

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**PARALEL ROBOT KİNEMATİĞİ, YÖRÜNGE PLANLAMASI
VE 3-UPS, 1-RU PARALEL ROBOT UYGULAMASI**

İLYAS DİRİN

KOCAELİ 2014

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

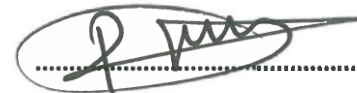
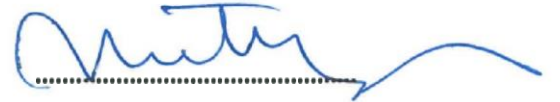
PARALEL ROBOT KİNEMATİĞİ, YÖRÜNGE PLANLAMASI
VE 3-UPS, 1-RU PARALEL ROBOT UYGULAMASI

İLYAS DİRİN

Yrd.Doç.Dr. Metin AYDIN
Danışman, Kocaeli Üniv.

Prof.Dr. Zafer BİNGÜL
Jüri Üyesi, Kocaeli Üniv.

Prof.Dr. Raşit KÖKER
Jüri Üyesi, Sakarya Üniv.



Tezin Savunulduğu Tarih: 11.04.2014

ÖNSÖZ ve TEŞEKKÜRLER

Son yıllarda paralel robotlar üzerine yapılan çalışmalar bu mekanizmaların endüstrideki kullanım alanlarını arttırmış ve endüstriyel alanda bu mekanizmalar önem kazanmıştır. Yüksek kinematik hassasiyetleri, kararlılıkları ve hassas pozisyon kontrolü gibi özellikleri bu mekanizmaları tercih sebebi yapmış ve günümüzde tıp, askeri uygulamalar, lojistik, eğitim, otomotiv gibi birçok alanda yaygın şekilde kullanılmaya başlanmıştır. Bu tez çalışması kapsamında paralel robotların seri robotlara göre avantajları ve dezavantajları değerlendirilmiş ve kinematik analizi yapılmıştır. Robotun hareketinin, kinematik çözümlerine göre yörünge planlaması sağlanmıştır. Elde edilen teorik çözümler kontrol algoritmaları ile desteklenerek 3-UPS, 1-RU yapısında bir paralel robot tasarımı ile birlikte kullanılmıştır.

Yüksek lisans tezimde bana verdiği destek ve yardımları için değerli danışman hocam Sn. Yrd. Doç. Dr. Metin AYDIN'a, akademik bilgi birikimini aktaran ve bu tezi yazabilmemdeki teorik bilgi donanımına sahip olmamı sağlayan değerli hocam Sn. Doç. Dr. Serdar KÜÇÜK'e, tüm çalışmam boyunca beni dinleyip fikirlerime katkıda bulunan ve mühendislik çözümleriyle çalışmamı bir adım öteye taşıyan değerli iş arkadaşım İdris HACIVELİOĞLU'na, yoğun tempolu acil ve önemli işlerin hiç bitmediği şirket ortamında derslerime devam edebilmem için her türlü özveriye ve anlayışı gösteren değerli yöneticilerim Mehmet SEVİM'e, Dursun AL'a, Gürcan KARAYEL'e ve eski yöneticim Murat DERELİ'ye, çalışmamı ezgilerini dinleyerek tamamladığım değerli bağlama hocam Sn. Erdal ERZİNCAN'a, farklı çalışma teknikleri ile hayatın her alanında kullanılabilecek farklı disiplinler öğrendiğim değerli bağlama hocam Sn. Ümit ŞİMŞEK'e, her türlü akademik süreci danıştığım değerli arkadaşım Arş. Gör. Mehmet GÜLEÇ'e, fikirlerime katkıda bulunan ve geliştiren arkadaşlarım Mahmut BAŞ ve Gürsel ODA'ya, beni her konuda destekleyen yardımlarını ve yorumlarını eksik etmeyen değerli arkadaşım Pınar TANRIVERDİ'ye içten teşekkürlerimi sunarım.

Ayrıca bugüne kadar sevgi ve emeklerini esirgemeyen, desteklerini her zaman hissettiğim aileme sonsuz teşekkürlerimi bir borç bilirim.

Nisan – 2014

İlyas DİRİN

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜRLER	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ	vi
SİMGELEr VE KISALTMALAR DİZİNİ	vii
ÖZET.....	ix
ABSTRACT.....	x
GİRİŞ	1
1. GENEL BİLGİLER.....	4
1.1. Robot Kavramı.....	4
1.2. Robotta Eklem ve Manipülatör Kavramı.....	5
1.3. Robot Tipleri.....	5
1.3.1. Serbestlik derecelerine göre robotlar	6
1.3.1.1. Kartezyen robotlar	6
1.3.1.2. Silindirik robotlar.....	6
1.3.1.3. Küresel robotlar	6
1.3.1.4. SCARA (RRP) Robotlar	7
1.3.1.5. Dönel robotlar	7
1.3.2. Mekanik tasarımına göre robot tipleri.....	8
1.3.2.1. Seri robotlar	8
1.3.2.2. Paralel Robotlar	9
1.3.2.3. Paralel robotların tarihi gelişimi	10
2. PARALEL ROBOT KİNEMATİĞİ	14
2.1. Kinematiksel Kavramlar	14
2.1.1. Konum Tanımı	14
2.1.2. Yönelim Tanımı	15
2.1.3. Öteleme İşlemi	16
2.2. İleri Kinematik.....	17
2.3. Ters Kinematik.....	18
3. YÖRÜNGE PLANLAMASI.....	22
4. ROBOTUN BİLGİSAYAR ORTAMINDA MODELLENMESİ	27
4.1. Robotun Elektronik Birimleri	27
4.1.1. Çevresel birimler.....	28
4.1.1.1. 3 Axis joystick	29
4.1.1.2. ADXL345 ivme sensörü	29
4.1.1.3. HMC5883L dijital pusula	30
4.1.1.4. T6963C grafik LCD mikro işlemcisi	32
4.1.1.5. L298 motor sürücü entegre	33
4.1.2. Kontrol birimi	34
4.2. I ² C Haberleşme Protokolü	37
4.3. CCS C Programlama Dili.....	38
4.4. Robotun Proteus Ortamında Tasarımı ve Simülasyonu.....	39
4.4.1. Robotun elektronik komponentlerinin simülasyonu	39

4.4.2. Robotun PCB tasarımı	40
5. ROBOTUN FİZİKSEL ORTAMDA UYGULAMASI	44
5.1. Robotun Mekanik Tasarımı	44
5.2. 3-UPS, 1-RU Robotun Çalışma Modları	54
6. PARALEL ROBOTLARDA YENİ KULLANIM ALANI ÖNERMELERİ.....	60
6.1. Paralel Robotların Anten Tilt Aparatı Olarak Kullanılması	60
6.2. Paralel Robotların Hareketli Baz İstasyonlarında Kullanılması	65
7. SONUÇLAR ve ÖNERİLER.....	71
KAYNAKLAR	73
EKLER.....	76
ÖZGEÇMİŞ	110

ŞEKİLLER DİZİNİ

Şekil 1.1.	LEGO Mindstorms NXT 2.0 ve ASIMO robotu	5
Şekil 1.2.	Kartezyen robot.....	6
Şekil 1.3.	Silindirik robot.....	6
Şekil 1.4.	Küresel robot.....	7
Şekil 1.5.	SCARA	7
Şekil 1.6.	Dönel.....	7
Şekil 1.7.	Seri robot kolu ve Lego Technic seri robot kolu	8
Şekil 1.8.	ABB Delta paralel robotu ve Lego Technic Delta robotu	9
Şekil 1.9.	İlk patentli paralel mekanizma.....	10
Şekil 1.10.	Paralel boya püskürtme mekanizması.....	11
Şekil 1.11.	a) İlk 6 serbestlik dereceli paralel mekanizma b)Gough platformunun 2000 yılında kullanılan hali.....	12
Şekil 1.12.	Stewart uçuş	12
Şekil 1.13.	Akıllı robotik sandalye.....	13
Şekil 1.14.	3-PRS paralel medikal robot.....	13
Şekil 2.1.	Eş merkezli iki koordinat	16
Şekil 2.2.	Yönelimleri ve merkezleri farklı koordinat	16
Şekil 2.3.	Üç eklemlili seri robot.....	17
Şekil 2.4.	3 DOF RRR Robot.....	18
Şekil 2.5.	Roll-Pitch-Yaw	19
Şekil 2.6.	3-UPS robot	19
Şekil 3.1.	Kübik yörünge planlaması a) Konum, b) Hız, c) ivme grafiği	26
Şekil 4.1.	Robotun elektronik bilgi akışı blok diyagramı	27
Şekil 4.2.	Robotun arabirimleri	28
Şekil 4.3.	3 axis joystick	29
Şekil 4.4.	ADXL345 bağlantı şeması.....	30
Şekil 4.5.	HMC5883L entegresinin	31
Şekil 4.6.	T6963C mikro işlemcisi ve LCD blok diyagramı.....	32
Şekil 4.7.	L298 H-Bridge devresi	33
Şekil 4.8.	PIC18F4685 mikro denetleyicisinin pinleri.....	35
Şekil 4.9.	dsPIC33FJ32GP202 mikro denetleyicisinin pin yapısı	36
Şekil 4.10.	I ² C mikrodenetleyici ve elemanların bağlantı şeması.....	37
Şekil 4.11.	I ² C iletişimde veri gönderme işlemi	38
Şekil 4.12.	Proteus ISIS ortamında elektronik komponentlerin analizi	40
Şekil 4.13.	PCB ARES çizimi.....	41
Şekil 4.14.	PCB ARES çıktısı	42
Şekil 4.15.	ARES 3 boyutlu komponent tasarımı	43
Şekil 4.16.	Kontrol kartı.....	43
Şekil 5.1.	3-UPS paralel mekanizma	45
Şekil 5.2.	3-UPS, 1RU paralel mekanizma	46
Şekil 5.3.	3-UPS, 1-RU robotun ölçüleri	47
Şekil 5.4.	X eksenindeki çalışma uzayı.....	48
Şekil 5.5.	Y eksenindeki çalışma uzayı.....	48

Şekil 5.6.	Z eksenindeki çalışma uzayı	49
Şekil 5.7.	X ekseninden toplam çalışma uzayı.....	49
Şekil 5.8.	Y ekseninden toplam çalışma uzayı.....	49
Şekil 5.9.	Z ekseninden toplam çalışma uzayı	50
Şekil 5.10.	Universal ve kübik eklem	50
Şekil 5.11.	Firgelli lineer servo motor	51
Şekil 5.12.	MG 995 servo motor	52
Şekil 5.13.	Servo motor çalışması için PWM sinyali	52
Şekil 5.14.	3-UPS,1-RU robotun	53
Şekil 5.15.	3-UPS,1-RU robotun elektronik komponentleri ile	53
Şekil 5.16.	3-UPS,1-RU robotun tüm donanımları ile birlikte çalıştırılması.....	54
Şekil 5.17.	3-UPS, 1-RU robotun joystick vasıtasıyla Y eksenindeki hareketi	56
Şekil 5.18.	3-UPS, 1-RU robotun joystick vasıtasıyla Z eksenindeki hareketi	56
Şekil 5.19.	3-UPS,1-RU robotun sensörler vasıtasıyla X ve Y eksenindeki hareketi	57
Şekil 5.20.	3-UPS,1-RU robotun sabit platformunun X ekseninde hareket ettirilmesi.....	58
Şekil 5.21.	3-UPS,1-RU robotun sabit platformunun Y ekseninde hareket ettirilmesi.....	59
Şekil 6.1.	Kablosuz haberleşme gelişimi	60
Şekil 6.2.	Baz istasyonu kapsamı	61
Şekil 6.3.	Mekanik tilt.....	61
Şekil 6.4.	Elektrik tilt	62
Şekil 6.5.	Paralel mekanizmalı anten tilt.....	63
Şekil 6.6.	3-UPS,1RU robotun anten yön	64
Şekil 6.7.	3-UPS,1RU robotun anten yön	64
Şekil 6.8.	3-UPS,1RU robotun anten yön değişimi	65
Şekil 6.9.	Sudaki botun savrulma eksenleri	66
Şekil 6.10.	Su ve kara arası radyolink haberleşmesi	67
Şekil 6.11.	Paralel mekanizmaların radyo.....	68
Şekil 6.12.	Radyo link sistemlerinin paralel	69
Şekil 6.13.	3-UPS, 1-RU robotun hareketli ortamda sabit platform olarak	70

TABLolar DİZİNİ

Tablo 4.1. HMC5883L Register listesi	31
Tablo 4.2. Kontrol birimi mikro denetleyici özellikleri.....	34

SİMGELER VE KISALTMALAR DİZİNİ

ACK	: ACKnowledgement (Alındı bildirimi)
ADC	: Analog Digital Converter (Analog Dijital Çevirici)
CAD	: Computer Aided Design (Bilgisayar Destekli Dizayn)
CAM	: Computer Aided Manufacturing (Bilgisayar Destekli Üretim)
CPU	: Central Processor Unit (Merkezi İşlem Birimi)
D-H	: Denavit-Hartenberg
DoF	: Degree of Freedom (Serbestlik Derecesi)
EDGE	: Enhanced Data rated for GSM Evoluation (Gelişmiş veri oranlı GSM Evrimi)
GLCD	: Graphical Liquid Crystal Display (Grafik Sıvı Kristal Ekran)
GPRS	: General Packet Radio Service (Genel Paket Radyo Servisi)
GSM	: Global System for Mobile Communications (Mobil İletişim için Küresel Sistem)
HDSL	: High bit rate Digital Subscriber Line (Yüksek bit oranlı Dijital Abone Hattı)
HEX	: Hexadesimal (Onaltılı)
HSPA	: High Speed Packet Access (Yüksek Hızlı Paket Erişimi)
I/O	: Input Output (Giriş Çıkış)
I2C	: Inter Integrated Circuit (Arası Entegre Devre)
LCD	: Liquid Crystal Display (Sıvı Kristal Ekran)
LOS	: Line Of Sight (Doğrudan Görüş)
LTE	: Long Term Evoluation (Uzun Vadeli haberleşme Evrimi)
MIPS	: Mega Instruction Per Second (Saniyede Milyon Talimat)
ms	: milisecond (milisaniye)
MSSP	: the Master Synchronous Serial Port (Master Senkron Seri Port)
NACK	: Not ACKnowledgement (Alınmadı Bildirimi)
PIC	: Programable Interface Controller (Programlanabilir Arabirim Kontrolörü)
PKM	: Paralel Kinematik Mekanizmalar
PLL	: Phase Lock Loop (Faz Kilitlemeli Döngü)
PTP	: Point To Point (Noktadan Noktaya)
PWM	: Pulse Width Modulation (Darbe Genişlik Modülasyonu)
RAM	: Random Access Memory (Rastgele Erişim Belleği)
ROM	: Read Only Memory (Sadece Okunabilir Bellek)
RPS	: Revolute Prismatic Spherical joint (Dönel Prizmatik Küresel eklem)
RRR	: Revolute Revolute Revolute joint (Dönel Dönel Dönel eklem)
RU	: Revolute Universal joint (Dönel Universal eklem)
SCL	: Serial Clock Line (Seri Senkronizasyon Hattı)
SDA	: Serial DATA Line (Deri Veri Hattı)
SPI	: Serial Peripheral Interface (Seri Çevresel Arabirim)
TTL	: Transistor Transistor Logic (Transistör Transistör Mantığı)
UMTS	: Universal Mobile Telecommunication System (Uluslararası Mobil Telekomünikasyon Sistemi)

UPS : Universal Prismatic Spherical joint (Universal Prizmatik Kresel Eklem)
VRAM : Video Random Access Memory (Video Rastgele Eriřim Belleęi)
ZIF : Zero Insertion Force (Sıfır Takma Gc)

PARALEL ROBOT KİNEMATİĞİ, YÖRÜNGE PLANLAMASI VE 3-UPS, 1-RU PARALEL ROBOT UYGULAMASI

ÖZET

Son yıllarda paralel mekanizmalar üzerine yapılan çalışmalar ve endüstride kullanım alanlarının artması ile birlikte, bu mekanizmalar önem kazanmıştır. Genel olarak altta sabit bir platform üzerinde paralel eklemlerle bağlanmış hareketli platformlardan oluşurlar. Birden fazla seri mekanizmanın birleştirilerek kapalı çevrim bir yapı oluşturması ile elde edilirler. Özellikle yüksek kinematik hassasiyeti, kararlılıkları, yüksek yük kapasitesi, hassas pozisyon kontrolü gibi avantajları, paralel mekanizmaların tercih sebeplerindedir. Bunun yanında seri mekanizmalara göre kinematik denklemlerinin zorluğu ve çalışma uzayının sınırlı olması hala paralel mekanizmaların üzerinde çalışılan dezavantajlarıdır.

Bu çalışmada 3-UPS, 1-RU paralel bir robot uygulaması üzerine yoğunlaşmıştır. Robotun mekanik yapısı gereği oluşan kinematik denklemleri incelenmiştir. Kinematik denklemleri ile elde edilen varılacak konuma yörünge planlaması yapılmıştır. Beji ve Pascal modeline eklemeler yapılarak yeni bir model önerilmiştir. Robotun teorik kavramlarda belirtildiği şekilde mekanik uygulaması yapılmış, kinematik ve yörünge planlaması denklemlerine göre kontrolü gerçekleştirilmiştir. Program kodu kaynaklarda verilmiştir.

Paralel robotların mevcutta olmayan telekomünikasyon alanında yeni kullanım alanları önerilmiştir. Paralel robotlar, üzerinde çalışmaların devam ettiği anten tilt ayarları ve hareketli radyo link problemlerinin çözümü olarak önerilmiştir.

Anahtar Kelimeler: 3 Serbestlik Dereceli, Kinematik, Paralel Robot, Stewart Platformu, Yörünge Planlaması

KINEMATICS OF PARALLEL ROBOTS, TRAJECTORY PLANNING AND 3-UPS, 1-RU PARALLEL ROBOT APPLICATION

ABSTRACT

Over the past decades, along with the increases of studies and usage area of the parallel mechanisms, the importance of these mechanisms gets more significant. Generally, parallel mechanisms consist of a fixed base platform which is connected to upper moving platform with parallel set of joints. Several serial joints are combined in parallel to form a closed loop construction. Especially, advantages such as higher kinematic sensibility, higher precision, greater load capacity, accurate position control are the preference reasons of parallel mechanisms. Besides these advantages, the complex kinematic calculations and limited workspaces according to serial mechanism are the disadvantages which are still in common study areas of parallel mechanisms.

In this study, 3-UPS, 1-RU parallel robot application is focused on. The kinematic problems which arise from the robot's natural mechanical structure, is examined. After the kinematic solution achieved, the orbit planning equations are studied in order to get the robot to its destination point. A new parallel robot model is offered over Beji and Pascal model by adding a new property to existing model. The mechanical structure is then applied as it was indicated in its theoretical concept and its control is achieved properly to its kinematic and trajectory planning equations. The source code are given in the addition section.

New usage fields are suggested in telecommunication sector, which are not in use yet. A new solution is discussed over the problem of adjusting antenna tilt and moving radio links.

Keywords: 3 Degree of Freedom, Kinematic, Parallel Robot, Stewart Platform, Trajectory Planning

GİRİŞ

Robotik bilimi insanoğlunun ihtiyaçlarını kendisi yerine farklı nesnelere yaptırma eğiliminden doğmaktadır. Bu eğilimden dolayı insanoğlu tarihinin başlangıcından beri yaşamını idame ettirebilmek ve kolaylaştırabilmek için icatlar geliştirmiştir.

Robot kelimesi ilk olarak Çek oyun yazarı Karel Kapek tarafından yazılan ‘Rossum’un Akıllı Robotları’ adlı tiyatro oyununda Çek dilinde hizmet eden kelimesi anlamına gelen “robota” kelimesi olarak karşımıza çıkar. 1920 yılı başlarında yazılan bu oyunda robotlar hisleri olmayan mekanizmalar olarak tanımlanmış ve sonrasında birçok bilim kurgu eserinde bahsi geçmiştir.

Antik çağlarda Yunan filozofu Aristo “Eğer tüm araçlar emredildiğinde veya kendi kendilerine çalışabilselerdi işçilere ve kölelere ihtiyaç kalmazdı” sözleri ile robotlar hakkında ilk kez bahsetmiştir. Fakat sanayi devrimine kadar bazı basit makineler dışında robotik biliminde elle tutulur bir gelişme kaydedilememiştir. Genel olarak yapılan çalışmalar teori, fikir ve hayali tasarımdan ibaret kalmıştır. Sanayi devrimi ile beraber teknolojiadaki gelişmelerin hızlanması robotik sistemlerin endüstriyel alana girmesini sağlamıştır. 1941 yılında Isaac Asimov ‘Robot’ kelimesinden ‘Robotik’ kelimesini türeterek ilk kez kullanmıştır. Isaac Asimov 1942’de ‘Runaround’ adlı hikayesinde robotların amaçlarının sınırlarını tutarlı bir şekilde çizdiği robot yasasını belirtmiştir;

1. Bir robot, bir insana zarar veremez veya kayıtsız kalmak suretiyle zarar görmesine izin veremez.
2. Birinci yasayla çatışmamak koşulu ile bir robot insanlar tarafından verilen emirlere uymak zorundadır.
3. Birinci ve ikinci yasayla çatışmamak koşulu ile bir robot kendi varlığını korumalıdır.

