ4=RQ4*180/pi;

elseif DDQ4>90 & DDQ4<=180
```

$MQ4=180-DDQ4;$

$RQ4=MQ4*\pi/180;$

$RS4=\sin(RQ4);$

$RC4=-\cos(RQ4);$

$GDQ4=RQ4*180/\pi;$

elseif $DDQ4>180 \ \& \ DDQ4\leq 270$

$MQ4=DDQ4-180;$

$RQ4=MQ4*\pi/180;$

$RS4=-\sin(RQ4);$

$RC4=-\cos(RQ4);$

$GDQ4=RQ4*180/\pi;$

elseif $DDQ4>270 \ \& \ DDQ4\leq 360$

$MQ4=360-DDQ4;$

$RQ4=MQ4*\pi/180;$

$RS4=-\sin(RQ4);$

$RC4=\cos(RQ4);$

$GDQ4=RQ4*180/\pi;$

elseif $DDQ4\geq -90 \ \& \ DDQ4<0$

$MQ4=-DDQ4;$

$RQ4=MQ4*\pi/180;$

$RS4=-\sin(RQ4);$

```

RC4=cos(RQ4);

GDQ4=-RQ4*180/pi;

elseif DDQ4>=-180 & DDQ4<-90

MQ4=180+DDQ4;

RQ4=MQ4*pi/180;

RS4=-sin(RQ4);

RC4=-cos(RQ4);

GDQ4=-RQ4*180/pi;

elseif DDQ4>=-270 & DDQ4<-180

MQ4=-DDQ4-180;

RQ4=MQ4*pi/180;

RS4=sin(RQ4);

RC4=-cos(RQ4);

GDQ4=-RQ4*180/pi;

else

DDQ4>=-360 & DDQ4<-270

MQ4=360+DDQ4;

RQ4=MQ4*pi/180;

RS4=sin(RQ4);

RC4=cos(RQ4);