İlerleyen yıllarda robotlar teknolojinin gelişmesi ile birlikte tekrar programlanabilir, birden fazla işi yerine getirebilir mekanizmalar tasarlanarak üretimde verimliliğin ve hızın vazgeçilmezi haline gelmiştir. Bilgisayar teknolojisi ile birlikte mikroişlemci kontrollü defalarca programlanabilen robotlar gelişen yarıiletken ve sensör teknolojilerini kullanarak saniyede milyonlarca hatasız işlem yapabilecek kapasitelere gelmiştir [1].

Günümüzde mekatronik sistemler olarak geliştirilen robotlar tıptan, uzay araştırmalarına kadar hemen hemen her alanda faaliyet göstermektedir.

Bu tez çalışmasında paralel robotların genel yapısı incelenmiş, tez uygulamasında konu olan 3 serbestlik dereceli paralel robotun ters kinematik denklemleri ve yörünge planlaması denklemleri çözülmüştür. Elde edilen teorik bilginin uygulaması yapılmış ve mikro denetleyicili bir sistem ile elde edilen denklemlerin tamamı robotik sistemde uygulanmıştır. Ayrıca tezde önerilen yapı ve paralel robotlar için telekomünikasyonda hareketli anten sistemlerinin oluşturduğu problem ve anten tilt problemi üzerine çözüm önerilmiştir. Tez kapsamında yapılan çalışmalar genel olarak 7 bölümde incelenmiştir.

Bölüm 1’de robotik kavramından ve tarihçesinden bahsedilmiştir. Robotların tarihsel gelişmesi incelenmiş ve temel robot kavramlarından bahsedilmiştir.

Bölüm 2’de paralel robotların kinematiği üzerine yoğunlaşmıştır. Kinematik hesabının yapılabilmesi için öncelikle yönelim, öteleme, konum kavramlarından bahsedilmiş, paralel robotların ileri ve ters kinematiğine değinilmiştir. Bu tez uygulamasında ters kinematik çözümleri kullanıldığı için ters kinematik derinlemesine irdelenmiştir.

Bölüm 3’de robot yörünge planlamasına değinilmiştir. Yörünge planlamasının tanımı, gerekliliğinden bahsedilerek ardından üçüncü dereceden polinom ile yörünge planlaması yapılmıştır.

Bölüm 4’de tez çalışmasında yapılacak olan uygulamanın bilgisayar ortamında elektronik komponentlerinin simülasyonu yapılmış ve kullanılan sensör, mikro denetleyiciler anlatılmıştır. Mikro denetleyicilerin programlanması, sensörler ve kendi aralarında haberleşmesi gibi elektronik konular irdelenmiştir. Oluşturulan devre bilgisayar ortamında çizilmiş, 3 boyutlu olarak modellenmiştir. İlerleyen aşamalarda bilgisayar ortamında tasarımı yapılan elektronik kartın PCB’si yapılmıştır.

Bölüm 5’de 3-UPS,1-RU robot Lego Technic parçaları ile tasarlanmıştır. Bölüm 4’te anlatılan elektronik komponentler fiziksel olarak board ortamında kullanılmış, mikrodenetleyiciler ile kontrol ünitesi oluşturulmuştur. Oluşturulan kontrol ünitesi ile mekanik sistem ve motorlar sürülmüş, Bölüm 2 ve 3’te anlatılan paralel robot kinematiği ve yörünge planlamasının uygulaması sağlanmıştır.

Bölüm 6’da paralel robotların mevcut kullanım alanları dışında telekomünikasyon alanında yeni kullanım alanı önerilmiş ve tartışılmıştır. Bunlar sıkça kullanılan anten tilt ve hareketli radyo link problemidir. Bu problemlere paralel robotlar vasıtasıyla çözüm önerisinde bulunulmuştur.

Bölüm 7’de yapılan çalışmanın özeti anlatılmış ve yeni önerilen kullanım alanlarına uygunluğundan kısaca bahsedilmiştir.

1. GENEL BİLGİLER

1.1. Robot Kavramı

Robot kelimesi hizmet eden anlamına gelmektedir. Robot, otonom veya bir amaca özel programlanmış, görevleri yerine getiren mekatronik sistemlerdir. Robotik sistemler, teknolojiye gelişimlerle birlikte özellikle bilgisayar destekli tasarım (Computer Aided Design – CAD) ve bilgisayar destekli üretim (Computer Aided Manufacturing – CAM) gibi tekniklerin geliştirilmesi ile endüstriyel alanda kısa zamanda yaygınlaşmıştır. Robotların günümüzde otomotivden tıpa, uzay arařtırmalarından askeri alanlar gibi hızlı, hatasız üretim, hassasiyet gerektiren birçok alanda uygulamaları mevcuttur.

Robotlar, kendisini oluřturan mekanik sistemlerin elektronik devreler ve bilgisayar yazılımlarıyla kontrol edilmesi ile oluřur. Robotun mekanięi eklemlerle birbirine baęlanmış sıralı katı cisimlerden oluřur. Bu sıralı cisimlerin hareketi eklemlerde bulunan eyleyiciler ile saęlanır. Robotik sistemin eyleyicileri hareketi kontrol eden ve yöneten bir kontrol sistemi ile saęlanır. Genel olarak mikro denetleyiciler ile robotların kontrol sistemleri oluřturulur. Bunun yanında bir robot çevresinden topladıęı bilgileri yorumlayarak bir sonraki hareketine karar verir.

Robotik sistemler birçok bilim kurgu eserinde insana benzer tanıtılmasına raęmen robotların büyük bir kısmı insana benzemeyen tipte sistemlerdir. Yine de her sistemin tasarımında insandan ve doęadan faydalanılmıřtır. İnsanoęlunun iskelet sistemi mekatronik bir sistemin mekanik aksamı, insanoęlunun duyu organları robotik sistemin geri beslemeli sensörleri, insanoęlunun beyin aktiviteleri de robotun mikro denetleyicisi ile gerçekleřtirdięi işlemlere benzetilebilir. Robotların deęişmez yasası ise duygularının olmaması ve kendi başlarına karar alamamalarıdır. Robotlar sadece önceden programlandıkları görevleri yerine getirirler. Şekil 1.1’de Lego technic ile oluřturulmuş Mindstorm robotları ve Asimo robotu gösterilmiřtir.



Şekil 1.1. LEGO Mindstorms NXT 2.0 ve ASIMO robotu [2 - 4]

1.2. Robotta Eklem ve Manipülâtör Kavramı

Genel olarak robotik bir sistem aşağıdaki bölümlerden oluşmaktadır;

- Eklemlerle birbirine bağlanmış sıralı katı cisimlerden (uzuvlardan) oluşan mekanik yapı, serbestliği sağlayan koldan (Arm), el becerisi sağlayan bir bilekten (Wrist) ve robotun yapması gereken görevi tamamlayan sonlandırıcıdan (End Effector)
- Eklemlerin hareketlenmesiyle manipülâtörün hareketini sağlayan hareketlendiriciler pnömatik, hidrolik veya motorlu sistemler (Actuators)
- Robotun çevre ile etkileşimini sağlayacak geri besleme sistemleri olarak sensörler (Sensors)
- Robotun iç ve dış birimlerinin uyum içerisinde çalışmasını sağlayacak merkezi bir kontrol sistemi

Bir robotun baştan uca kontrol edilmesinde mekanik, elektronik ve kontrol yazılım ve iyi bir matematik yetkinliği gerekmektedir. Birden fazla bilim dalını içeren robotik bilimi aynı zamanda robotun tasarımına göre zaman zaman birçok kompleks denklem anlamına da gelmektedir.

1.3. Robot Tipleri

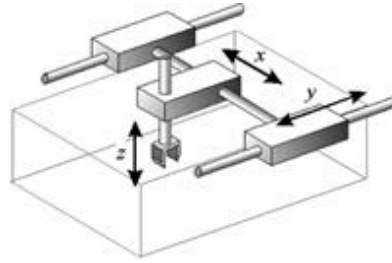
Robotlar serbestlik derecelerine, kontrol yöntemlerine, kullandıkları güç kaynaklarına, çözünürlüklerine ve mekaniksel tasarımlarına göre farklı şekillerde sınıflandırılabilirler.

1.3.1. Serbestlik derecelerine göre robotlar

Robotlar serbestlik derecelerine göre sınıflandırılırken genel olarak ilk üç ekleminin türüne göre sınıflandırılırlar.

1.3.1.1. Kartezyen robotlar

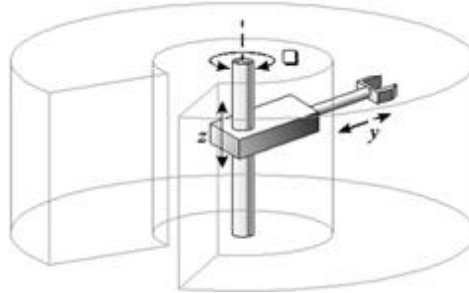
İlk üç eklemi prizmatik (PPP) olarak tasarlanan robotlardır. Kinematik çözümleri en kolay olan robotlardır. Bu robotlar genel olarak taşıma gibi ağır işlerde kullanılırlar. Şekil 1.2’de tipik bir Kartezyen robot örneği görünmektedir.



Şekil 1.2. Kartezyen robot [5]

1.3.1.2. Silindirik robotlar

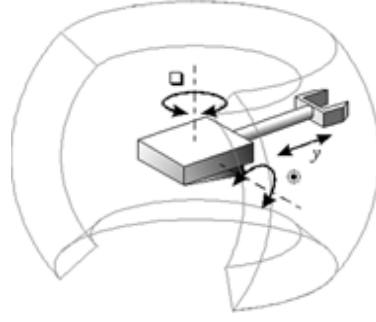
İlk eklemi dönel, ikinci ve üçüncü eklemi prizmatik olan (RPP) robotlardır. Şekil 1.3’te genel silindirik robot tasarımı görünmektedir.



Şekil 1.3. Silindirik robot [5]

1.3.1.3. Küresel robotlar

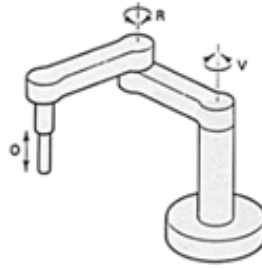
Küresel robotlar ilk eklemi dönel, üçüncü eklemi ise prizmatik olan mekanizmalardır. Kutupsal koordinat sistemine sahip robotlardır. Şekil 1.4’te örnek bir küresel robot gösterilmiştir.



Şekil 1.4. Küresel robot [5]

1.3.1.4. SCARA (RRP) Robotlar

SCARA tip robotların ilk iki eklemi dönel üçüncü eklemi ise prizmatik şekildedir. Endüstri alanında bir çok al ve yerleştir uygulamalarında kullanılan SCARA tip robot Şekil 1.5’te gösterilmiştir.



Şekil 1.5. SCARA robot [5]

1.3.1.5. Dönel robotlar

Dönel robotlar en az üç adet dönel eklemden (RRR) oluşan robotlardır. Endüstriyel birçok uygulaması bulunan dönel robotun yapısı Şekil 1.6’da gösterilmiştir.



Şekil 1.6. Dönel robot [5]

1.3.2. Mekanik tasarımına göre robot tipleri

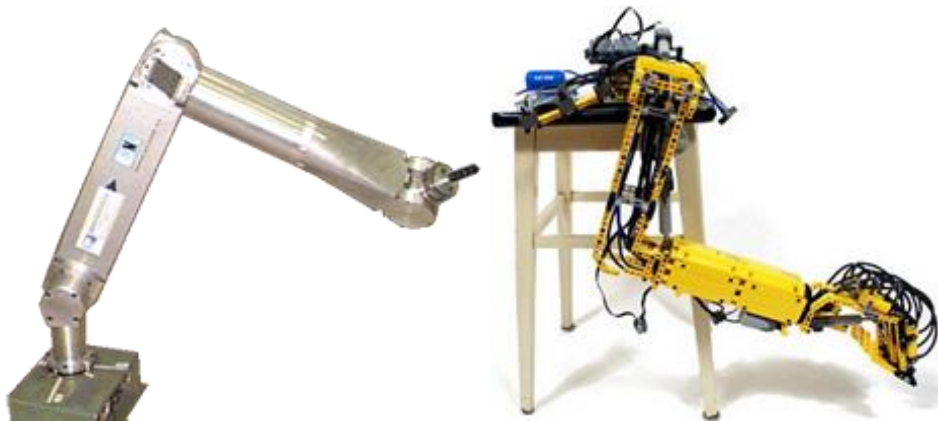
Mekanik tasarımına göre robot tipleri, robot eklemlerinin seri veya paralel bağlanmasına göre sınıflandırılırlar.

1.3.2.1. Seri robotlar

Bu tez çalışmasının odağı paralel robotlar olmasına rağmen paralel robotların anlaşılabilmesi için öncelikle seri robot kavramının anlaşılması gerekmektedir. Seri robotlar seri bir hat üzerinde bir dizi katı uzuvlardan ve bu uzuvları birbirine bağlayan eklemlerden oluşur.

Seri robotlar endüstriyel alanda otomotivde, kaynak makinelerinde, seçme, yerleştirme, paketlenme, delme, çizim gibi alanlarda uygulamaları mevcuttur.

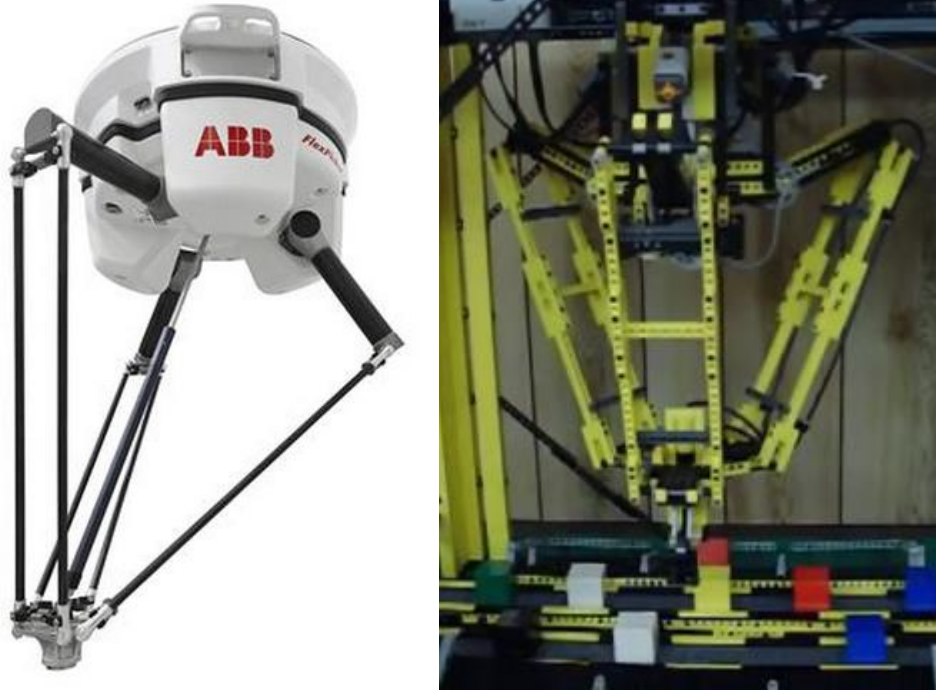
Seri robotlar geniş çalışma uzayına sahiptirler. Paralel robotlara göre az sayıda mekanik parçaya sahiptirler. Seri robotların kinematik denklemleri ve yörünge planlama denklemleri paralel robotlara göre daha basittir. Buna karşılık seri robotların yüklenme oranları paralel robotlara oranla düşüktür. Seri robotların mekanik aksamaları seri bir hat şeklinde olduğundan hata oranları bu seri hat üzerinden toplanarak gider. Bu yüzden paralel robotlara oranla hassasiyetleri ve kararlılıkları daha düşüktür. Şekil 1.7’de seri robot örnekleri gösterilmiştir.



Şekil 1.7. Seri robot kolu ve Lego Technic seri robot kolu [6, 7]

1.3.2.2. Paralel Robotlar

Zaman zaman hexapod olarak da isimlendirilen paralel robotlar veya Paralel Kinematik Mekanizmaları (PKM) yüksek katılık ve hassasiyet, yüksek hız ve yüksek yük kapasitesi olan mekanizmalardır. Astronomiden uçuş simülatörlerine, tıptan askeri endüstriye kadar uzanan geniş kullanımı ile paralel robotlar üzerinde yapılan çalışmalar ve uygulamalar günden güne artmaktadır [8].

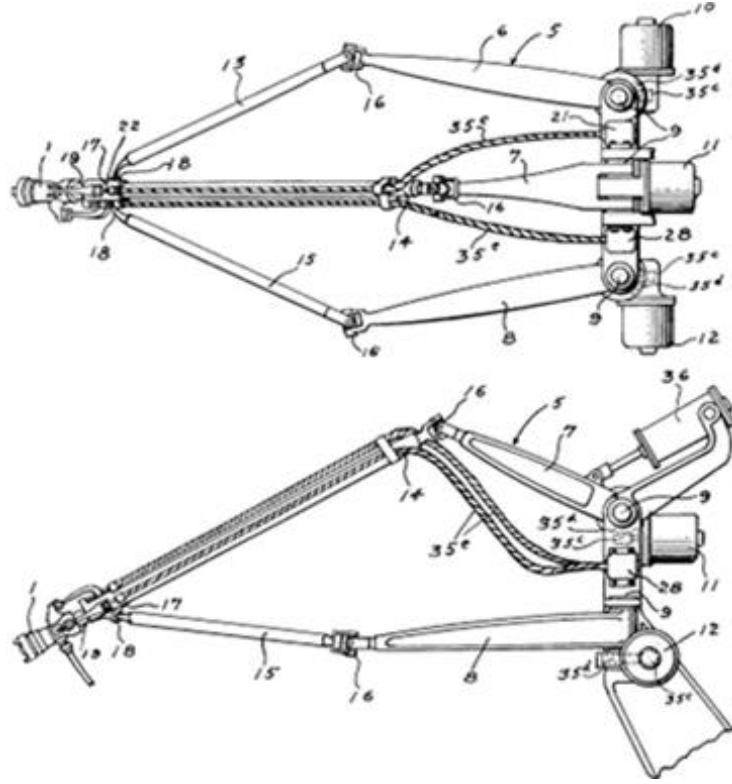


Şekil 1.8. ABB Delta paralel robotu ve Lego Technic Delta robotu [9, 10]

Paralel mekanizmalar genellikle sabit bir baz platforma birden fazla seri mekanizmanın bağlanması ile oluşturulan kapalı çevrim mekanizmalardır. Seri mekanizmalar ile belirli eklemler ile birbirlerine bağlanırlar ve robotun ucunda bileşke hareketin uygulandığı manipülatör bulunur. Paralel bağlı bulunan uzuvların boyları veya açıları değiştirilerek uç işlevci istenilen pozisyona getirilir. Şekil 1.8’de ABB firmasının ürettiği Delta robotu ve aynı Delta robotun Lego technic versiyonu gösterilmiştir.

Paralel robotların seri robotlara göre üstünlüğü aynı uç işlevciye birden fazla kol bağlandığı için seri robotlara oranla yük kapasitesi daha fazladır. Hata oranı düşük, hassasiyet ve kararlılığı yüksek robotlardır. Fakat bunun yanında seri robotlara göre

Yaklaşık on yıl sonra Willard L.V. Pollard tarafından ilk otomatik sprej boyama mekanizması geliştirilmiştir. Şekil 1.10'da gösterilen 5 serbestlik dereceli olarak gerçekleştirilen bu robot ilk endüstriyel paralel robot olarak bilinmektedir.



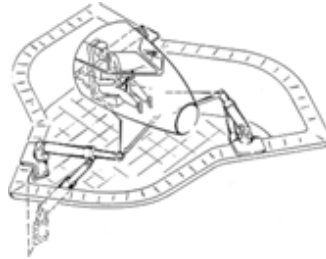
Şekil 1.10. Paralel boya püskürtme mekanizması [13]

1947 yılında Dr. Eric Gough, endüstriye yön verecek olan octahedral hexapod 6 serbestlik dereceli paralel mekanizmayı geliştirdi. Bu mekanizma otomotiv sanayisinde lastiklerin dinamik bileşke kuvvetlere olan dayanımını test amaçlı olarak geliştirildi. Benzer mekanik yapılar elektronik sistemlerle donatılmış mikroişlemci kontrollü şekilde günümüzde de kullanılmaktadır. Dr. Eric Gough'un geliştirdiği paralel model ve günümüzde kullanılan şekli Şekil 1.11'de gösterilmiştir.



Şekil 1.11. a) İlk 6 serbestlik dereceli paralel mekanizma b) Gough platformunun 2000 yılında kullanılan hali [12]

1965 yılında D. Stewart 6 serbestlik dereceli Şekil 1.12’de gösterilen uçuş simülâtöründen bahsettiği makalesini yayınladı. Bu makale içerisinde Dr. Gough’unda bulunduğu bilimadamları arasında büyük bir tartışma ortamı yarattı. Dr. Gough kendisinin geliştirdiği 6 serbestlik mekanizmanın var olduğunu belirtmiştir [12].



Şekil 1.12. Stewart uçuş simülâtörü

2011 yılında geliştirilen diğer güncel bir uygulama ise Şekil 1.13’de gösterilen hasta bakımı için geliştirilmiş çok fonksiyonel bir sandalyedir. Stewart Platform’undan türetilen bu sandalye bir hastanın ergonomisi için tasarlanmış ve 6 serbestlik dereceli bir Stewart Platform’unun 4 serbestlik derecesine indirgenmesi ile oluşturulmuştur. 4 serbestlik derecesi ile hastanın sandalyeye istenilen eğimde istenile konumu sağlaması imkanı sunulmuştur. Bunun yanında sandalye ilgili merkeze hasta ile ilgili bir takım bilgileri yollamakta ve hastanın durumu ile ilgili geribildirim vermektedir [14].



Şekil 1.13. Akıllı robotik sandalye [14]

Diğer bir güncel çalışma ise delta tip paralel robotun hassas suni teneffüs cihazı olarak kurgulanmasıdır. Bu çalışmada hareketli platform insandan çok daha hassas periyodik bir şekilde kalp masajı yapabilmektedir [15]. 3-RPS paralel medikal robot Şekil 1.14’te gösterilmiştir.



Şekil 1.14. 3-PRS paralel medikal robot [15]

Gelişen teknoloji ile birlikte Stewart platformları radar, petrol platformları, eğlence parkları, kimyasal taşımacılık, mikro tıbbi operasyonlarda, uçak, tank, deprem simülasyonlarında olmak üzere birçok alanda kullanılmaktadır.

2. PARALEL ROBOT KİNEMATİĞİ

2.1. Kinematiksel Kavramlar

Mekatronik bir sistemin yapısını ve yapısından kaynaklanan hareketlerini inceleyen bilim dalına kinematik denir. Robotların kinematik hesaplamalarının yapılabilmesi için kendisinin ve çevresindeki nesnelerin içerisinde bulunduğu üç boyutlu uzayda, konum ve yönelim tanımlamalarının yapılması gerekmektedir. Bu tanımlamaların yapılabilmesi için robotun kendisi ile birlikte bütün nesnelerin birer koordinat sistemine ihtiyacı vardır. Bu koordinat sistemleri evrensel bir koordinat sistemi ile ilişkilendirilir. Tanımlanan tüm konum ve yönelimler hem ilgili nesnenin koordinat sistemine göre hem de tanımlanan bu evrensel koordinat sistemine göre hesaplamalara dahil olur.

Kinematik hesaplamalarında Homojen Dönüşüm Yöntemi, Üssel Yöntem, Sıfır Referans Konum Yöntemi, Pieper-Roth Yöntemi, Tam ve Parametrik Sürekli Yöntem gibi birçok yöntem bulunmasına rağmen ileri yön kinematiğini doğrudan çıkaran ve daha basite indirgeyen yöntem olarak kabul edilen Homojen Dönüşüm Yöntemi kullanılmaktadır. D-H yöntemi olarak da bilinen bu yöntem Denavit-Hartenberg gösterimini kullanan yöntemdir. D-H değişkenleri; iki eklem arasındaki bağ uzunluğu, bağ açısı, üst üste çakışan bağlar arasındaki bağ kayması, iki bağ arasında oluşan eklem açısıdır [16].

2.1.1. Konum Tanımı

Üç boyutlu uzayda herhangi bir nesne, koordinat sistemi tanımlanmak suretiyle 3×1 boyutlu bir konum vektörü ile gösterilebilir. Temel koordinat sistemi içerisinde birden çok koordinat sistemi belirtilebilir. Bu yüzden konum vektörü tanımlı olduğu koordinat sistemi ile ilişkilendirilerek isimlendirilir. Örneğin A koordinat sisteminde tanımlı bir P noktası ${}^A P$ aşağıdaki gibi gösterilir.

$${}^A P = \begin{bmatrix} {}^A P_x \\ {}^A P_y \\ {}^A P_z \end{bmatrix} \quad (2.1)$$

2.1.2. Yönelim Tanımı

Üç boyutlu uzay içerisinde birden fazla koordinat sistemi tanımlanabilir. Bu koordinat sistemlerinin birbirlerine olan açı farkları yönelimleri olarak tanımlanır. Yönelim 3x3 boyutlu bir matris ile ifade edilir. Bir nesnenin başka bir koordinat sistemli nesneye göre yönelimini tanımlayabilmek için koordinat sistemi tanımlanır. Örneğin A referans koordinatının yanında B koordinat sistemi tanımlansın. B koordinat sisteminin birim vektörü aşağıdaki gibi gösterilir.

$$\{B\} = \hat{X}_B, \hat{Y}_B, \hat{Z}_B \quad (2.2)$$

B koordinat sisteminin vektörünün A koordinat sistemine göre gösterilmesi ise aşağıdaki gibidir.

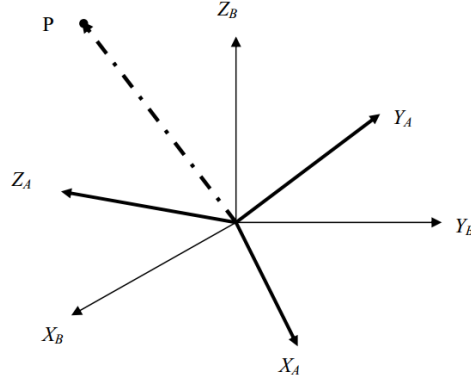
$$\{A\} = {}^A \hat{X}_B, {}^A \hat{Y}_B, {}^A \hat{Z}_B \quad (2.3)$$

Yönelim kavramının 3x3 boyutlu bir matris ile gösterildiğinden bahsedilmişti. Bu matrise koordinatlar arasındaki dönme matrisi de denir. Dönme matrisi koordinat sistemleri arasındaki her eksenin birim vektörlerinin arasındaki açının kosinüs ifadesi olarak belirlenir. {B} koordinat sisteminin yönelimini {A} koordinat sistemine göre veren dönme matrisi aşağıdaki gibi ifade edilir.

$${}^A R = \begin{bmatrix} |X_B||X_A|\cos\gamma_{AB} & |Y_B||X_A|\cos\gamma_{AB} & |Z_B||X_A|\cos\gamma_{AB} \\ |X_B||Y_A|\cos\beta_{AB} & |Y_B||Y_A|\cos\beta_{AB} & |Z_B||Y_A|\cos\beta_{AB} \\ |X_B||Z_A|\cos\alpha_{AB} & |Y_B||Z_A|\cos\alpha_{AB} & |Z_B||Z_A|\cos\alpha_{AB} \end{bmatrix} \quad (2.4)$$

A ve B koordinat sistemleri eş merkezli olduğu durumlar için herhangi bir P noktasının A koordinat sistemine göre B koordinat sistemi cinsinden gösterimi Şekil 2.1'de gösterilmiştir. Matematiksel gösterimi ise aşağıdaki gibi ifade edilir.

$${}^A P = {}^A R {}^B P \quad (2.5)$$

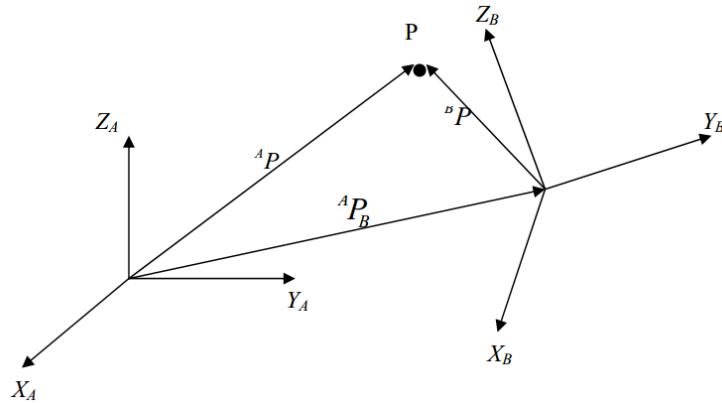


Şekil 2.1. Eş merkezli iki koordinat sisteminin gösterimi [17]

2.1.3. Öteleme İşlemi

Paralel mekanizmalar baz kısımları sabit uç eyleyicinin bulunduğu kısımları ise hareketli sistemlerdir. Hareketli platformun baz platforma göre bir yönelimi bulunmaktadır. Uç eyleyicinin bulunduğu nokta hareketli platformun koordinat sistemine göre belirtilmesine rağmen baz platformun koordinat sistemi farklıdır. Bu durumda hareketli platformun koordinat sisteminin baz platform koordinat sistemine göre ifade edilmesi gerekmektedir. Diğer bir deyişle eksenleri çakışık olmayan iki koordinat sisteminin birbirine göre ifade edilmesidir. Farklı merkezli iki koordinat sistemi Şekil 2.2’de gösterilmiştir. Matematiksel olarak aşağıdaki şekilde ifade edilir.

$${}^A P = {}^A_B R {}^B P + {}^B P \quad (2.6)$$



Şekil 2.2. Yönelimleri ve merkezleri farklı koordinat sistemleri [17]

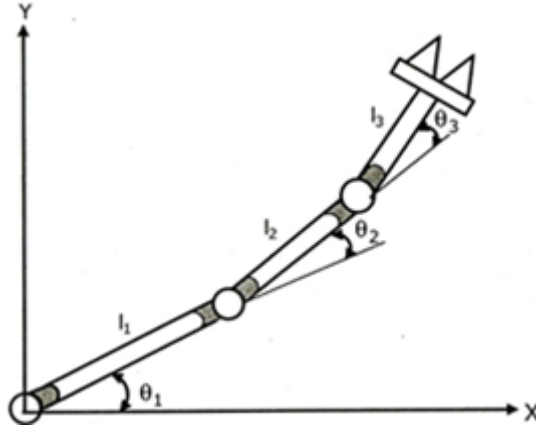
Yönelim ve merkezleri farklı iki koordinat sistemi noktasının birbiri cinsinden ifadesi tek bir transformasyon matrisi altında aşağıdaki şekilde toplanmaktadır.

$$\begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A R & {}^B P_B \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^B P \\ 1 \end{bmatrix} \quad (2.7)$$

2.2. İleri Kinematik

İleri kinematik hesaplamaları bölüm 2.1. Kinematiksel Kavramlar'da belirtilen D-H yöntemi üzerinden anlatılmıştır. Kinematik bilimi robotik sistemlerin hareket özelliklerini inceler. İleri kinematik eklem değişkenlerini girdi olarak alarak her eklem için transformasyon denklemleri ardışık olarak hesaplar. ${}^{i-1}_i T$ dönüşüm matrisi her bir eklem için oluşturularak zincir şeklinde son ekleme kadar hesaplanır. Tüm eklemler için yapılan bu hesaplama sonucu robotun uç işlevcisinin nihai konumunu verir. Diğer bir deyişle ileri kinematik eklem değişkenlerinin açı veya uzunluklarının verildiği, çıktı olarak konumun hesaplandığı kinematik hesabdır. Şekil 2.3'te 3 eklemli bir seri robotun çizimi verilmiştir. Bu seri robot için ileri kinematik hesabı ${}^0_N T$ genel transformasyon matrisinin $N=0 \dots N$ 'e kadar olan transformasyon matrislerinin ardarda çarpılması ile aşağıdaki şekilde elde edilir.

$${}^0_N T = {}^0_1 T {}^1_2 T {}^2_3 T \dots {}^{N-1}_N T \quad (2.8)$$



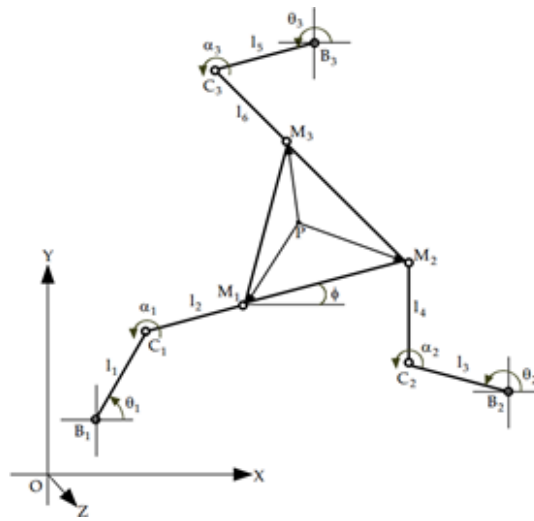
Şekil 2.3. Üç eklemli seri robot [17]

Paralel robotlarda ise ileri kinematik çözümleri seri robotlara oranla çok daha fazla karmaşıktır. Seri robotlarda tek bir transformasyon dizi çözümü ile sonuca ulaşılırken paralel robotlarda birden fazla seri yapının zincirleme eşitliklerinin çözümünün uyumluluk içerisinde hesabını gerektirir. Şekil 2.4'te 3 serbestlik dereceli bir paralel robot gösterilmiştir. Hareketli platformun köşe noktaları M_i olarak belirtilmiştir. M_i noktalarına giden iki farklı yol bulunur. Yolların matematiksel karşılaştırması denklem

2.7’de gösterilmiştir. Birincisi robot kolları üzerinden ilgili eklemlerin transformasyon matrisleri üzerinden bulunan ${}^{M_i}_n T^1$ matrisidir. İkinci yol ise P noktasının ana koordinat sistemine göre konumu ve yönelimi üzerinden M_i noktalarına ulaşılmasını ifade eden ${}^{M_i}_n T^2$ matrisidir. Bu iki farklı yoldan elde edilen eşitlikler birbirine eşitlenerek P konum vektörü denklemden çekilerek hesaplanır [18].

$$OB_i + B_i M_i = OP + P M_i \quad (2.9)$$

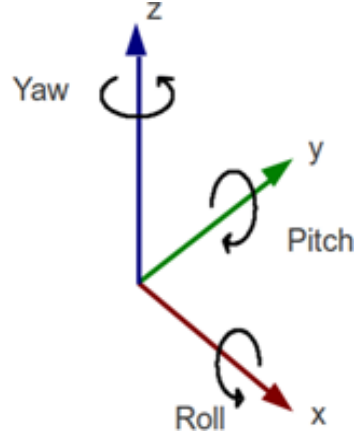
$${}^{M_i}_n T^1 = {}^{M_i}_n T^2 \quad (2.10)$$



Şekil 2.4. 3 DOF RRR Robot [18]

2.3. Ters Kinematik

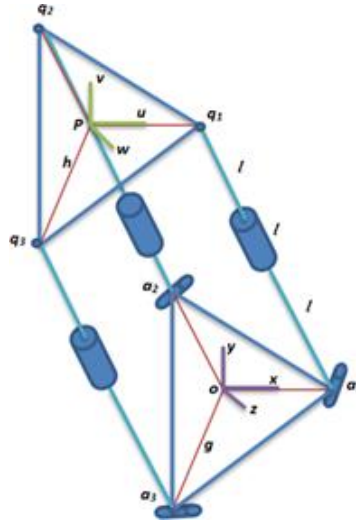
Ters kinematik robotun gideceği noktadan yola çıkarak eklem açılarının hesaplandığı kinematik yöntemidir. Ters kinematik yöntemi, kompleks ileri kinematik hesaplarına göre işlemleri basite indirgediği için özellikle paralel robotlarda anlam kazanmaktadır. Varılacak konum ya da açının bilinmesi ile eklemlerin açı ve uzunluk değişkenlerine ulaşıldığı yöntemdir. Paralel robotlarda bu girdi Roll-Pitch-Yaw açısı olan hareketli platformun sabit platforma göre olan yönelimidir. Roll-Pitch-Yaw açı seti olarak bahsedilen platformlar arasındaki yönelim farklı Şekil 2.5’te gösterilmiştir. Çıktıları ise robotun nihai konumuna ulaşmasını sağlayan kol uzunlukları ve eklem açılarıdır. Bu tezin çalışma amacına uyması açısından yapılacak ters kinematik hesapları 3-UPS robot için hesaplanmıştır.



Şekil 2.5. Roll-Pitch-Yaw açı seti [19]

Şekil 2.6’da tipik bir 3-UPS robot gösterilmiştir. Robot altta sabit platformu, her birinde bir dönel, prizmatik ve kübik eklem ile hareketli platforma bağlanmış 3 kolu mevcuttur. Kol uzunlukları aktif bir prizmatik eklem yardımı ile değiştirilmektedir. Baz platforma kollar 120 derecelik açı aralıkları ile yerleştirilmiştir [20].

Ters kinematik hesaplarında sabit ve aktif platform için iki farklı koordinat sistemi tanımlanır. Platformların köşe noktaları kendi koordinat sistemlerine göre belirlenir.



Şekil 2.6. 3-UPS robot

Şekil 2.6’da belirtilen koordinat sistemlerine ve köşe noktalarına göre sabit platformun köşe noktaları a_i ve hareketli platformun köşe noktaları q_i için sıfır pozisyonunda konum denklemleri aşağıdaki şekilde verilir.

$$a_1 = \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix} \quad a_2 = \begin{bmatrix} -g/2 \\ \sqrt{3}g/2 \\ 0 \end{bmatrix} \quad a_3 = \begin{bmatrix} -g/2 \\ -\sqrt{3}g/2 \\ 0 \end{bmatrix} \quad (2.11)$$

$$q_1 = \begin{bmatrix} h \\ 0 \\ 0 \end{bmatrix} \quad q_2 = \begin{bmatrix} -h/2 \\ \sqrt{3}h/2 \\ 0 \end{bmatrix} \quad q_3 = \begin{bmatrix} -h/2 \\ -\sqrt{3}h/2 \\ 0 \end{bmatrix} \quad (2.12)$$

Hareketli platformun koordinat sistemine göre köşe noktaları q_i , b_i olarak ifade edildiğini varsayalım. Bu durumda sabit platforma göre 3-UPS robotun sabit genel konum denklemi ise aşağıdaki gibi verilir.

$$q_i = P + {}^A_B R^A b_i \quad (2.13)$$

Ters kinematik çözümü için Roll-Pitch-Yaw açı setine göre dönme rotasyon matrisinin hesaplanması gerekmektedir. Bu matris x, y, z eksenlerindeki toplam dönme olarak ifade edilir.

$$R_{xyz} = R_z(Q_z) R_y(Q_y) R_x(Q_x) \quad (2.14)$$

Dönme matrisleri sırası ile yazılırsa;

$$R_{xyz} = \begin{bmatrix} \cos(Q_z) & -\sin(Q_z) & 0 \\ \sin(Q_z) & \cos(Q_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(Q_y) & 0 & \sin(Q_y) \\ 0 & 1 & 0 \\ -\sin(Q_y) & 0 & \cos(Q_y) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\sin(Q_x) & \cos(Q_x) \\ 0 & \sin(Q_x) & \cos(Q_x) \end{bmatrix} \quad (2.15)$$

Rotasyon matrisleri art arda çarpıldığında;

$$R_{xyz} = \begin{bmatrix} \cos(Q_z) \cos(Q_y) & \cos(Q_z) \sin(Q_y) \sin(Q_x) - \sin(Q_z) \cos(Q_x) & \cos(Q_z) \sin(Q_y) \cos(Q_x) + \sin(Q_z) \sin(Q_x) \\ \sin(Q_z) \cos(Q_y) & \sin(Q_z) \sin(Q_y) \sin(Q_x) + \cos(Q_z) \cos(Q_x) & \sin(Q_z) \sin(Q_y) \cos(Q_x) - \cos(Q_z) \sin(Q_x) \\ -\sin(Q_y) & \cos(Q_y) \sin(Q_x) & \cos(Q_y) \cos(Q_x) \end{bmatrix} \quad (2.16)$$

Hareketli platformun sabit platforma göre olan dönme matrisi aşağıdaki şekilde özetlenirse;

$$R_{xyz} = \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix} \quad (2.17)$$

Denklem 2.11'deki konum denkleminde yerine konursa, robotun hareketli platformunun yeni konumları sırası ile Denklem 2.16, 17 ve 18'de verilmiştir.

$$q_i = P + {}^A_B R^A b_i \quad (2.18)$$

$$q_1 = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix} \begin{bmatrix} h \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} P_x + U_x \cdot h \\ P_y + U_y \cdot h \\ P_z + U_z \cdot h \end{bmatrix} \quad (2.19)$$

$$q_2 = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix} \begin{bmatrix} -h/2 \\ \sqrt{3}h/2 \\ 0 \end{bmatrix} = \begin{bmatrix} P_x - U_x \frac{h}{2} + V_x \frac{\sqrt{3}h}{2} \\ P_y - U_y \frac{h}{2} + V_y \frac{\sqrt{3}h}{2} \\ P_z - U_z \frac{h}{2} + V_z \frac{\sqrt{3}h}{2} \end{bmatrix} \quad (2.20)$$

$$q_3 = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix} \begin{bmatrix} -h/2 \\ \sqrt{3}h/2 \\ 0 \end{bmatrix} = \begin{bmatrix} P_x - U_x \frac{h}{2} - V_x \frac{\sqrt{3}h}{2} \\ P_y - U_y \frac{h}{2} - V_y \frac{\sqrt{3}h}{2} \\ P_z - U_z \frac{h}{2} - V_z \frac{\sqrt{3}h}{2} \end{bmatrix} \quad (2.21)$$

Robotun herhangi bir kol uzunluğu d_i olarak alınırsa, kol uzunluğunu veren eşitlik aşağıdaki gibi olacaktır.

$$d_i^2 = [q_i - a_i]^T [q_i - a_i] \quad (2.22)$$

$$d_i = \sqrt{[q_i - a_i]^T [q_i - a_i]} \quad (2.23)$$

Sonuç olarak ulaşılmak istenen Roll-Pitch-Yaw açı seti ile belirlenen yönelimi robota verebilmek için gerekli kol uzunlukları hesaplanmıştır. Aktif prizmatik eyleyiciler ile robot ters kinematik sonucu istenilen pozisyona konumlandırılır. Bu tez çalışmasında yapılan uygulamada ters kinematik kullanılmıştır. İleri kinematiğe göre işlemleri basite indirgemiş ve robotun kontrol birimi mikro denetleyicide kinematik hesaplamalarında geçen sürede büyük azalma sağlamıştır. Böylece robotun sensörlerden gelen bilgilere tepkime süresi azaltılmış gerçek zamanlıya daha fazla yakınsanmıştır.