GDQ4=-RQ4*180/pi;

```

end

DQ4=MQ4;

S4=RS4;

C4=RC4;

MATLAB scripti ile üretilen DQ1, DQ2, DQ3 ve DQ4 açıları slave kolun eklemlerinin olması gereken açı değeridir. Bu eklem açılarına göre eklemlerin konum vektörleri, slave kolun anlık grafiği ve slave kolun eklemlerinde bulunan servo motorların konumunu belirten analog sinyaller kontrol programının birinci sekmesinde olduğu gibi bulunur. Sadece slave kol uç işlevcisini açıp/kapatmak için bu sekmeye Q5 sanal potansiyometresi eklenmiştir. Uç işlevcisinin açıklığı Şekil 3.129 'deki sanal potansiyometre ile kontrol edilir.



Şekil 3.129 Uç işlevcisi açıklığını kontrol eden potansiyometre

Program üzerinde uç işlevcisinin 3 boyutlu düzlemdeki koordinatlarının girildiği kısım Şekil 3.130 'da gösterilmiştir.



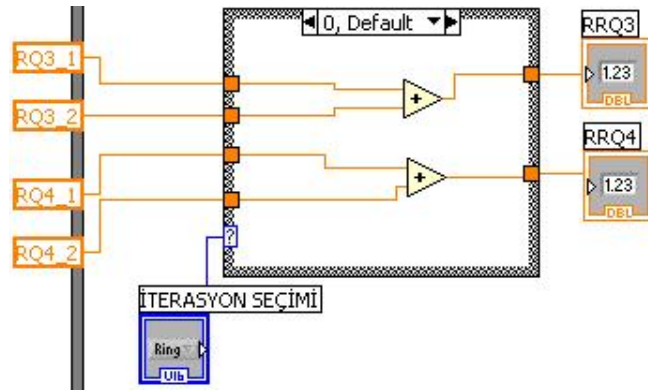
Şekil 3.130 Uç işlevcisinin X,Y,Z koordinat girişi

Ters kinematik eşitliklerinden elde edilen eklem açılarının değeri bilgi amaçlı olarak gösterildiği kısım programda Şekil 3.131 'da gösterilmiştir. Buradaki Q5 açısı değeri uç işlevcisinin açıklığını göstermektedir.



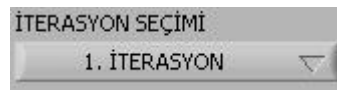
Şekil 3.131 Bulunan Eklem Açıları

Ters kinematik eşitliklerinde 3. ve 4. eşitliklerde bulunan $-/+$ lıklara göre 4 tane iterasyon oluşmaktadır. Şekil 3.132 'deki blok diyagramda görüldüğü gibi Birinci iterasyonda RQ3_1 ile RQ3_2 toplanıp RQ3, RQ4_1 ile RQ4_2 toplanıp RQ4 elde edilir. İkinci iterasyonda RQ3_1 ile RQ3_2 toplanıp RQ3, RQ4_1 ile