3. YÖRÜNGE PLANLAMASI

Robotik sistemlerde uç işlevcinin istenilen pozisyona konumlandırılması robotun en temel problemlerinden biridir. Robot ilk pozisyonundan son konumuna doğrudan gitmez. Bunun sebebi her eklem hareket ekseninin farklılığı, hızlarının farklılığı veya çevredeki bir takım engellerdir. Bu sebeple robotun istenilen konuma ulaşabilmesi için belirli ara noktalardan geçen bir yörünge planlaması yapılması gerekmektedir. İzlenecek olan yoldaki ara noktalar robotun hızı, gidilecek mesafe, ivme gibi değişkenlere göre belirlenir.

Robotun uç eyleyicisinin başlangıç noktasından, varılacak noktaya olan hareketi sırasında yer değiştirme ve dönme açılarının belirlendiği diziye yörünge denir. Yörünge sürekli yol hareketi şeklinde planlanabileceği gibi noktadan noktaya hareket olarak da planlanabilir. Noktadan noktaya (PTP) hareket planlamasında, yörünge ilk ve son adımları tanımlanmış fakat hareketin türü ve zaman planlaması yapılmamıştır. Robot noktadan noktaya hareketi için birden fazla yöntem bulunmaktadır. Bunlar sıralı eklem hareketi, koordine edilmemiş eklem hareketi ve koordineli eklem hareketidir. Sıralı eklem hareketi, bir eklem hareketli robottaki diğer bütün eklemlerin sabit olması ilkesine dayanır. Bu hareket metodu yörünge planlaması hesaplarını basite indirgemiş olsa da hareketin süresini uzatmaktadır. Koordine edilmemiş harekette ise robotun geçiş noktaları arasında yolunun ve hızının koordine edilmediği hareket türüdür. Koordine edilmiş eklem hareketinde ise noktadan noktaya hareket hızı ve yol kontrol edilerek eklemlerin son pozisyonlara aynı anda varmaları sağlanır [21].

Bu tez çalışmasında noktadan noktaya koordineli yörünge planlaması kullanılmıştır. Paralel robotlarda eklemler arası koordinasyon sağlanmaz ise robotun montajını patlatma riski vardır. Bu yüzden tüm eklemler için hareket belirlenen orantılarda olmalıdır. Robotun eklemlerinin belirlenen noktalar dizinde zamanın bir fonksiyonu olacak şekilde aktif eklemler hareketlendirilmiş ve tüm eklemleri aynı orantıda

ilerlemeleri sağlanmıştır. Yörünge planlaması fonksiyonları için bu orantı dönelemler için açı orantısı, prizmatik eklemler için ise kol uzunluğunun orantısıdır.

Noktadan noktaya planlanan yörünge planlamasında ilgili noktalar zamanın bir fonksiyonu olarak belirtilir. Eklem uzayında yörünge planlaması üçüncü veya daha yüksek derecen polinomlar ile ifade edilir [17]. Uç işlevcinin t_0 anındaki başlangıç konumu $\theta(0)=\theta_0$ ifade edilsin. Uç işlevcinin son konumuna ulaşma süresi t_f ve t_f anındaki konumu $\theta(t_f)=\theta_f$ olarak ifade edilsin. Bu şekilde t_0 ve t_f arasındaki noktalar bir polinom vasıtasıyla n adet noktaya bölünür. Başlangıç ve bitiş konumlarında robotun hızları sıfır olması sebebiyle konum fonksiyonlarının başlangıç ve bitiş noktalarındaki türevleri sıfırdır.

$$\theta(0)=\theta_0 \quad (3.1)$$

$$\theta(t_f)=\theta_f \quad (3.2)$$

$$\dot{\theta}(0)=0 \quad (3.3)$$

$$\dot{\theta}(t_f)=0 \quad (3.4)$$

Belirlenen ifadeler üçüncü derece bir polinomun çözümü için yeterlidir. Zamana bağlı üçüncü dereceden bir konum denklemi aşağıdaki şekilde gösterilir.

$$\theta(t)=a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (3.5)$$

t_0 başlangıç anı için denklem yerleştirilirse;

$$\theta(0)=a_0 + a_1 0 + a_2 0 + a_3 0 \quad (3.6)$$

$$\theta(0)=a_0 \quad (3.7)$$

$$\theta_0=a_0 \quad (3.8)$$

t_f başlangıç anı için denklem yerleştirilirse;

$$\theta(t_f)=a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \quad (3.9)$$

$$\theta(t_f)=\theta_f \quad (3.10)$$

Konum denkleminin t_0 anı için türevi alınırsa;

$$\dot{\theta}(0)=a_1+ a_2 \cdot 0+ a_3 \cdot 0 \quad (3.11)$$

$$a_1=0 \quad (3.12)$$

Denklem 3.3'te $\dot{\theta}(0)=0$ olduğu hatırlanmalıdır. Konum denkleminin t_f anı için türevi alınırsa;

$$\dot{\theta}(t_f)=a_1+ a_2 t_f+ a_3 t_f^2=0 \quad (3.13)$$

Denklem 3.4'te $\dot{\theta}(t_f)=0$ olduğu hatırlanmalıdır. Elde edilen $a_0=\theta_0$ ve $a_1=0$ katsayıları kullanılarak a_2 ve a_3 birbirleri cinsinden yazılırlar.

$$\dot{\theta}(t_f)=a_1+ a_2 t_f+ a_3 t_f^2=0$$

$$\dot{\theta}(t_f)=0$$

$$-2a_2 t_f=3a_3 t_f^2 \quad (3.14)$$

$$a_2=-\frac{3a_3 t_f}{2} \quad (3.15)$$

$$a_3=-\frac{2a_2}{3t_f} \quad (3.16)$$

Konum denkleminde $a_2=-\frac{3a_3 t_f}{2}$ yerine konursa;

$$\theta(t_f)=a_0+ a_1 t_f+ a_2 t_f^2+ a_3 t_f^3$$

$$\theta(t_f)=\theta_f, a_1=0, a_0=\theta_0$$

$$\theta_f=\theta_0-\frac{3a_3 t_f}{2} t_f^2+a_3 t_f^3$$

$$(\theta_f-\theta_0)=-\frac{a_3 t_f^3}{2}$$

$$a_3=-\frac{2(\theta_f - \theta_0)}{t_f^3} \quad (3.17)$$

Konum denkleminde $a_3=-\frac{2a_2}{3t_f}$ yerine konursa;

$$\theta(t_f) = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3$$

$$\theta(t_f) = \theta_f, a_1 = 0, a_0 = \theta_0$$

$$\theta_f = \theta_0 + a_2 t_f^2 - \frac{2a_2}{3t_f} t_f^3$$

$$(\theta_f - \theta_0) = \frac{a_2 t_f^2}{3}$$

$$a_2 = \frac{3(\theta_f - \theta_0)}{t_f^2} \quad (3.18)$$

Sonuç olarak üçüncü dereceden bir polinomun katsayıları aşağıdaki şekilde hesaplanır;

$$a_0 = \theta_0$$

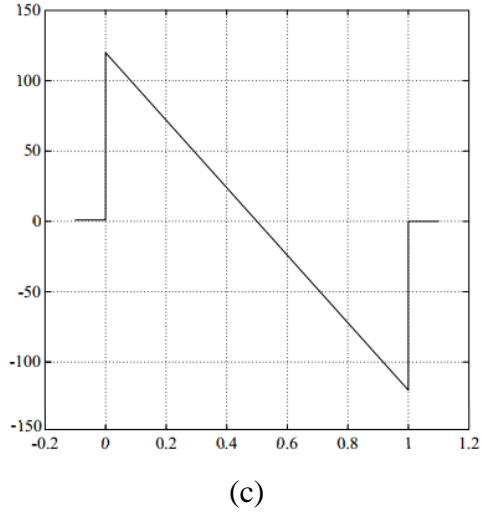
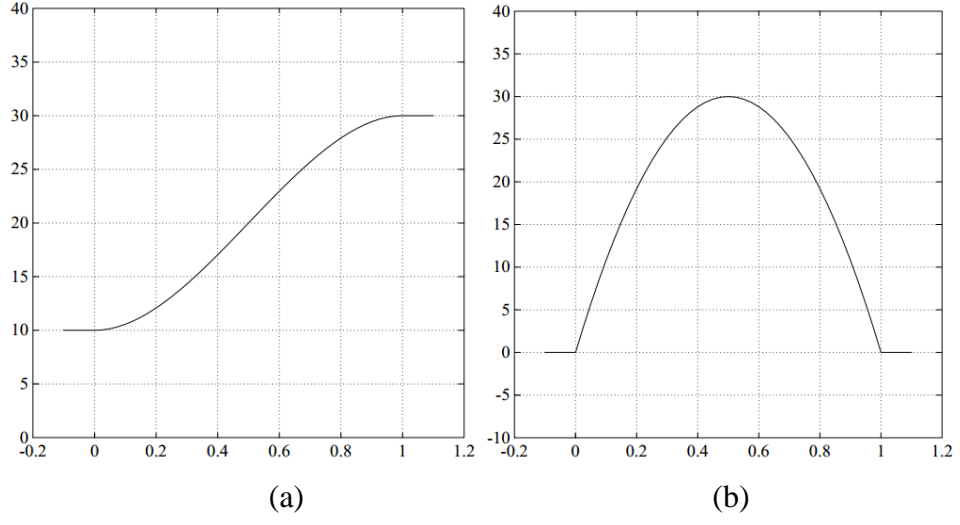
$$a_1 = 0$$

$$a_2 = \frac{3(\theta_f - \theta_0)}{t_f^2}$$

$$a_3 = -\frac{2(\theta_f - \theta_0)}{t_f^3} \quad (3.19)$$

Robotun dönel bir eklemının $\theta_0=10$ derece, $\theta_f=30$ derece ve hareket süresi $t_f=1$ saniye için konum, hız ve ivme grafikleri kübik polinomlu yörünge planlaması ile aşağıdaki şekilde oluşur.

Yörünge planlamasında kullanılan kübik polinom ile elde edilen konum, hız ve ivme grafiği Şekil 3.1’de gösterilmiştir. Üçüncü dereceden kullanılan polinom ile elde edilen konum kübik polinomun gereği iki tepe noktalıdır. Polinomun türevinden yola çıkarak elde edilen hız grafiği tek tepe noktalı artan ve azalan şekildedir. Benzer şekilde konum denkleminin ikinci dereceden türevi lineer şekilde ivmeyi verir.



Şekil 3.1. Kübik yörünge planlaması a) Konum, b) Hız, c) ivme grafiği

4. ROBOTUN BİLGİSAYAR ORTAMINDA MODELLENMESİ

Bu tez çalışması kapsamında 3-UPS, 1-RU paralel robotun bilgisayar ortamında elektriksel simülasyonu ve fiziksel uygulaması yapılmıştır. Bu bölümde robotun kontrolünü sağlayan elektronik donanımının tanıtımı ve bilgisayar ortamında simülasyonundan bahsedilecektir.

4.1. Robotun Elektronik Birimleri

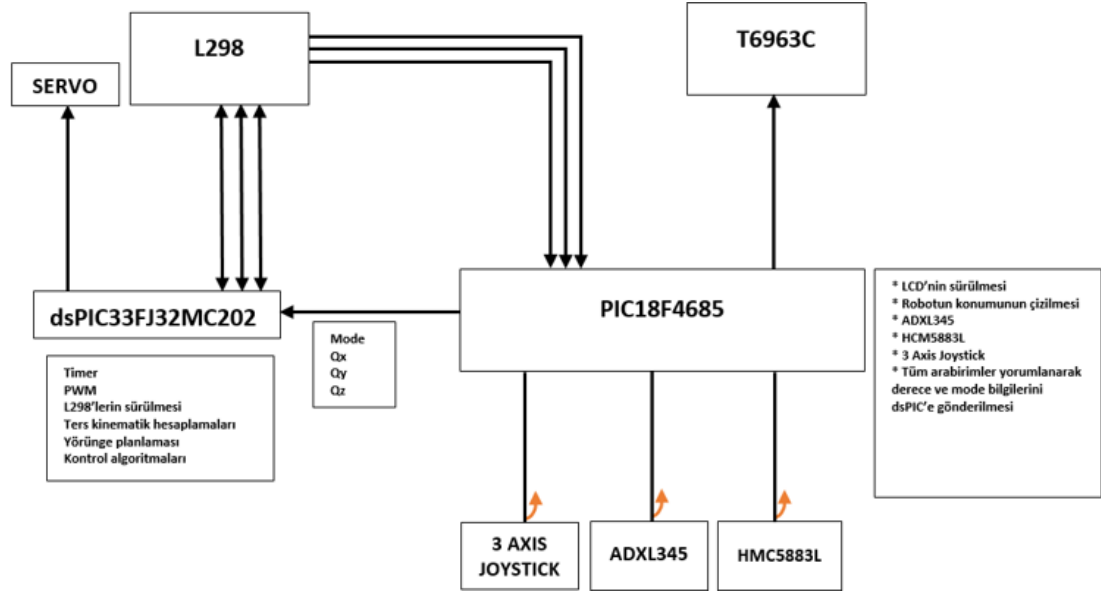
Robotun kontrol ve sürücü bilgi akışı için blok diyagramı Şekil 4.1’de gösterilmiştir. Robot çevresini dinleyen sensörler, sensörden aldığı bilgiyi yorumlayan ve robotun gideceği konumu belirleyen kontrol birimi, kontrol biriminden gelen konum bilgisine göre robotun aktif eklemlerini süren sürücü birim ve kullanıcıyı bilgilendirme amaçlı arayüzden oluşmuştur.



Şekil 4.1. Robotun elektronik bilgi akışı blok diyagramı

Kontrol birimi enerjilendirilmesinin ardından kendisine yüklenen koda göre gerekli konfigürasyon ayarlarını yapar. Sistemde bulunan sensörlerle iletişimi kurar. Bilgilendirme arabirimleri vasıtasıyla kullanıcının anlayacağı dilde robot değişkenleri ve sensörlerin okuduğu değişkenler hakkında bilgi verir. Sensörlerden gelen bilgileri yorumlayarak robotun kinematik hesaplarını ve yörünge planlamasını yapar. Yörünge planına göre sürücü devrenin anlayacağı dilde konum bilgisini sürücü

devre ile paylaşır. Bilgilendirme birimi, kullanıcının anlayacağı dilde robot değişkenlerini ve sensör bilgilerini görüntüler. Sensörler, sürekli dış çevreyi dinleyerek ortam koşullarını robotun kontrol ünitesine bildirir. Sürücü devre ise kontrol ünitesinden gelen bilgiye göre robotun eklemlerinin konumlanması için ilgili aktif eklemleri sürer.



Şekil 4.2. Robotun arabirimleri

3-UPS, 1-RU robotun arabirimleri ve bilgi akış yönleri Şekil 4.2’de gösterilmiştir. Robotun kontrol birimi ikiye bölünerek sensörler ve bilgilendirme arabirimleri ile haberleşme bir tarafta, kinematik hesapları ve yörünge planlamasının yapıldığı matematiksel işlemler ve eklemlerin sürülmedi bir tarafta toplanmıştır. Robota bağlı her bir arabirimle haberleşmesi ve bilgiyi işlemesi için ihtiyaç duyduğu belli bir zaman aralığı vardır. Şekil 4.2’de gösterilen paralel yapı sensör ve bilgilendirme işlemlerini bir işlemci yüklenirken, kompleks matematiksel işlemleri diğer işlemci üstlenmektedir. Kısacası robotun tepkime süresini düşürecek olan işlemler bölünerek robotun gerçek zamanlıya yakın derecede tepki vermesi sağlanmıştır.

4.1.1. Çevresel birimler

3-UPS, 1-RU robotun hareketi çevresel birimlerden beslenmektedir. Bunun nedeni robotun birden fazla çalışma modunda birden fazla amaç ile çalıştırılabilmesini sağlamaktır.

4.1.1.1. 3 Axis joystick

Robotun tepki ve kinematiğinin belirgin bir şekilde izlenebilmesi için 3 eksen etrafında dönebilen ve bu açıları analog geri besleme ile verebilen bir joystick kullanılmıştır. Robotun teorik olarak gerçekleştirilmesinin yanında teorik elde edilen sonuçların birebir uygulaması joystick ile deneysel olarak sağlanmıştır. Joystick x, y, z eksenin 50 derecelik dönme imkanı sağlamakta ve üç eksendeki açı değerini kutupları arasına uygulanan gerilim ile orantılı olarak geri besleme vermektedir. ADC işlemine tabi tutulan değer derece cinsinden kinematik hesaplarına tabi tutulur. Uygulamada kullanılan joystick Şekil 4.3'te gösterilmiştir.

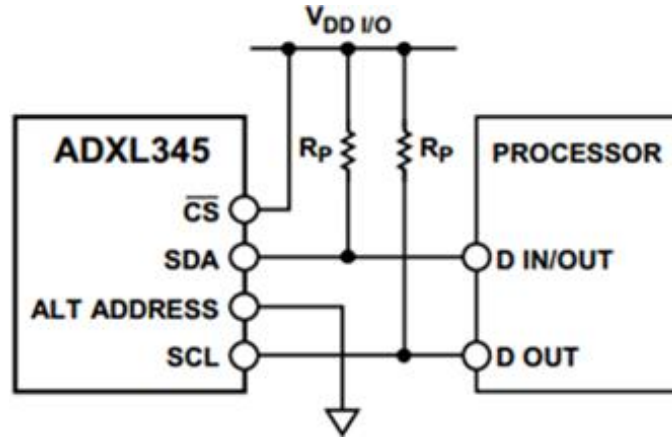


Şekil 4.3. 3 axis joystick [22]

4.1.1.2. ADXL345 ivme sensörü

3-UPS, 1-RU robotun ikinci bir uygulaması olarak bir açı sensörü ile algılanan açı değerinin işlemcide yorumlanması ile robotun konumunun değiştirilmesi olarak uygulanmıştır. Robot, mobil bir platform üzerinde iken bir sensör vasıtasıyla x ve y eksenlerindeki açı değişiminin algılanması sağlanmış ve hareketli platformun sürekli deniz seviyesine paralel kalması sağlanmıştır. Bu gibi uygulamalar kimyasal taşımacılık, lojistik gibi alanlarda sıkça kullanılmaktadır. Bu tez çalışmasında benzer uygulamada açı ölçümü Analog Devices firmasının ürünü olan ADXL345 ivme sensörü ile gerçekleştirilmiştir.

ADXL345 dijital ivme sensörü 1 dereceden küçük değişiklikleri algılayabilir. Şekil 4.4'te ADXL345 entegresinin bağlantı şeması verilmiştir. Bu sensör I²C seri protokolü ile slave modunda bağlı bulunduğu master işlemci ile haberleşebilmektedir. Algıladığı açı bilgisini içerisinde tanımlı olan belirli register alanlarında sekizer bitlik 2 byte halinde tutar. Bu byte'ların ilki LSB, diğeri ise MSB olarak yorumlanır. Bağlı bulunduğu hat üzerinden ilgili master bu register alanlarında bulunan 8 bitlik veriler okunur. Bu veri mikro denetleyicide LSB ve MSB birleştirilerek 16 bitlik veri halinde yorumlanarak kinematik denklemlerine girdi olacak formatta derece cinsinden açı haline dönüştürülür.

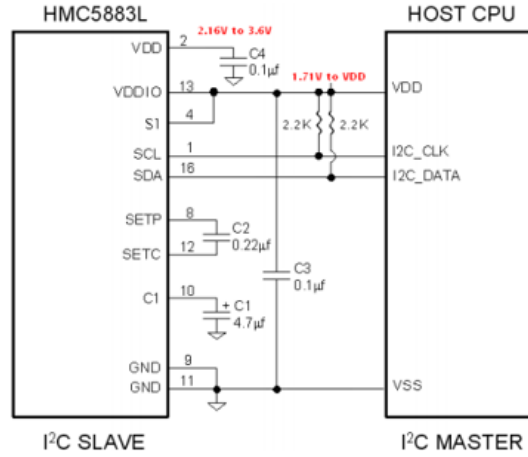


Şekil 4.4. ADXL345 bağlantı şeması [23]

4.1.1.3. HMC5883L dijital pusula

HMC5883L Honeywell firmasının ürettiği dijital pusula sensörüdür. 1 derece hassasiyeti olan bu dijital sensör yerin doğal magnetik alanından faydalanarak analog pusulalar ile aynı çalışma mantığına sahiptir. Cep telefonları, diz üstü bilgisayarları, navigasyon cihazları gibi birçok elektronik cihazda kullanılmaktadır. Bu tez çalışması kapsamında yapılan sensör modu uygulamasında z eksenindeki dönme açısını HMC5883L entegresi sağlamaktadır. ADXL345 entegresine benzer bir register yapısı bulunan bu entegre z eksenindeki açı bilgisini 8 bitlik iki register halinde LSB ve MSB olarak tutmaktadır. Sensörün register yapısı Tablo 4.1'de gösterilmiştir. I²C protokolü ile bağlı bulunduğu master cihaz ile haberleşebilmektedir. İlgili master cihaz devre enerjilendiğinde gerekli konfigürasyon bitlerini ilgili register adreslerine yazar. Belirtilen konfigürasyona göre HMC5883L entegresi ölçüm yapmaya başlar. Z eksenindeki açı bilgisi entegrenin ilgili register adresinden master cihaz tarafından 8

bitlik veri halinde LSB ve MSB olarak alınarak işlemci tarafında 16 bitlik veri haline getirilir. İşlemcide yorumlanan bilgiye göre kinematik denklemlerinde kullanılmak üzere derece cinsinden açıya çevrilir. Bu tez kapsamında yapılan uygulamada HMC5883L, PIC18F4685 ile 100kHz’te I²C seri protokolü ile haberleşmektedir. HMC5883L ve mikroişlemci bağlantı şeması Şekil 4.5’te gösterilmiştir.



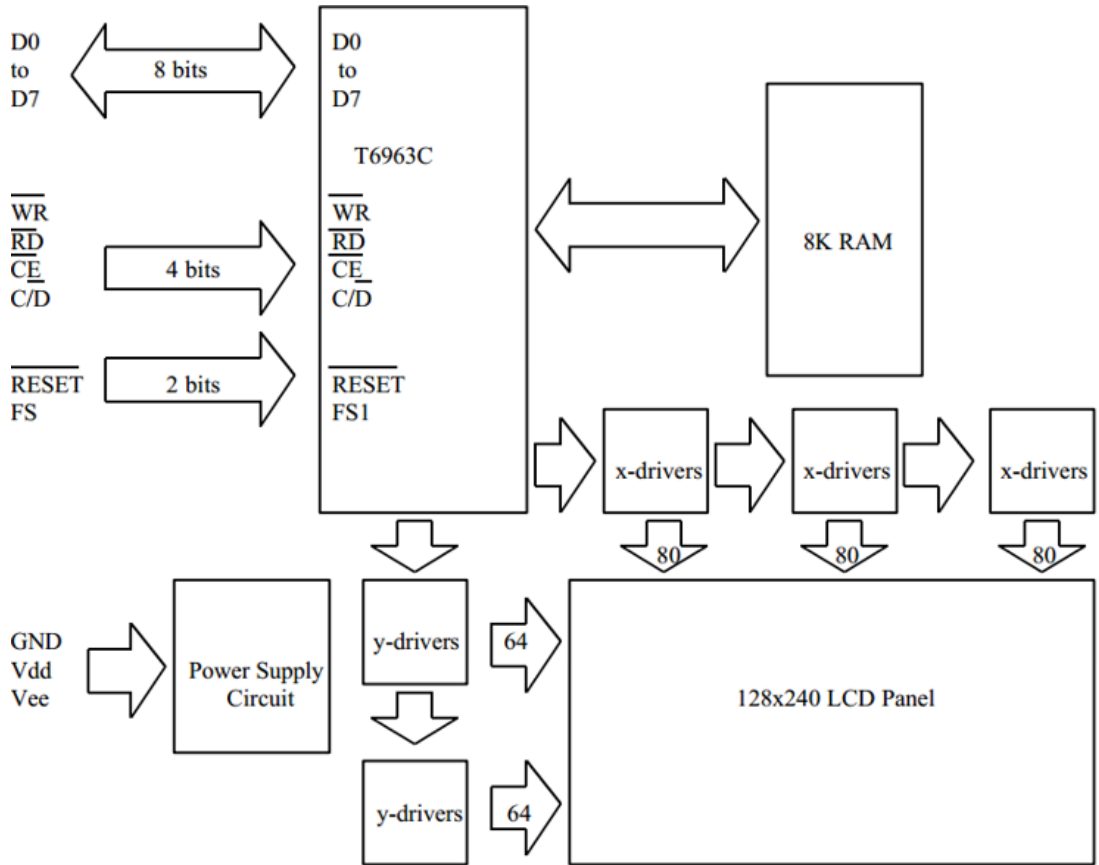
Şekil 4.5. HMC5883L entegresinin bağlantı şeması [24]

Tablo 4.1. HMC5883L Register listesi [24]

Address	Name	Acces
00	Configuration Register A	Read/Write
01	Configuration Register B	Read/Write
02	Mode Register	Read/Write
03	Data Output X MSB Register	Read
04	Data Output X LSB Register	Read
05	Data Output Z MSB Register	Read
06	Data Output Z LSB Register	Read
07	Data Output Y MSB Register	Read
08	Data Output Y LSB Register	Read
09	Status Register	Read
10	Identification Register A	Read
11	Identification Register B	Read
12	Identification Register C	Read

4.1.1.4. T6963C grafik LCD mikro işlemcisi

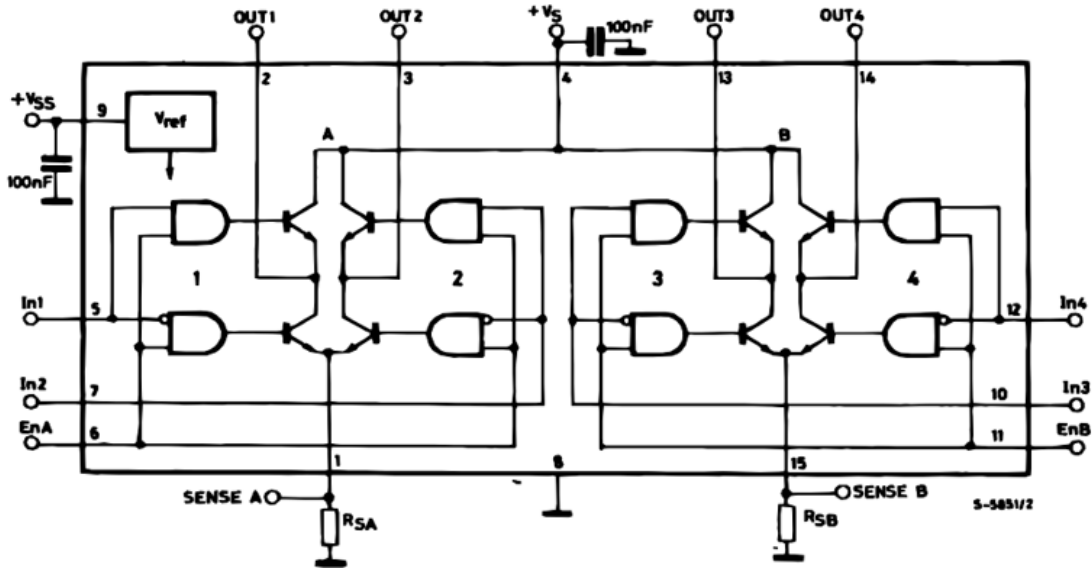
T6963C mikro işlemcisi küçük ve orta ölçekli grafik LCD modüllerini kontrol etmek için dizayn edilmiştir. Projelerde kullanılan 8 bitlik mikro işlemcilerden bir farkı yoktur. T6963C mikro işlemcisi, çalışmada kullanılan PIC18F4685 ile 240x128 karakterlik GLCD VRAM'i arasındaki haberleşmeyi sağlayan arayüz görevi görmektedir. GLCD için gerekli olan zamanlama ve veri sinyalizasyonunu sağlar. Sahip olduğu 128 karakterlik Character-Generator özelliği ile 64Kb'a kadar RAM belleği olan GLCD birimlerinin kontrolünü sağlayabilir. Birçok LCD tipini desteklemesinin yanında grafik ve text verilerinin konsolide edilerek LCD'ye aktarılmasını sağlar. T6963C mikro işlemcisinin panel LCD ile çalışmasının blok diyagramı Şekil 4.6'da gösterilmiştir.



Şekil 4.6. T6963C mikro işlemcisi ve LCD blok diyagramı [25]

4.1.1.5. L298 motor sürücü entegre

L298 entegresi yüksek akım, yüksek gerilim kapasiteli çift taraflı motor sürücü entegredir. Mikro denetleyiciden gelen TTL lojik seviyelerine göre içinde bulundurduğu H-bridge devreleri ile bağlı olduğu motorları sürer. L298 entegresinin iki adet beslemesi vardır. İlki TTL seviyesinde kendisine gelen sinyalleri algılanabilmesi içindir. Diğeri ise kendisine bağlı bulunan motorları TTL seviyesinden yüksek gerilim ve akımlarla sürebilmesi içindir. Şekil 4.7’de L298 entegresi içerisinde bulunan H-Bridge devresi verilmiştir. Lojik kontrolü sağlayan enable uçları ile input uçları And kapısı ile birbirine bağlanmıştır. İlgili hattın +Vdd durumunda olabilmesi için enable ucunun ve input ucunun +Vdd gerilimi uygulanması gerekmektedir. Input ucu ve enable ucu +Vdd gerilimi uygulanan uçlar ilgili transistörün iletme geçmesini ve motorun dönmesini sağlar. L298 entegresi içinde bulundurdu iki adet H-Bridge entegresi ile 46V’a kadar iki adet motoru aynı anda çift taraflı sürme kapasitesine sahiptir.



Şekil 4.7. L298 H-Bridge devresi [26]

Tez çalışmasında L298 ile yapılmış sürücü devreler kullanılmıştır. Sürücü devreleri detayları Ekler 2 bölümünde verilmiştir.

4.1.2. Kontrol birimi

3-UPS, 1-RU robotun kontrol birimi olarak iki adet mikro denetleyici kullanılmıştır. Çevresel birimler ile haberleşme için Microchip firmasının PIC18F4685 mikro denetleyicisi kullanılmıştır. Bu işlemcinin tercih edilmesinin sebebi 40 pine sahip olması, yüksek RAM, ROM ve Flash disk belleğinin bulunmasının yanında çalışmada ihtiyaç duyulan I²C (Inter Integrated Circuit) gibi protokolleri desteklemesi yeterli sayıda ADC pini bulunmasıdır.

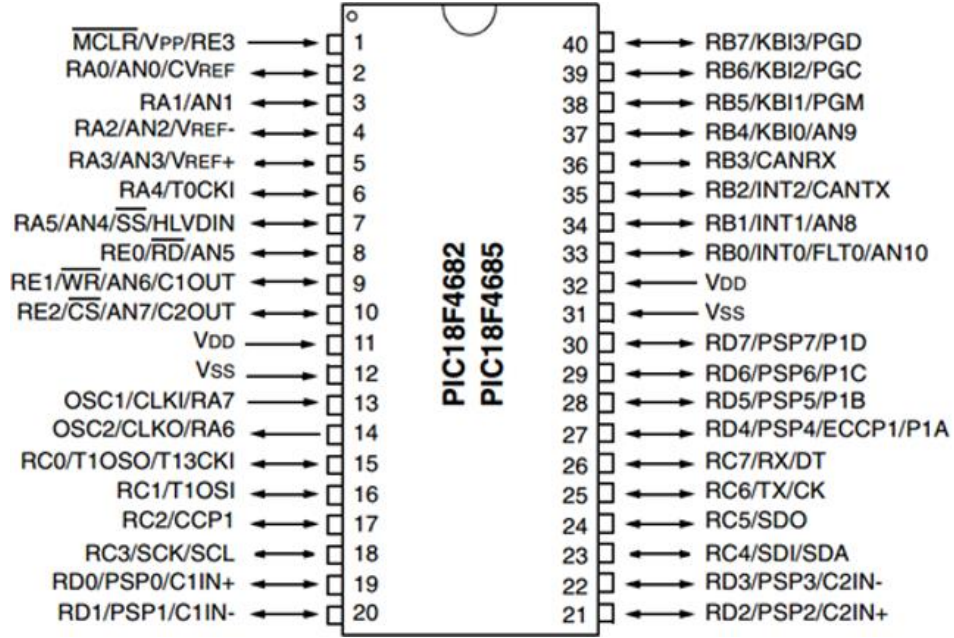
PIC18F4685 8 bitlik mimarisi ile 10MIPS hıza çıkabilmektedir. Bu durumda PLL (Phase Lock Loop) denen frekans çarpıcı özelliğinde kullanılması gerekmektedir. Uygulamada bu özellikte kullanıldığında grafik LCD ile haberleşmesinde problem yaşanmıştır. Bu yüzden standart hızı 5 MIPS'te kullanılmıştır. PIC18F4685 4 çevrimde bir işlem yapabilme yeteneğinde olup 20MHz harici bir kristal osilatör ile çalışmaktadır. Tablo 4.1'de 18F4685 işlemcisinin özellikleri görünmektedir. Çevresel birimler ile haberleşmede ve GLCD'yi sürmede kullanılacağı için pin sayısı önem kazanmaktadır. Bu nedenle 40 pinlik bir işlemci tercih edilmiştir. Robotun çalıştırılmasında birden fazla mod uyarlanmış bunun için birden fazla dış birim ihtiyacı doğmuştur.

Tablo 4.2. Kontrol birimi mikro denetleyici özellikleri

Mikrodenetleyici	Volume Pricing	CPU Speed (MIPS)	Pins	Flash Memory Kb	RAM Kb	Timer 8/16 bit	Comparator	EUSART	SPI	I2C	CCP1/ECCP1(PW)	Total ADC
PIC18F4685	\$6.08	10	40	96	3328	1/3	2	1	1	1	2	11
dsPIC33FJ32GP202	\$2.53	40	28	32	2048	1/3	2	1	1	1	2	10

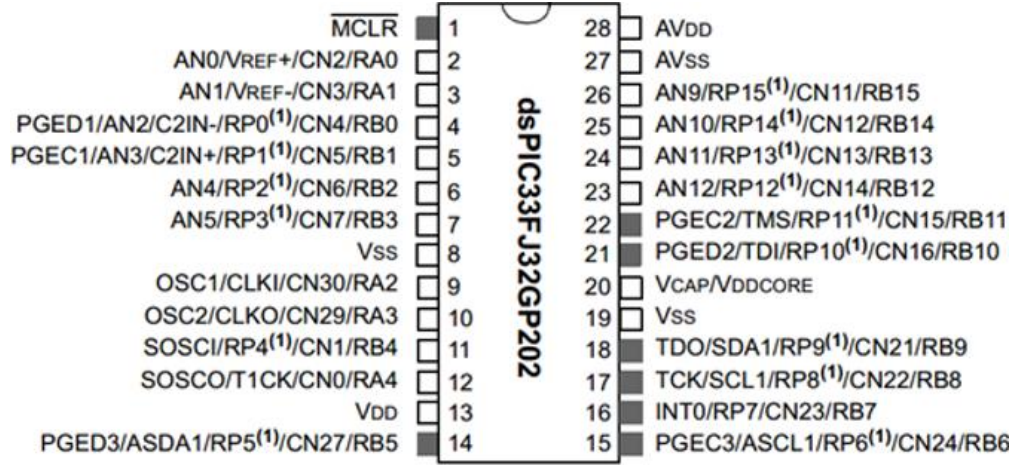
PIC18F4685 mikro denetleyicisinin pin diyagramı Şekil 4.8'de gösterilmiştir. 10 adet ADC pini ile 3-axis joystick ve robot kollarında bulunan 3 adet lineer servodan gelen analog geri besleme bilgisi yorumlanabilmektedir. Entegrenin d portunun tamamı GLCD'ye gönderilen yazım ve çizim bilgisi için ayrılmıştır. Bunun yanında I²C haberleşmesi için SCL ve SDA pinleri ayrılmıştır. Donanımsal olarak bu pinlerin kullanılabileceği gibi yazılımsal I²C modu ile farklı pinlerde haberleşme protokolünde

kullanılabilir. PIC18F4685 mikro denetleyicisi I²C haberleşme protokolünde master rol almakta ve haberleşmenin sürdürülmesi için gerekli olan slave birimlere 100kHz'lik SCL senkronizasyon işaretini göndermektedir.



Şekil 4.8. PIC18F4685 mikro denetleyicisinin pinleri [27]

dsPIC33FJ32GP202 mikro denetleyicisi 18F4685'e göre yeni nesil bir işlemci olup 16 bit mimarisine sahiptir. 2 çevrimde bir işlem yapma yeteneğinin yanında donanımsal çarpıcı özelliği sayesinde kompleks matematiksel işlemleri tek komutta yerine getirebilmektedir. PIC18F4685'te ise bu özelliğin olmamasından dolayı matematiksel çarpma ve bölme işlemleri öteleme yöntemi ile yapılmakta ve bu yüzden kinematik hesapları gibi kompleks işlemlerde işlemci gecikmektedir. Çalışmada bu sebepten dolayı dsPIC33FJ32GP202 mikro denetleyicisi kinematik hesaplamaları, yörünge planlaması ve motorların sürülmesi gibi cevap süresinin büyük önem taşıdığı ve yüklü matematiksel işlemin olduğu görevleri yerine getirme de tercih edilmiştir. dsPIC33FJ32GP202 mikro denetleyicisinin genel özellikleri Tablo 4.2'de belirtilmiştir.



Şekil 4.9. dsPIC33FJ32GP202 mikro denetleyicisinin pin yapısı [28]

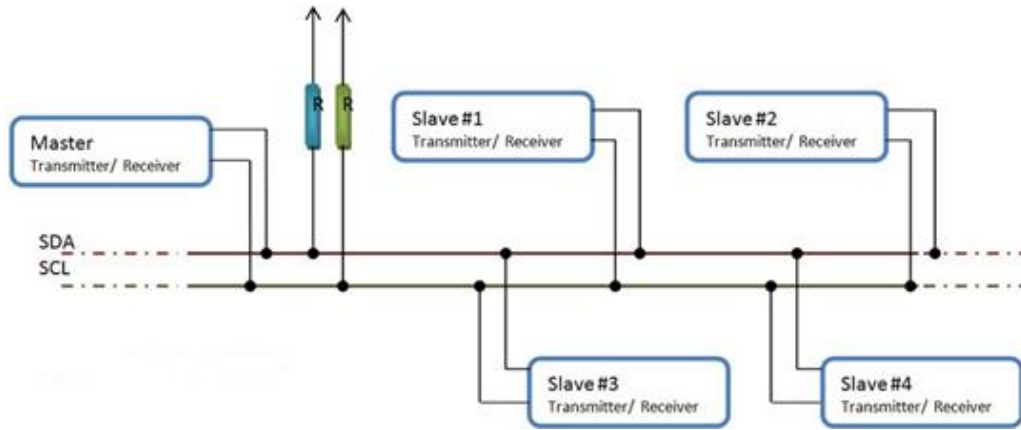
Şekil 4.9’da dsPIC33FJ32GP202 mikro denetleyicisinin pin yapısı verilmiştir. Yörünge planlaması kontrol algoritmalarında kullanılmak üzere her bir motordan analog geri besleme toplamaktadır. Zamanın bir fonksiyonu olarak verilen yörünge planlamasında tüm motorların orantısal olarak ilerlemesini ve montaja zarar gelmemesini sağlar. PIC18F4685’in çevresel birimlerden topladığı verileri derece cinsinden açıya dönüştürülmesinin ardından alır ve kinematik denklemlerinde girdi olarak kullanır. Robotun kinematik denklemlerini milisaniyeler mertebesinde çözümler ve varılacak konum pozisyon bilgilerini üretir. Ardından noktadan noktaya koordineli yörünge planlaması yaparak motorları sürer. I²C haberleşme protokolünde slave cihaz olarak haberleşmek kanalını dinler.