RQ4_2 farkından RQ4 elde edilir. Üçüncü iterasyonda RQ3_1 ile RQ3_2 farkından RQ3, RQ4_1 ile RQ4_2 toplanıp RQ4 elde edilir. Dördüncü iterasyonda RQ3_1 ile RQ3_2 farkından RQ3, RQ4_1 ile RQ4_2 farkından RQ4 elde edilir. İterasyonlar eklem açılarının bulunmasında etkilidir.



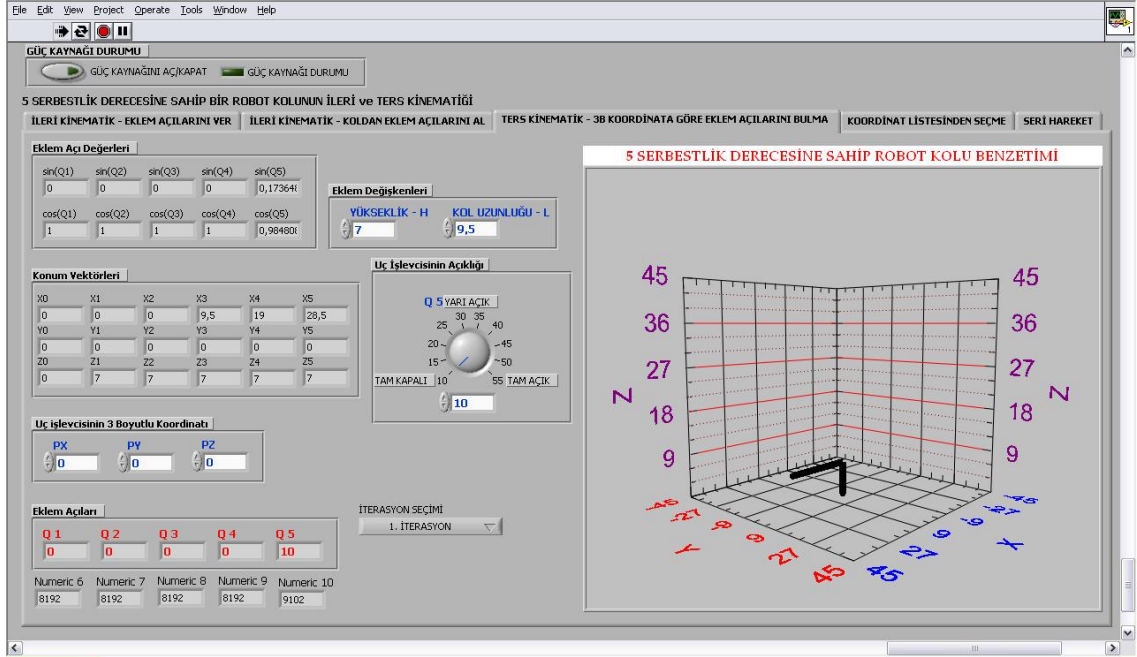
Şekil 3.132 İterasyon Blok diyagramı

Bu iterasyonlar programda Şekil 3.133 'deki seçim listesi ile belirlenebilir.



Şekil 3.133 İterasyon seçim listesi

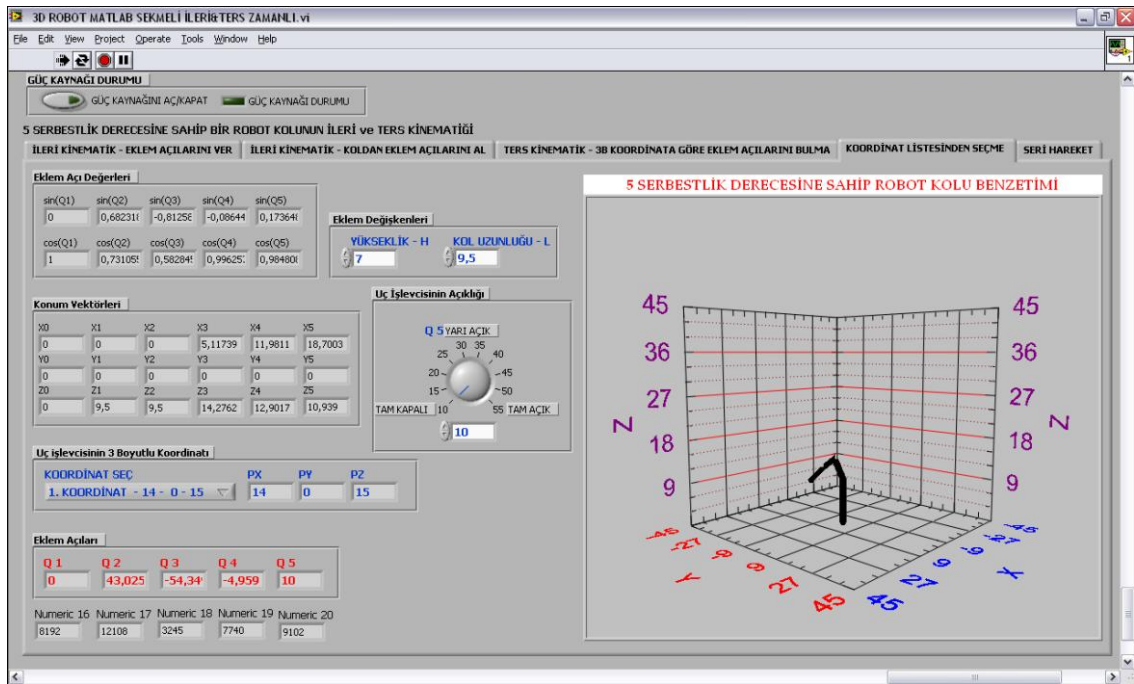
Aşağıdaki şekilde “Ters Kinematik – 3B Koordinata Göre Eklem Açılarını Bulma” sekmesinin kullanıcı arayüzü görülmektedir.



Şekil 3.134 3B Koordinata Göre Eklem Açılarını Bulma sekmesinin kullanıcı arayüzü

3.3.2.4.4 Ters Kinematik – Koordinat Listesinden Seçme

Bu sekmede “Ters Kinematik – 3B Koordinata Göre Eklem Açılarını Bulma” sekmesiyle aynı olmakla birlikte uç işlevcisinin üç boyutlu koordinat sisteminde olması istenen noktanın X, Y ve Z koordinatlarını tek tek girilmesi yerine önceden hazırlanmış bir listeden seçilerek eklem açılarının bulunması sağlanır. Tüm hesaplamalar “Ters Kinematik – 3B Koordinata Göre Eklem Açılarını Bulma” sekmesiyle aynıdır. Bu sekmenin kullanıcı arayüzü Şekil 3.135 ‘de görülmektedir.

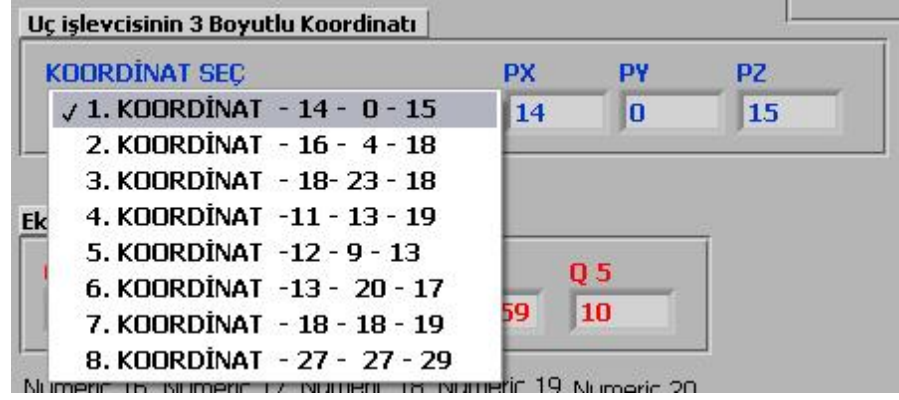


Şekil 3.135 Koordinat Listesinden Seçme sekmesine ait kullanıcı arayüzü

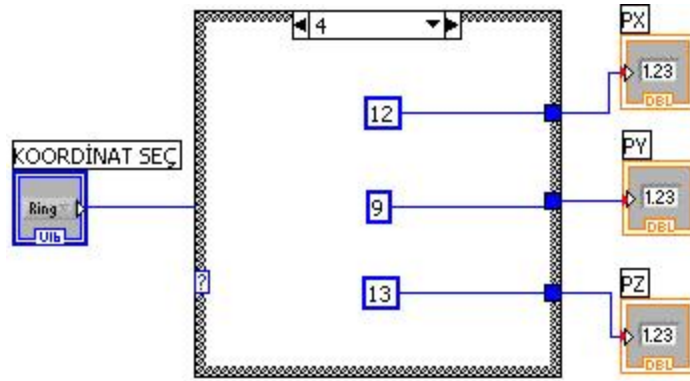
Uç işlevcisinin koordinatı programa eklenen Şekil 3.136 ve Şekil 3.137 ‘daki koordinat listesinden seçilir ve koordinat listesinin blok diyagramı Şekil 3.138 ‘de görülmektedir.



Şekil 3.136 Koordinat Seçim butonu



Şekil 3.137 Koordinat Seçim listesi



Şekil 3.138 Koordinat seçim blok diyagramı

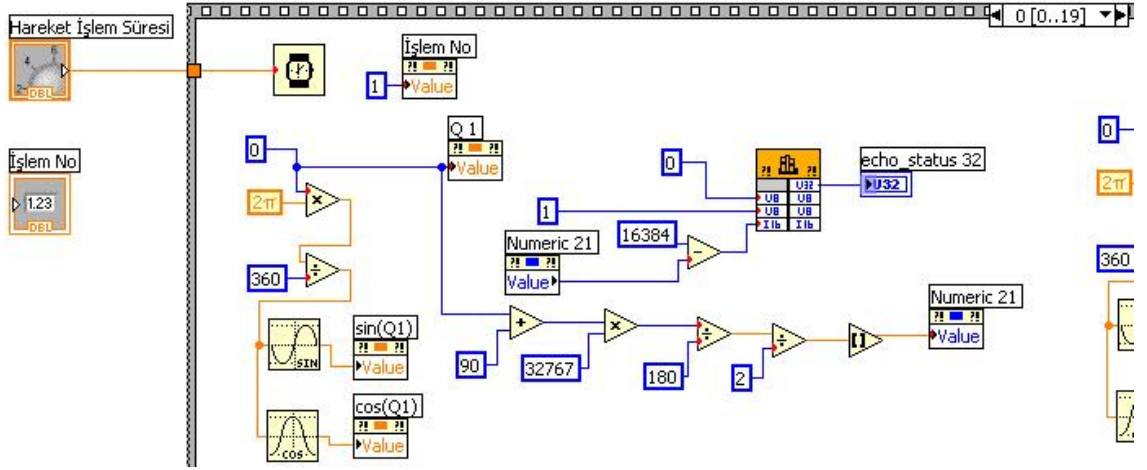
3.3.2.4.5 Seri Hareket

“Seri Hareket” sekmesi “İleri Kinematik – Eklem Açılarını Ver” sekmesiyle tamamen aynı olup programın blok diyagramına eklenen Stacked Sequence[14] karar yapısıyla önceden belirlenmiş eklem açıları belirlenen işlem süresi boyunca periyodik olarak slave kola hareket verir. Stacked Sequence yapısına 20 tane frame (işlem) yani Slave kolun eklemlerine 20 farklı Q1, Q2, Q3, Q4 ve Q5 açı değerleri verilmiş ve Kolun periyodik olarak bu açı setlerine göre hareket etmesi sağlanır. Bu açı değerleri aşağıdaki tabloda verilmiştir.

İşlem/Açı	Q1	Q2	Q3	Q4	Q5
1	0	0	0	0	10
2	0	0	0	0	55
3	0	45	0	0	10
4	0	45	45	0	10
5	0	45	45	-45	10
6	0	45	45	-45	55
7	0	45	-90	90	10
8	0	90	-90	0	10
9	0	90	-90	0	55
10	0	135	-135	0	10
11	0	180	-135	-90	10
12	-45	180	-135	-90	10
13	45	180	-135	-90	10
14	-45	90	0	0	10
15	-90	90	0	0	10
16	-90	90	0	0	55
17	45	90	0	0	10
18	45	90	0	0	55
19	90	90	0	0	10
20	90	90	0	0	55

Tablo 3.15 Seri hareket eklem açı tablosu

Stacked Sequence karar yapısının her işlem yapısına eklenen Time Wait fonksiyonuyla işlemlerin tamamlanma süresi ayarlanabilir. Şekil 3.139 ‘de Seri Hareket sekmesinin blok diyagramındaki Time Wait fonksiyonu görülmektedir.