PIC18F4685 her 10 milisaniyede çevresel birimlerden topladığı açı ve mod bilgisini dsPIC33FJ32GP202’ye gönderir. dsPIC33FJ32GP202 gelen bilgiyi yorumlayarak 2 dereceden fazla bir değişme olduğunda kinematik hesaplamalarını ve yörünge planlamasını tekrarlar. Robotu yeni güncel konumuna doğru koordineli bir şekilde sürer. Robot hareket halindeyken sürekli olarak 10 milisaniyede bir gelen konum bilgisini kontrol etmeye devam eder. Belirlenen eşik değerinin üzerinde gelen konum farkı için yukarıda anlatılan süreç tekrarlanır.

4.2. I²C Haberleşme Protokolü

I²C (Inter Integrated Circuit) protokolü, haberleşme hızının çok yüksek olmasının gerekmediği, pin sayısının önemli olduğu uygulamalarda gereksinimi ortaya çıkmıştır. Bu protokol PHILIPS firması tarafından geliştirilmiştir. İlk aşamalarda PHILIPS bu protokolü sadece televizyon ve video cihazlarında kullanmasına rağmen yarı iletken teknolojisinin gelişmesi ile birlikte birçok yarı iletken eleman tarafından kullanılan bir protokol haline gelmiştir. Birçok PIC mikro denetleyicisi içerisinde hem I²C hem de SPI birimini içerisinde barındıran MSSP (The Master Synchronous Serial Port) modülü bulundurmaktadır [29].

3-UPS, 1-RU paralel robot uygulamasında bölüm 4.1'de bahsedilen ADXL345, HMC5883L, PIC18F4685, dsPIC33FJ32GP202 birimleri master ya da slave olarak haberleşme protokolünde yer almaktadır. Tek bir seri hat üzerinden birden fazla cihazın haberleşmesine olanak sağlayan I²C haberleşme protokolü kullanılmıştır. Haberleşme hattı üzerinde cihazların konumlandırılması Şekil 4.10'da gösterilmiştir. 100Kb veri aktarım hızı 3-UPS, 1-RU paralel robot uygulamasında yeterli olmaktadır.

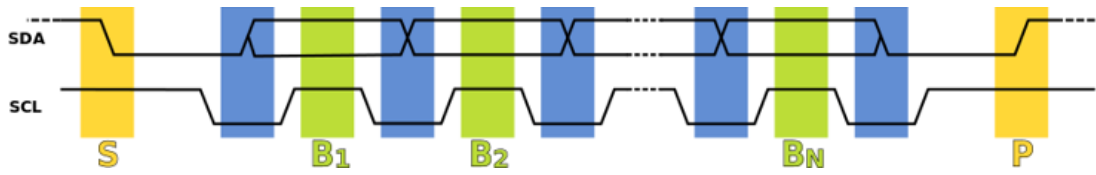


Şekil 4.10. I²C mikrodnetleyici ve elemanların bağlantı şeması [30]

I²C iletişimde master cihaz iletişimi start şartını oluşturarak başlatır. Start şartı SCL lojik 1 geriliminde iken SDA hattının da lojik 0'a çekilmesi ile oluşur. Ardından master cihaz 7 bit slave adresini ve 8. bit olarak Read veya Write işlemlerinden hangisinin yapılacağını belirterek gönderir. 8. bit 0 ise yazma 1 ise okuma işleminin yapılacağı anlamına gelmektedir. Master cihazın gönderdiği adres hangi cihaza ait ise o cihaz seçilmiş olur. Daha sonrasında master cihaz veri bilgilerini gönderir ve ilgili slave

cihazdan alındığına dair ACK alındı biti gönderilir. Bazı durumlarda ACK biti yerine NACK biti gönderilir. Bu durumda başlangıç şartı tekrar sağlanarak haberleşme tekrar edilir. Veri gönderimi tamamlandıktan sonra SDA hattı lojik 1 seviyesine çekilerek haberleşme sonlandırılır. Haberleşme protokolünün ilerleyişi Şekil 4.11’de gösterilmiştir.

Eğer I²C haberleşmesinde, haberleşme hattı üzerindeki tüm cihazlara bilgi göndermek istenirse 0x00 genel çağırma adresi kullanılır. Genel çağırma adresinden sonraki byte hatta iletilecek olan bilgiyi içerir.



Şekil 4.11. I²C iletişimde veri gönderme işlemi [31]

4.3. CCS C Programlama Dili

C Programlama dili 1970 yılı başlarında Dennis Ritchie tarafından AT&T laboratuvarlarında geliştirilmiştir. Ritchie, tasarımcısı Ken Thomson olan B dili üzerine çalışmalarında, B diline yeni fikirler getirerek C dilinin temellerini atmıştır. C dili ilk zamanlar Unix sistemler için kullanılmaktayken zaman içerisinde kişisel bilgisayarların yaygınlaşması ile en fazla tercih edilen programlama dili olmuştur. Günümüzde hemen hemen her türlü sistemde C derleyicileri mevcuttur. 1972 yılında üzerine yoğunlaştığı için C dilinin miladı olarak kabul edilir. 1983 yılında ANSI (American National Standards Institute) tarafından standartlaştırıldıktan sonra, yüksek oranda taşınabilir bir programlama dili olmuştur. Günümüzde birçok işletim sistemi, uygulama, derleyici C veya C tabanlı C++, C# gibi dillerle programlanmaktadır [29].

CCS C dili ANSI standartlarının hemen hemen tüm tanımlamalarını kabul ederek hazırlanmıştır. CCS C dili C dili ile tamamen uyumlu ve aynı formattadır. Bu tez çalışmasında kullanılan dsPIC33FJ32GP202 ve PIC18F4685 mikro denetleyicileri CCS C dili ile programlanmıştır. CCS C dili standart C dili kabullerinin tamamını esas aldığı için mikro denetleyicilerin programlama sintaksı, tanımlanan değişkenler ve türleri, kullanılan döngüler C dili ile birebir uyumludur.

CCS C yüksek seviyeli bir programlama dilidir. Mikro denetleyicilerin kodunun CCS C dilinde yazılmasına rağmen, mikro denetleyicilerin anlayacağı alt seviye makine diline çevrilmesi gerekmektedir. Bu işlem için CCS Inc. (Custom Computer Services Incorporation) firmasının CCS C Compiler 5.015 programı kullanılmıştır. İlgili C program parçacığı derlendikten sonra makine dili olan Hex koduna çevrilmiştir. Hex kodunun oluşturulmasının ardından PIC KIT 2 arabirimi sayesinde ilgili makine kodu mikro denetleyicilere yüklenmekte ve bu sayede mikro denetleyici üzerindeki portlar ve modüller arzu edilen şekilde kullanılmaktadır.

4.4. Robotun Proteus Ortamında Tasarımı ve Simülasyonu

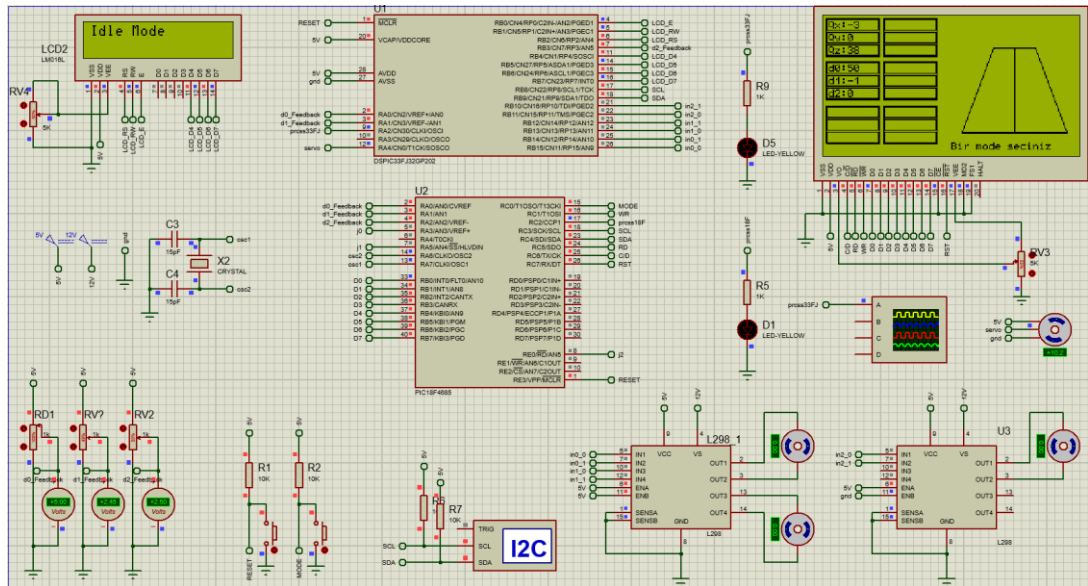
Proteus programı bilgisayar ortamında devre tasarımını ve simülasyonunu mümkün kılan bir uygulamadır. Labcenter firmasının ürettiği Proteus programı, sahip olduğu bir çok elektronik komponent kütüphanesi sayesinde büyük ölçekli projeleri bilgisayar ortamında yapma deneme imkanı sunmaktadır. Mikro işlemcileri sanal ortamda programlanabilir ve gerçek zamanlı simülasyonları koşturulabilir. Kullanıcıya büyük ölçüde zaman tasarrufu sağlan Proteus programı aynı zamanda internet ortamında açık kaynak kodlu zengin kütüphane desteği ile elektronik arşivi genişletilebilmektedir.

Bu uygulamada Proteus 8.0 versiyonu kullanılmıştır. Proteus uygulamasının ISIS modülü ile robotun tüm elektronik komponentlerinin analizi yapılmış, sanal ortamda çalışır hale getirilmiştir. Projenin fiziksel gerçekleştirilmesinden önce mikro denetleyiciler CCS C dili kullanılarak programlanmış komponentler ile uyum içerisinde çalışması sağlanmıştır. Elektronik komponentlerin çalıştırılmasının yanı sıra devre tamamlandıktan sonra baskı devre çizimi Proteus uygulamasının ARES modülünde çizilmiş ve 3 boyutlu görsel analizi yapılmıştır.

4.4.1. Robotun elektronik komponentlerinin simülasyonu

Proteus 8.0 versiyonunun ISIS modülü ile projedeki elektronik komponentlerinin simülasyonu gerçekleştirilmiştir. Şekilde 4.12'de gösterildiği üzere çalışma alanı üzerinde elektronik komponentler yerleştirilmiş ve birbirleri ile olan bağlantıları yapılmıştır. ISIS ortamında LCD, GLCD gibi arabirimlerin bulunması ve gerçek ortam ile aynı koşullarda çalıştırılabilmesi devrenin fiziksel gerçekleştirilmesi öncesinde

kullanıcıya büyük kolaylık sağlamıştır. Bunun yanında ISIS ortamında dijital osiloskop ile sinyaller izlenebilmekte haberleşme ve PWM analizleri rahatlıkla yapılabilmektedir. Çalışmada kullanılan dsPIC33FJ32GP202 ve PIC18F4685 mikro denetleyicileri çalışma alanına yerleştirilmiş ve CCS C dili aracılığı ile oluşturulan HEX kodları işlemcilerde kaydedilmiştir. Bunun ardından gerçek zamanlı uygulamalarda mikro denetleyicilerin çalışması incelenmiş, konfigürasyon ayarları yapılmış ve birbirleri ile I²C protokolü üzerinden haberleşmeleri sağlanmıştır. Kullanıcıya robotun pozisyonu ve yönelimi hakkında bilgi veren LCD'ler simüle edilmiştir. dsPIC33FJ32GP202 işlemcisinin PLL ile çalıştırılması ve haberleştiği birimlerin tepkileri kontrol edilmiştir.

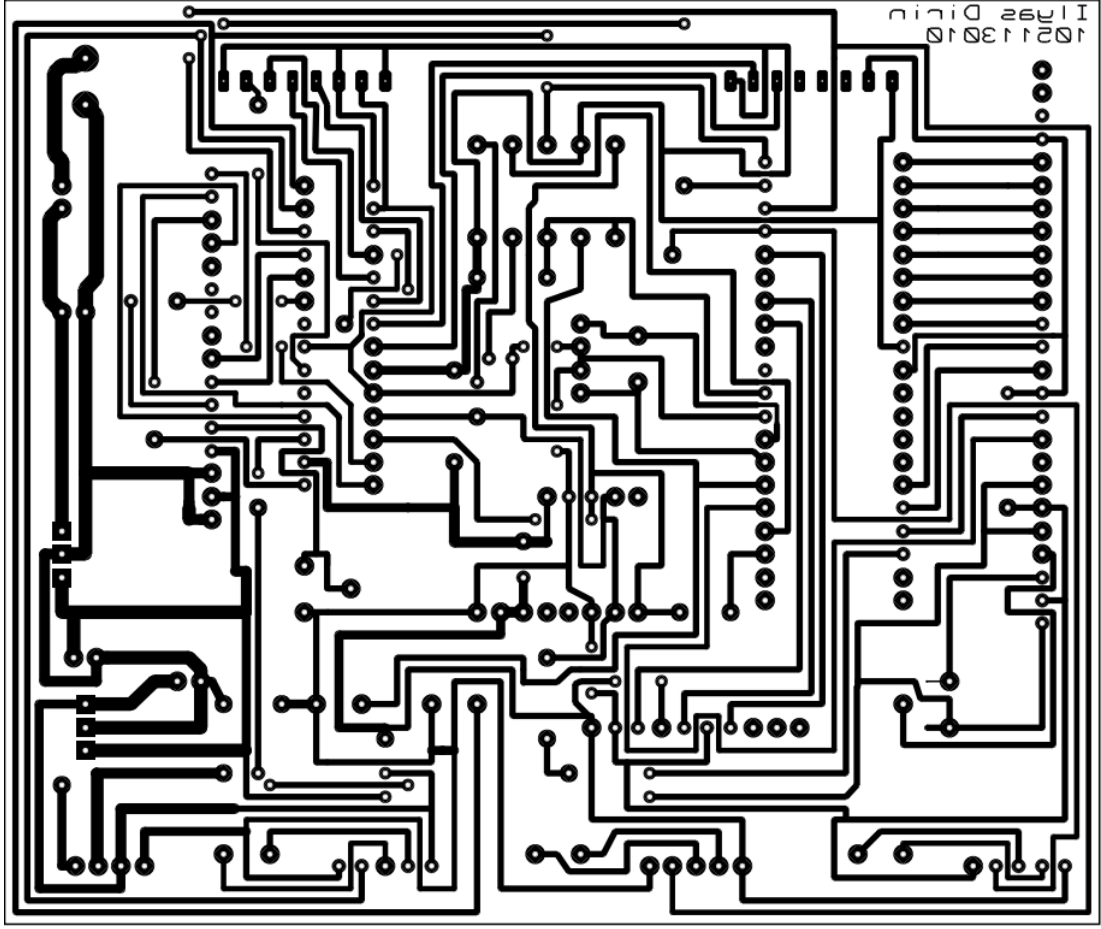


Şekil 4.12. Proteus ISIS ortamında elektronik komponentlerin analizi

Sonuç olarak robotun tüm elektronik komponentlerinin bilgisayar ortamında simülasyonu yapılmış ve sistemin tamamı ayağa kaldırılmıştır.

4.4.2. Robotun PCB tasarımı

Robotun ISIS ortamında elektronik komponentlerinin sağlıklı bir şekilde çalıştırılmasının ardından breadboard ortamında gerçek komponentler ve motorlar ile robotun çalışması sağlanmıştır. Robotun mekanik olarak tasarımı ve elektronik devresinin çalıştırılması Bölüm 5'te detaylı bir şekilde anlatılacaktır. Robotun board ortamında çalışmasının ardından elektronik komponentlerin daha kararlı çalışmasını sağlayacak ve robota ilâştirilecek olan bir PCB tasarlanmıştır. Proteus 8.0



Şekil 4.14. PCB ARES çıktısı

Ares uygulaması kullanıcıya aynı zamanda üç boyutlu component tasarım olanağı sağlamaktadır. Elektronik componentlerin bacak sayısı ve boyutları her zaman componentin kılıfı ile aynı ebatta olamamaktadır. Bu durum sadece bacak sayısı ve boyutu göz önüne alınarak yapılan çizimlerde gerçek montajda problem yaratabilmektedir. Örneğin çalışmada kullanılan PIC mikrodenetleyiciler ile takıldıkları ZIF soketlerinin boyutları arasında ciddi oranda fark bulunmaktadır. ARES ortamında sağlanan 3 boyutlu component tasarım olanağı ile devrede optimum eleman yerleştirme sağlanmaktadır. Dizayn için oluşturulmuş hazır component kılıfları kullanılabilir, harici kütüphaneler bulunup kullanılabilir veya Google Sketch gibi 3 boyutlu çizim programlarında çizim yapılarak yeni kılıflar oluşturulabilir. Çalışmanın ARES 3 boyutlu component dizaynı Şekil 4.15'te, devrenin fiziksel ortamda gerçekleştirilmesi Şekil 4.16'da gösterilmiştir.

5. ROBOTUN FİZİKSEL ORTAMDA UYGULAMASI

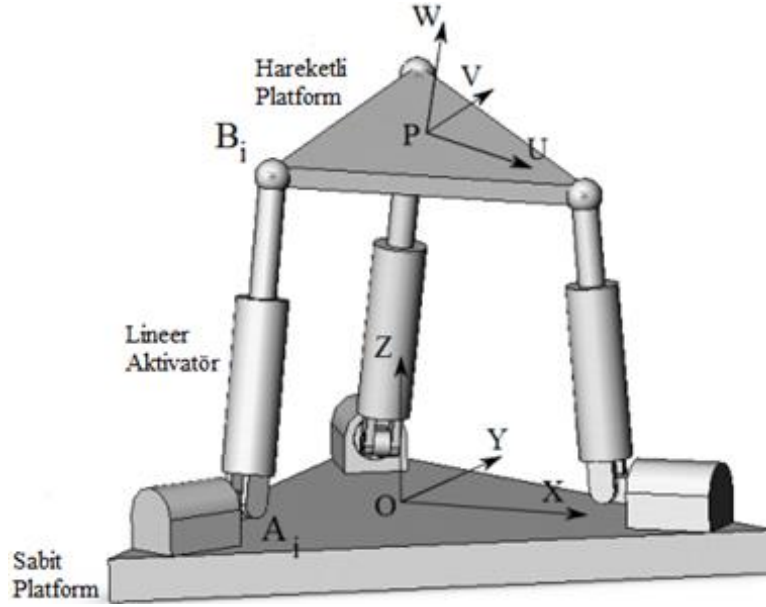
Robotun Bölüm 4'te anlatılan bilgisayar ortamındaki simülasyonu ve elektronik tasarımından sonra mekanik tasarımı gerçekleştirildi. Bu bölümde robotun mekanik tasarımı, eklem yapıları, aktif eyleyiciler, mobil platform, tasarımda karşılaşılan problemler ve çözümleri anlatılacaktır.

5.1. Robotun Mekanik Tasarımı

3-UPS, 1-RU robotun mekanik tasarımı, Lego Technic parçaları ile gerçekleştirilmiştir. Robotun tasarımında klasik Stewart Platformu kabulleri baz alınmış, sabit bir alt platform ile hareketli bir üst platform 120°'er derecelik açılar ile 3 kol ile bağlanarak kapalı çevrim bir paralel robot oluşturulmuştur. Robotun her bir kolu sabit platformda 2 serbestlik dereceli universal eklem, 1 serbestlik dereceli prizmatik eklem, 3 serbestlik dereceli kübik eklem ile hareketli platforma bağlanarak 6 serbestlik dereceli bir yapı oluşturulmuştur. Şekil 5.1'de 6 serbestlik dereceli paralel robotun yapısı ve koordinat sistemleri gösterilmektedir.

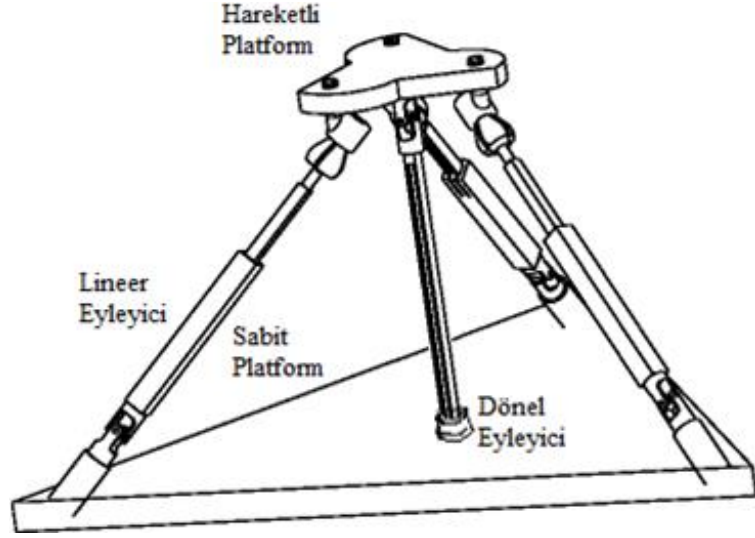
3 kollu 6 serbestlik dereceli bu mekanizma ilk olarak Beji ve Pascal (1999) tarafından dizayn edilmiştir [18].