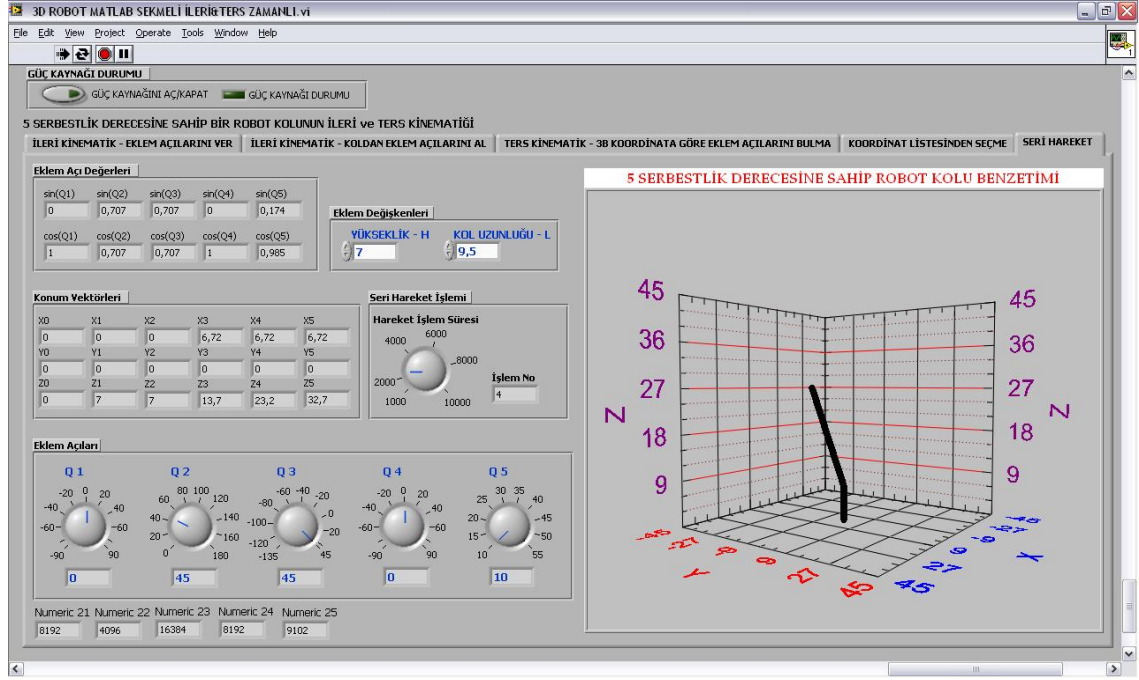
Şekil 3.139 Seri Hareket sekmesinin blok diyagramındaki Time Wait fonksiyonu

Programa eklenen Şekil 3.140 'daki Hareket İşlem Süresi kontrol nesnesiyle işlem süresi 1000ms ile 10000ms arasında 500ms aralıklarla arttırılıp azaltılabilir. Ayrıca slave kolun hangi işlemde olduğu Şekil 3.140 'da İşlem No gösterge nesnesiyle programda bilgi amaçlı olarak gösterilmektedir.



Şekil 3.140 Seri Hareket işlem süresi ve numarası

Aşağıdaki Şekil 3.141 'de Seri hareket Sekmensinin kullanıcı arayüzü görülmektedir.



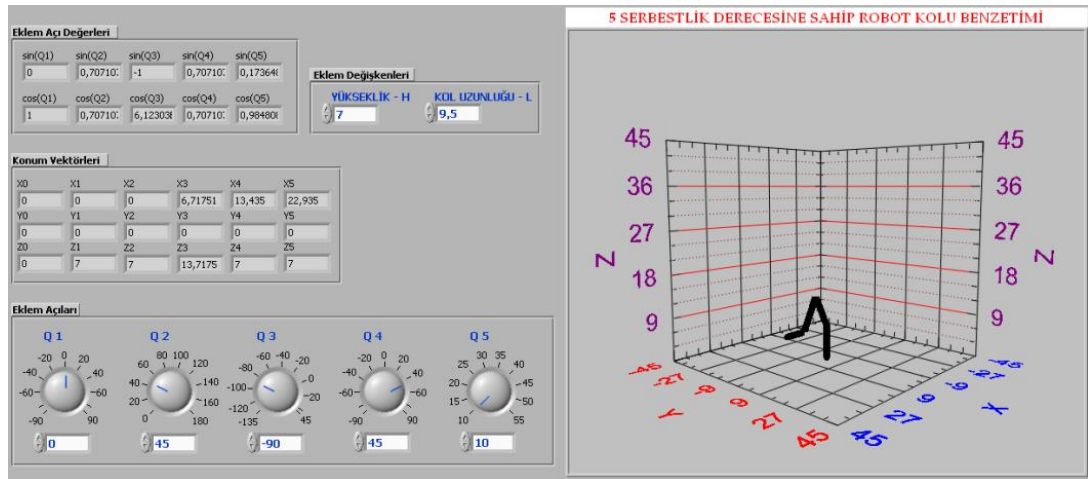
Şekil 3.141 Seri Hareket sekmensinin kullanıcı arayüzü

4 HEDEFLENEN SONUÇLAR VE ÖNERİLER

4.1 Hedeflenen Sonuçlar

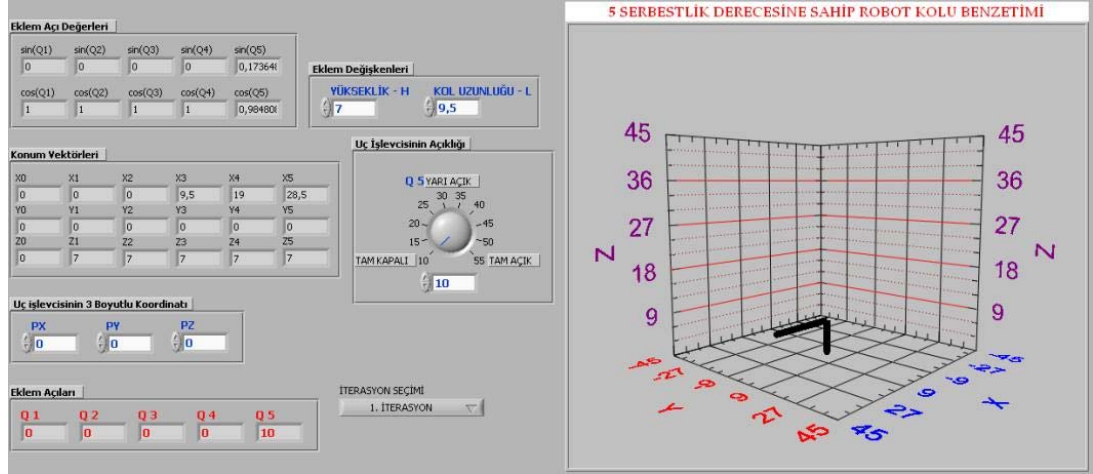
Bu çalışmada Slave kolun master kol veya kontrol programı aracılığıyla kontrol edilmesi amaçlanmış ve bu amaca ulaşılmıştır.

- Kontrol programına yerleştirilmiş Q1, Q2, Q3 ve Q4 sanal potansiyometreleriyle üretilen eklem açılarına bağlı anlık olarak her eklemin ve uç işlevcisinin konum vektörleri hatasız olarak bulunduğu program üzerine yerleştirilmiş konum vektörleri matrisinden kontrol edilmiştir. Ayrıca slave kolun uç işlevcisinin ve diğer eklemlerin eklem açılarına göre olması gereken konumlarını üç boyutlu olarak veren grafikte ve gerçek hareketi ile doğrulanmıştır. Uç işlevcisinin açıklığı programa yerleştirilen Q5 sanal potansiyometresiyle sorunsuz olarak kontrol edildiği görülmüştür. Şekil 4.1



Şekil 4.1 Eklem açılarına göre konum vektörleri ve üç boyutlu grafiği

- Kontrol programında uç işlevcisinin olması istenen üç boyutlu x,y,z koordinatları verildiğinde eklem açıları olması gereken açı değerlerine eşit veya yakın değerler aldığı program üzerine yerleştirilmiş konum vektörleri matrisi ve üç boyutlu grafikten kontrol edilmiştir. Şekil 4.2



Şekil 4.2 Üç boyutlu koordinatlara göre eklem açıları ve üç boyutlu grafiği

- Master koldan alınan analog forma dönüştürülmüş eklem açıları kontrol programı üzerinde sayısal formda hatasız olarak görüntülenmiş, her eklem için konum vektörleri hesaplanarak yine konum vektörleri matrisinde gösterilmiştir. Master kolun eklem açıları tekrar analog forma dönüştürülerek slave kolun kontrolü için Servo motor kontrol kartı ve ona bağlı olan Slave kol hatasız olarak kontrol edilmiştir.

4.2 Öneriler

Bu çalışma bir robot kolunun yine bir başka robot kolu veya bu uygulama için özel olarak tasarlanmış kontrol programıyla kontrol edilmesi amaçlanmıştır. Bu amaçla bu konu alanı için bir ön çalışma niteliğinde olan bu çalışma sistem elemanları değiştirilip geliştirilerek ve geliştirilebilecek başka kontrol yöntemleri ile hareket yeteneği ve etkinliği artırılabilir.

5 TEZ SIRASINDA YAPILAN ÇALIŞMALAR

5.1 Uluslararası Yayınlar

ÇETİNKAYA Ö., KUŞÇU H., "Classification of the Industrial Robot Arms", International Scientific Conference AMTECH'07 Gabrovo, Advanced Manufacturing Technologies Proceedings, Volume-II, ISBN 978-954-683-383-9, pp.I_126-I_131, 23-24 November, Bulgaria, 2007