Paralel robottaki prizmatik eklemler lineer aktivatör olarak kullanılmıştır. Robotun diğer universal ve küresel eklemleri pasif eklemlerdir. 3 adet lineer eyleyici eklem ile 6 adet pasif eklem kontrol edilmektedir. Bu durum robotun kontrolünde kararlılık problemi oluşturmaktadır. Pasif eklemlerin çoğunluğundan dolayı, robottaki aktif eklemlerin teorik olarak istenilen noktaya konumlandırılmasına rağmen pratik uygulamada pasif eklemlerdeki hareket olasılığının sonsuz sayıda olması robotun istenilen konuma ulaşmasını engellemektedir. Yani aktif eklemlerinin kinematik hesaplamaları sonucunda elde edilen varılacak noktaya yönlendirilmesinin ardından çok sayıda son pozisyon noktasının olmasından dolayı robot yörünge planlamasında kararsız davranmaktadır.



Şekil 5.1. 3-UPS paralel mekanizma [18]

Bu tez çalışmasında Beji ve Pascal modeli üzerinden türetilerek yeni bir model önerilmiştir. Bu model Beji ve Pascal modeline ek olarak alt ve üst platformun orta noktalarından bir dönele ve bir universal eklem ile bağlanması ile oluşturulmuştur. Alt platformdan üst platforma sırası ile dönele ve universal eklem Roll-Pitch-Yaw açı setinde dönmeye olanak sağlayarak robotun kararlı hareket etmesini sağlamıştır. Bu eklenen orta kolda dönele eklem aktif, universal eklem ise pasif eklemdir. Bu şekilde bir yapının oluşturulmasının sebebi 3 adet aktif prizmatik eklem ile verilen ulaşılacak olası konum sayısı 2 adete düşürülmüştür. Yani robotun orta noktası olan P noktasının sabit kalması şartı ile Roll-Pitch-Yaw açı setinde dönmesi sağlanmış ve tüm olası varılacak son nokta sayısı 2 adete düşürülmüştür. Bu olası 2 hareket ise Yaw açı seti çevresinde pozitif veya negatif yönünde olmak suretiyledir. Bu olası iki hareket aktif dönele eklem ile yörünge planlaması yapılarak robotun istenilen konuma gitmesi sağlanır.

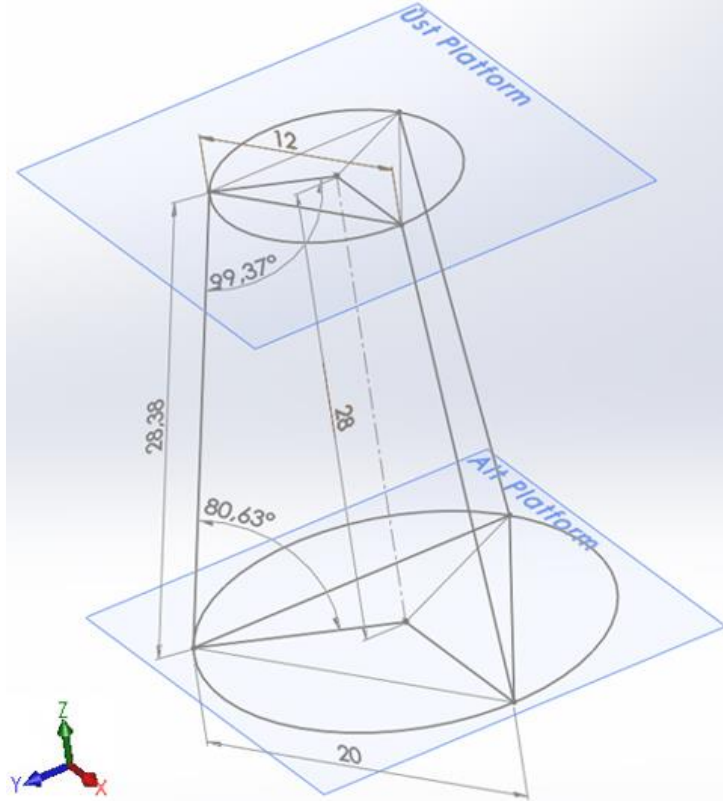


Şekil 5.2. 3-UPS, 1RU paralel mekanizma

Beji ve Pascal modeline karşı önerilen 3-UPS, 1RU paralel robot Şekil 5.2’de belirtilmiştir. Bu orta kolun gövdeye eklenmesi ile robotun taşıma kapasitesi, kinematik tutarlılığı ve yörünge kararlılığı arttırılmıştır. Bu sayede robot istenilen konuma planlandığı yörünge içerisinde kalarak gitmesi sağlanmış pasif eklemlerden kaynaklı olası hatalı yörüngelere gitmesinin önüne geçilmiştir. Buna karşılık orta kol robota yeni iki eklem eklenmesi açısından fazladan maliyet anlamına gelmektedir.

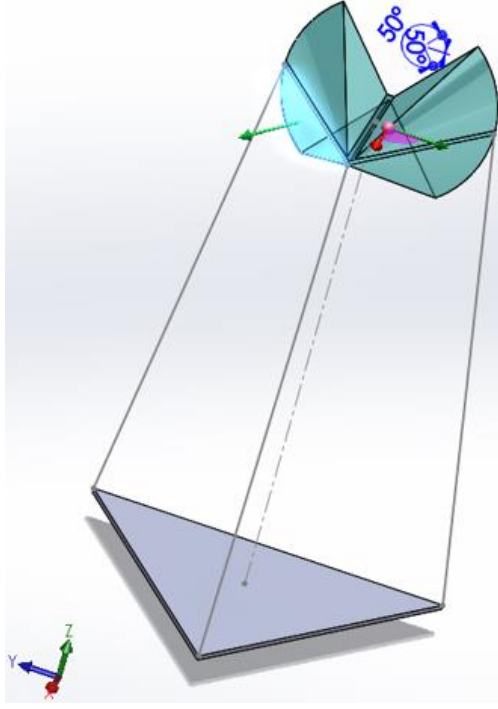
Bölüm 2’de anlatılan kinematik hesaplamalarının tamamı robotun hareketli platformunun orta noktası olan P noktasının sabit kalması esasına dayanmasından dolayı önerilen robot tipi için tamamen geçerlidir. Tez çalışması kapsamında yapılan uygulamada Bölüm 2 ve 3’te anlatılan sırası ile kinematik ve yörünge planlaması hesaplamaları uygulanmış ve doğruluğu teyit edilmiştir.

Şekil 5.3’te 3-UPS, 1-RU robotun ölçüleri ve çizimi gösterilmiştir. Robotun alt platformu ile yüksekliği arasındaki açı 80,63 derece seçilmiştir. Robotun üstüne ağırlık yüklendiği düşünüldüğünde robot sıfır pozisyonunda iken motorlarının gücünün $\cos(80,63)$ oranında yani 0,986’lık oranını dikey olarak üstteki platforma aktarmaktadır. Bununla birlikte üst platformun yönelime başlamadan önce lineer aktivatörlerinin $\sin(80,63)$ oranında yani 0,163’lük oranını yan bileşenlerine uygulayabilmektedir. Bu kuvvet üst platformun yöneliminin artması ile birlikte açığı yan kolların açılarına göre değişmektedir.

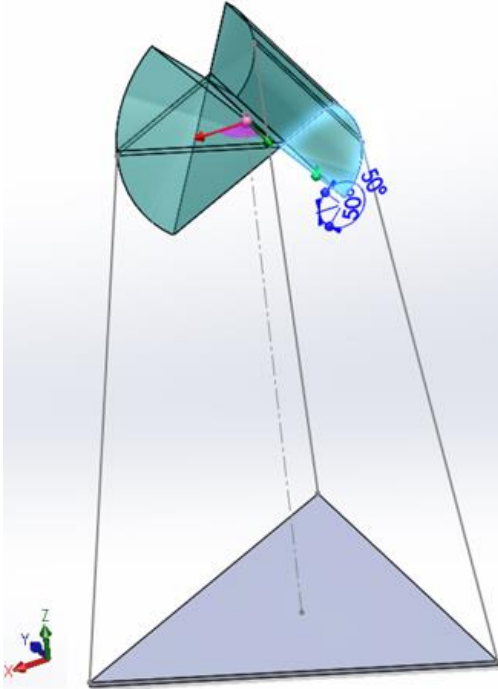


Şekil 5.3. 3-UPS, 1-RU robotun ölçüleri

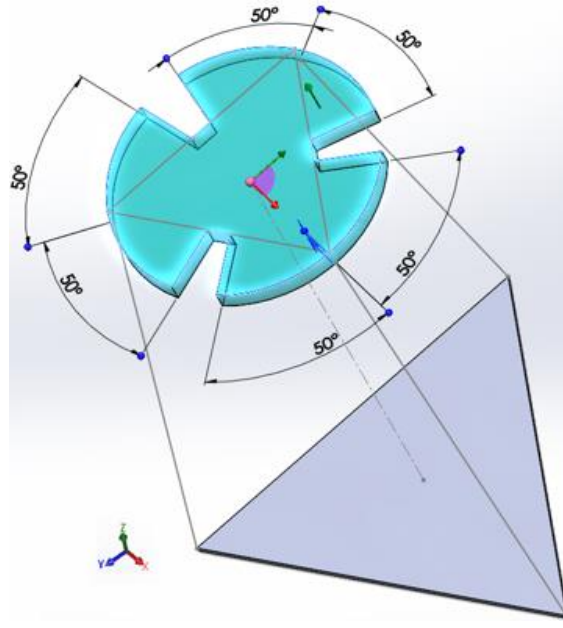
3-RPS, 1-RU robotun çalışma uzayı her ekseninde ellişer derece olacak şekilde tasarlanmıştır. Bu sayede artı ve eksi yönde 50 derece olacak şekilde robotun x,y,z eksenlerinde toplamda 100 derecelik çalışma uzayı sağlanmıştır. Şekil 5.4, 5.5 ve 5.6'da sırasıyla x, y, ve z eksenlerindeki robotun çalışma uzayı gösterilmektedir. Robotun her eksenindeki toplamda 100 derecelik dönme hareketi 3 eksenin bileşke hareketi incelendiğinde küresel bir hareket oluşturmaktadır. 3 eksenindeki bileşke hareketlerin tamamı robotun çalışma uzayını oluşturmaktadır. Robotun çalışma toplam çalışma uzayının şekli x, y ve z eksenlerinden görülecek şekilde sırası ile Şekil 5.7, 5.8 ve 5.9'da gösterilmiştir.



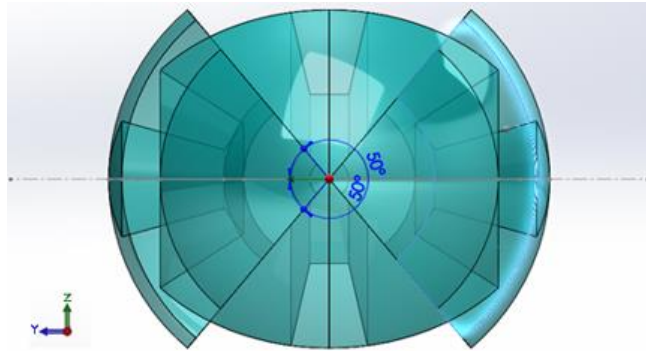
Şekil 5.4. X eksenindeki çalışma uzayı



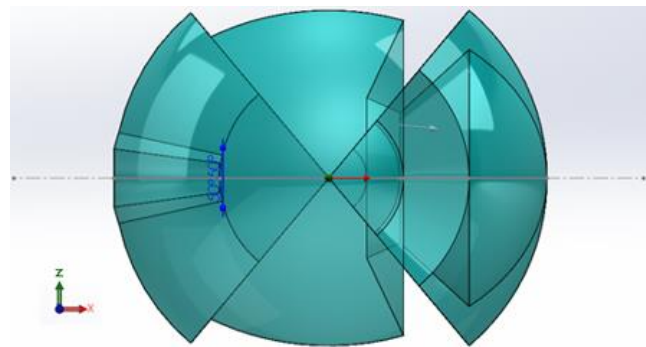
Şekil 5.5. Y eksenindeki çalışma uzayı



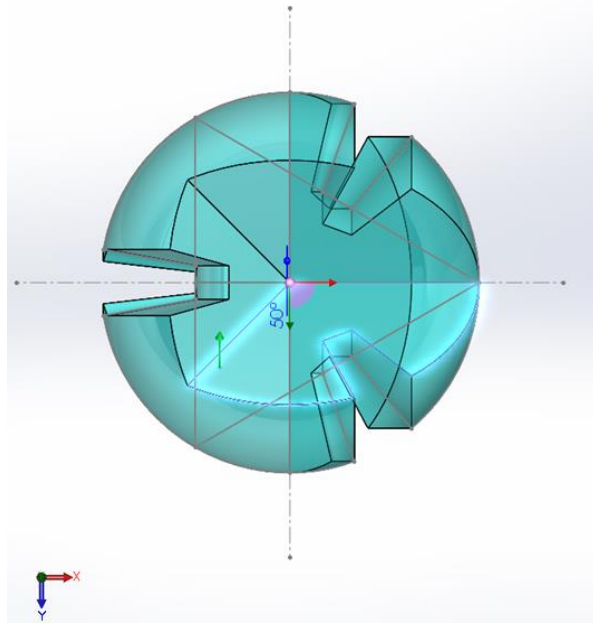
Şekil 5.6. Z eksenindeki çalışma uzayı



Şekil 5.7. X ekseninden toplam çalışma uzayı



Şekil 5.8. Y ekseninden toplam çalışma uzayı



Şekil 5.9. Z ekseninden toplam çalışma uzayı

Robotun önerilen modelinin tasarımı Lego Technic parçaları ile gerçekleştirilmiştir. Robot alt taraftaki sabit platformdan üst taraftaki hareketli platforma doğru sırası ile universal eklem, prizmatik eklem, küresel eklem olmak üzere 6 serbestlik dereceli eklemler zinciri ile bağlanmıştır. Robotun sabit ve hareketli platformlarını birleştiren orta kol ise sırası ile dönele ve universal eklem ile 3 serbestlik dereceli şekilde bağlanmıştır. Robotun yapımında kullanılan universal eklemler ve kübik eklemler Şekil 5.10'da gösterilmiştir.



Şekil 5.10. Universal ve kübik eklem [32, 33]

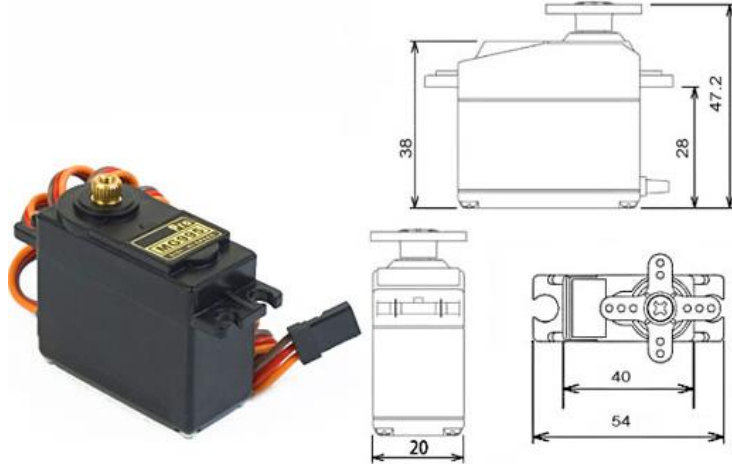
3-UPS,1-RU robotun eyleyicileri olarak Fircelli firmasının ürettiği lineer servo motorlar kullanılmıştır. Bu servo motorlar uygulandığı gerilimin kutbuna göre hareket etmektedir. Çalışmada bu motorların tercih edilmesinin sebebi, yeterli tork üretmesi, geniş çalışma gerilim aralığı ve robotun pozisyonu hakkında bilgi vermesini sağlayan

analog pozisyon geri beslemesidir. Analog geri besleme özelliği, uygulanan gerilim değerini, servonun içerisindeki bir potansiyometre mahareti ile servonun pozisyonu ile doğru orantılı olarak gerilim çıkışı vermesi ile sağlanmaktadır. Çalışmada kullanılan lineer servo Şekil 5.11’de gösterilmiştir. Ayrıca yine Firgelli firmasının lineer servoları için ürettiği kontrol kartları bulunmaktadır. Bu kontrol kartı ile servo farklı yöntemler ile çalıştırılabilmektedir. Bu çalışmada servonun kontrol kartı kullanılmamıştır. Servonun kontrolü L298 motor sürücü entegresi aracılığı ile doğrudan dsPIC33FJ32GP202 mikro denetleyicisi üzerinden sürülmüştür. Robotun sürülmesi için tüm aktif eklemlerin yörünge planlaması ile uyumlu şekilde hareketini sağlayacak bağımlı PWM algoritması geliştirilmiştir. Bu algoritma ile tüm aktif eklemler PWM sinyali ile sürülmekte ve sürekli olarak pozisyonları kontrol edilmektedir. Hızı aşan yörünge planlamasının önünde giden eyleyicileri yavaşlatıp, yörünge gerisinde kalan eyleyicilerin hızlandırılmasını sağlamaktadır. İlgili algoritmanın kodları Ekler bölümünde “setPosition” fonksiyonu içerisinde verilmiştir.



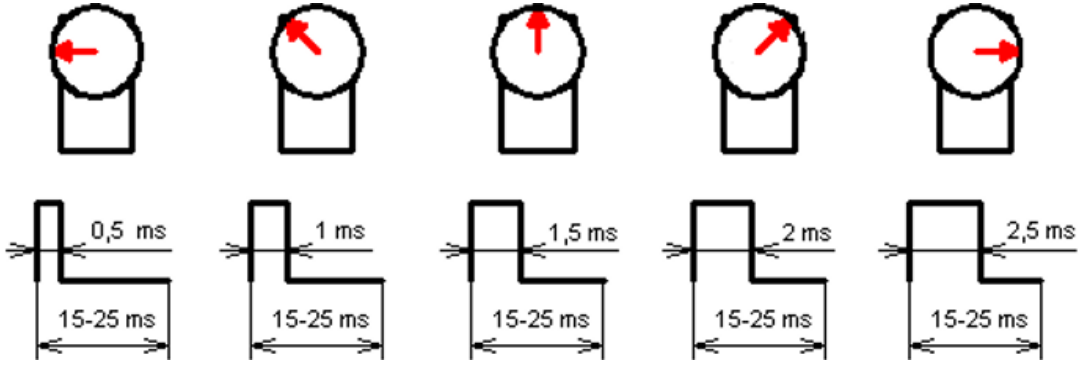
Şekil 5.11. Firgelli lineer servo motor [34]

3-UPS,1-RU robotun kararlı hareketi ve Z eksenini etrafındaki yönelimini sağlayan orta kol, aktif dönel eklem ve pasif universal eklemden oluşmaktadır. Bu kolun aktif dönel hareketi Şekil 5.12’de gösterilen MG 995 servo motor ile sağlanmıştır. Robotun Z eksenindeki dönüşünü sağlaması ve güç kapasitesini arttırması için metal dişli olan bu servo motor kullanılmıştır. Bu servo motor 4.8V’ta 13kg/cm tork üretme kapasitesine sahiptir.



Şekil 5.12. MG 995 servo motor [35]

Robotun çalışma uzayı olarak Roll-Pitch-Yaw açı setlerinin tamamında 50 derece olarak belirlenmiştir. Z eksenindeki hareketi sağlayan MG 995 dönel servo motor klasik PWM sinyali ile çalıştırılmıştır. 50 Hz’te çalışan servo motorlar uygulanan PWM sinyalinin görev çevrimine göre Şekil 5.13’te gösterildiği üzere pozisyon almaktadırlar. Tipik bir servonun 180 derece çalışma aralığının olduğu düşünüldüğünde robotun artı ve eksi yönde 50 derecelik ihtiyacını karşılamaktadır.



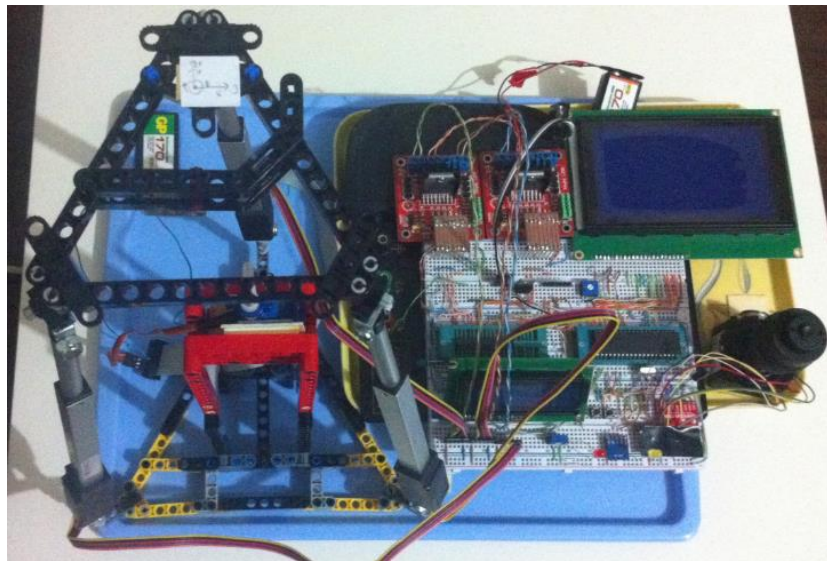
Şekil 5.13. Servo motor çalışması için PWM sinyali [36]

Robotun eklemlerinin dışında kalan ve robotun gövdesini oluşturan platformlar 120 derecelik açı aralıkları ile oluşturulan eşkenar üçgen şeklindedir. Yukarıda anlatılan eklemler ile robotun mekanik yapısı Şekil 5.14’te görüldüğü şekilde oluşturulmuştur.