ÇETİNKAYA Ö., KUŞÇU H., "Position Controls of 4 Servo Motors With Potentiometer, by Using Microcontroller", International Scientific Conference UNITECH'08 Gabrovo, Proceedings Volume-I, ISSN 1313-230X, pp.I_519-I_524, 21-22 November, Bulgaria, 2008

5.2 Ulusal Yayınlar

Ulusal yayın yoktur.

KAYNAKLAR

- [1] YALÇIN K., “Robotlar”, Elektronik Kitap, 2003, www.electrobotic.org/dkm/rbt1.pdf, 26.10.2008
- [2] TÜRK CAD, “Robot Tarihi”, <http://www.turk-cad.com/content/view/920/218/> , 26.10.2008
- [3] BİNGÜL,Z., KÜÇÜK,S., 2005, “Robot Tekniği” Birsen Yayınevi, İstanbul, 6, 13-15, 17-20, 21-22, 25, 41-42, 50-51, 53, 55-58, 65, 73-75, 105, 132-134
- [4] YANG, A. T., FREUDENSTEIN, F., 1964, “Application of Dual Number Quaternions Algebra to the Analysis of Spatial Mechanism”, Tran. ASME, J. Appl. Mechanics, Vol. 31, 300–3008
- [5] DENAVIT, T., HARTENBERG, R. S., 1955, “A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices”, ASME J. Appl. Mechanics, 215–221
- [6] BİNGÜL,Z., KÜÇÜK,S., 2005, “The Inverse Kinematics Solutions of Fundamental Robot Manipulators with Offset Wrist”, IEEE ICM/HIMA, Taipei, Taiwan
- [7] WOLOVICH, W. A., 1987, “Robotics: Basic Analysis and Design”, CBS College Publishing, USA
- [8] KÜÇÜK, S., 2004 , “Endüstriyel Robotların Modellenmesi ve Çevrimdışı programlanması”, Doktora Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü
- [9] CRAIG, J. J., 1989, “Introduction to Robotics:Mechanics and Control”, USA
- [10] MICROCHIP, “PIC16F877 Data sheet”, <http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf>, 01.06.2008
- [11] ÇETİNKAYA, Ö., KUŞÇU, H., “Position Controls of 4 Servo Motors With Potentiometer, by Using Microcontroller”, International Scientific Conference UNITECH'08 Gabrovo, Proceedings Volume-I, ISSN 1313-230X, pp.I_519-I_524, 21-22 November, Bulgaria, 2008

[12] JAC AIO3320 's Hardware Data sheet,
http://ftp.automation.com.tw/download/Manual_Driver/AioCard/AIO3320_1/manual/aio3320e.pdf, 08.05.2008

[13] JAC AIO3320 's LabVIEW Driver,
http://ftp.automation.com.tw/download/Manual_Driver/labview_driver/labview_driver.exe, 08.05.2008

[14] ÜNSAÇAR, F., EŞME, E., 2007, “ Grafik Programlama Dili LabVIEW”, Seçkin Yayıncılık, Ankara, 13-17, 31-32, 175-178

ÖZGEÇMİŞ

Özcan ÇETİNKAYA 1982 yılında Kars 'ta doğdu. İlkokul ve Ortaokul 'u Babaeski 'de, Liseyi Kırklareli Teknik Lisesinde okul birincisi derecesiyle tamamladı. 2000 yılında girdiği Marmara Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi bölümünden 2004 yılında mezun oldu. 2005 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü Makina Mühendisliği Anabilim Dalında Yüksek Lisans Eğitimine başladı. 2006–2008 yılları arasında Tem Mobilya San. ve Tic. A.Ş. Babaeski Fabrikasında Elektrik-Makina Bakım ve Onarım Şefi olarak çalıştı. 2008 yılından beri Trakya Üniversitesi Keşan Meslek Yüksekokulu Elektrik Programında Öğretim Görevlisi olarak çalışmaktadır.