Şekil 5.14. 3-UPS,1-RU robotun mekanik tasarımı

Robotun mekanik tasarımının tamamlanmasının ardından board ortamında Bölüm 4’te anlatılan elektronik komponentleri ile mekanik aksamlarının hareketi çalışılmıştır. Bölüm 2 ve 3’te anlatılan kinematik hesaplamaları ve yörünge planlamasına birebir uyulmuş ve robotun tüm eklemleri uyum içerisinde montajı patlatmadan hareketi sağlanmıştır. Şekil 5.15’te robotun mekaniği ve board ortamında elektronik aksamları ile birlikte çalıştırılması gösterilmiştir.



Şekil 5.15. 3-UPS,1-RU robotun elektronik komponentleri ile birlikte çalıştırılması

Robotun elektronik komponentlerinin board ortamında çalıştırılmasının ardından, bölüm 4'te anlatılan baskı devresi ile birlikte çalışması gerçekleştirilmiştir. Robotun tüm elektronik ve mekanik komponentleri bir sunta üzerine monte edilmiş böylece tüm donanımın bir arada olması sağlanmıştır. Şekil 5.16'da 3-UPS,1-RU robotun tüm donanımları ile birlikte çalıştırılması gösterilmiştir.



Şekil 5.16. 3-UPS,1-RU robotun tüm donanımları ile birlikte çalıştırılması

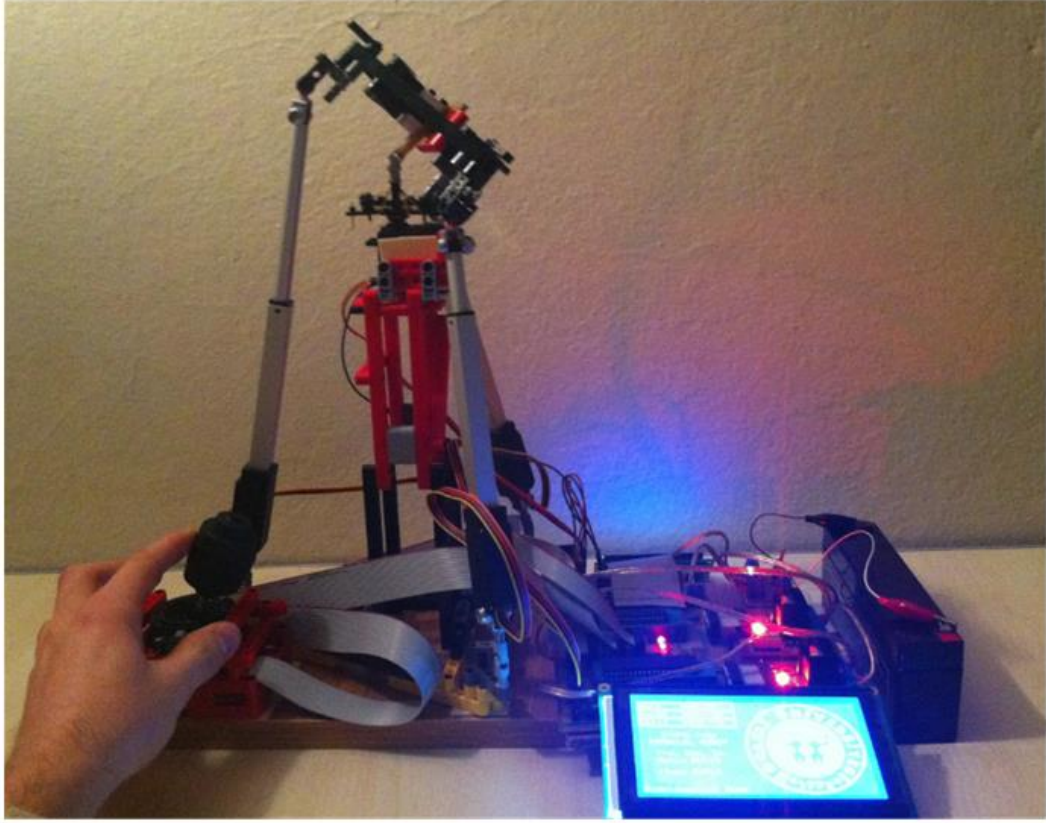
5.2. 3-UPS, 1-RU Robotun Çalışma Modları

Robotun tasarımında kullanılan elektronik komponentleri ile birlikte birden fazla kullanım alanı düşünülmüştür. Yazılımsal olarak her bir kullanım alanı için farklı prosedürler yazılmıştır. Bunlar kısaca aşağıdaki şekilde sıralanabilir

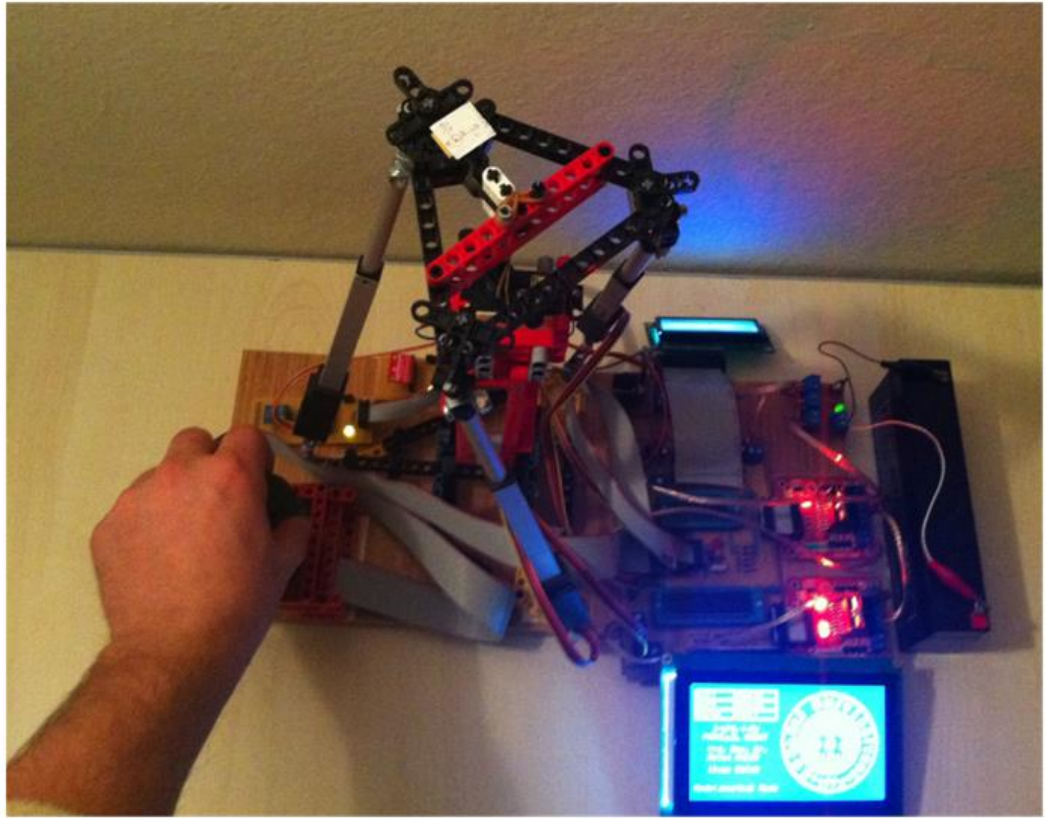
1. Idle Mode: Bu modda robot hareketsiz olarak beklemektedir.
2. Joystick Mode: Robotun kontrolü endüstriyel uygulamalarda kullanılan bir joystick ile sağlanmaktadır.
3. Zero Position Mode: Robot sıfır pozisyonuna yani Roll-Picth-Yaw açısı 0,0,0 olduğu pozisyona alınmaktadır.
4. Workspace Mode: Robotun çalışma alanını gösteren demo bir hareket programcıdır.

5. Sensor Mode: Robotun hareketi ADXL345 ve HMC5883L sensörleri ile sağlanmaktadır.
6. Parallel Mode: Robotun sabit platformu hangi açığa getirilirse getirilsin robotun hareketli üst platformu her zaman deniz seviyesine paralel durmaktadır.

Robotun en yaygın endüstriyel uygulamalarda kullanılacak modu joystick moddur. Bu seçenek ile robotun hareketli platformu 3 eksenli bir joystick yardımı ile kontrol edilmektedir. 3 eksenle her iki yöne 50 derece dönebilen joystick ile robota Roll-Pitch-Yaw açı setinde 50 derecelik hareket uzayı sağlanmaktadır. Joystick, üzerinde her eksen için ayrı ayrı bulunan potansiyometre vasıtasıyla açı bilgisini, kendisine uygulanan gerilim ile doğru orantılı bir şekilde çıkışa vermektedir. Bu bilgi kinematik denklemlerinde Roll-Pitch-Yaw açı seti parametresi olarak kullanılmakta ve robotu uygulanan açı değerlerine konumlandırmaktadır. Robot bu özelliği ile manuel takip edilecek endüstriyel uygulamalarda kullanılabilir. Örnek olarak Bölüm 6'da anlatılacak olan telekomünikasyon sektöründe antenlere istenen tilt ayarının verilmesinde kullanılabilir. Şekil 5.17 ve 5.18'de sırası ile robotun Y ve Z eksenlerindeki joystick vasıtasıyla yaptığı hareket görülmektedir.

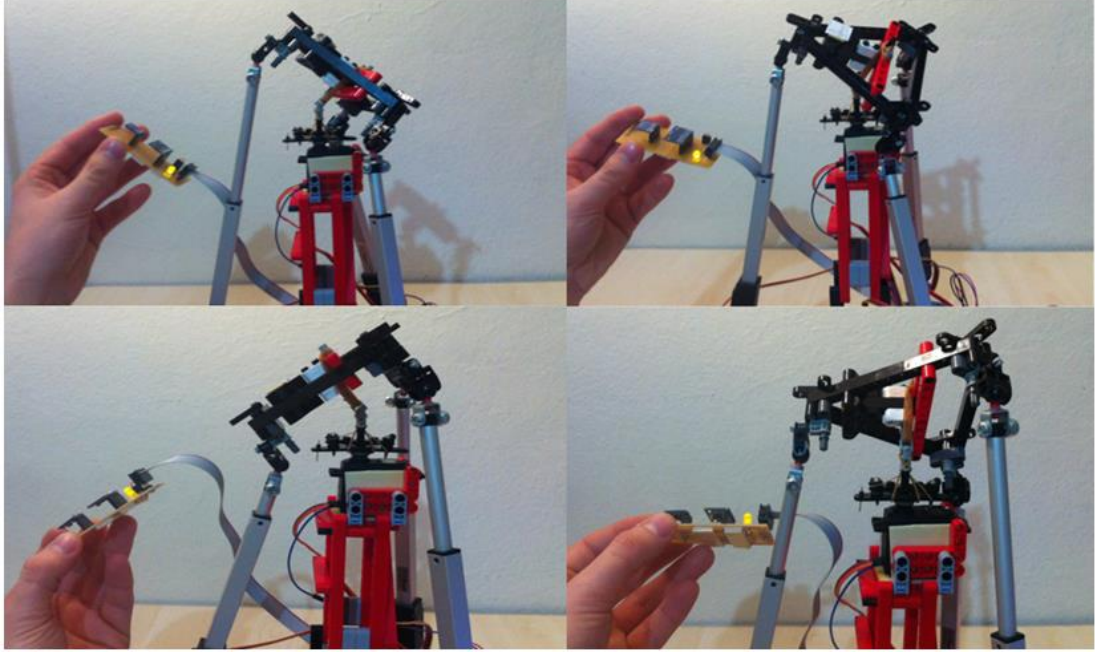


Şekil 5.17. 3-UPS, 1-RU robotun joystick vasıtasıyla Y eksenindeki hareketi



Şekil 5.18. 3-UPS, 1-RU robotun joystick vasıtasıyla Z eksenindeki hareketi

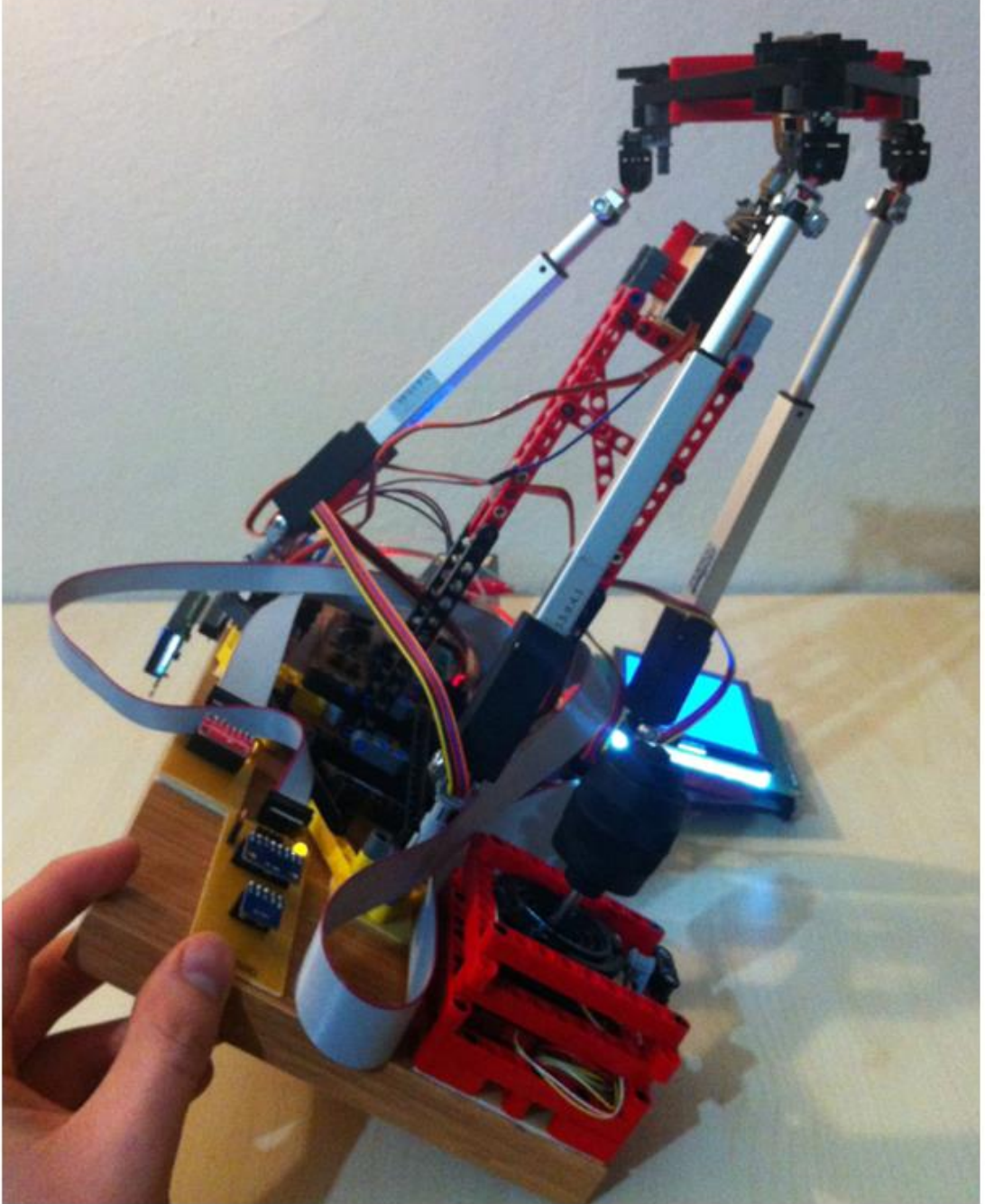
Robotun endüstriyel uygulamalarda kullanılabilir olan diğer bir özelliği ise ADXL345 ivme sensörü ve HMC5883L dijital pusulası ile yapılan kontrolüdür. Bu sensörler sayesinde algılanan X,Y,Z eksenindeki açı seti robotun kinematik denklemlerinin girdisini oluşturmaktadır. Bu sayede sensörler ile belirlenen eğimin, robotun gideceği Roll-Pitch-Yaw açı seti olarak belirlenir. Bu sensörler vasıtasıyla, sensörlerin takıldığı bir platforma, robotun hareketli platformunun sürekli paralel kalması sağlanabilir. Örnek olarak anahtar kopyalama makinelerinde tek eksenle yapılan paralel işlem düşünülebilir. Tek bir eksenle hareket kopyalanacak tarafta tekrarlanır. Paralel robotların 3 eksenle bu hareketi yaptığı düşünülürse 3 eksenle bir kopyalama işlemi gerçekleştirilebilir. Şekil 5.19'da robotun sensörler vasıtasıyla yaptığı hareket bulunmaktadır.



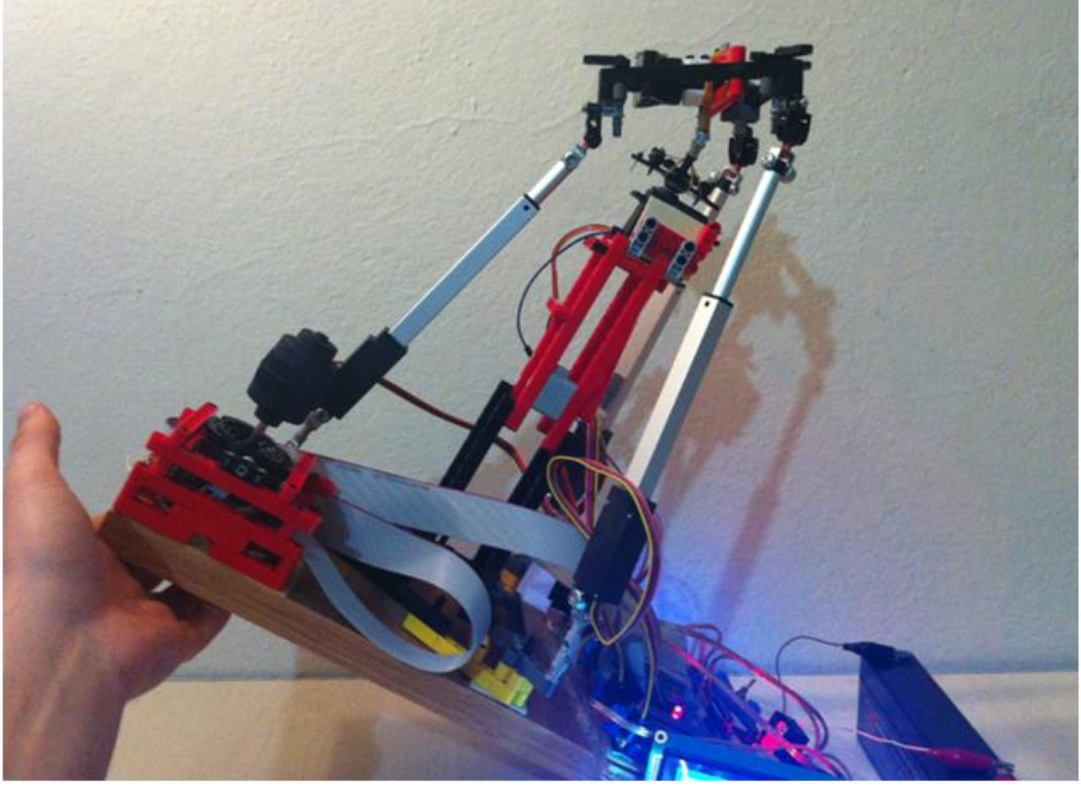
Şekil 5.19. 3-UPS,1-RU robotun sensörler vasıtasıyla X ve Y eksenindeki hareketi

Robotun diğer bir kullanım modu ise Paralel moddur. Bu modda X ve Y eksenindeki açılar ADXL345 sensörü ile algılanarak robotun kinematik denklemlerinde girdi olarak kullanılır. Bu modun amacı, robotun sabit alt platformunun hangi açıya gelirse gelsin, üst hareketli platformun daima deniz seviyesine paralel kalmasıdır. Bu mod ile robot, lojistik sektöründe sallanmadan veya devrilmeden taşınması gereken ekipmanlar için veya özel kimyasal sallanması durumunda tepkimeye girme olasılığı olan maddelerin taşınmasında kullanılabilir. Yine bu modda robot Bölüm 6'da bahsedilecek olan telekomünikasyon sektöründe deniz üzerinde kullanılacak olan baz

istasyonlarında kullanılabilir. Robotun paralel moddaki sırasıyla X,Y ve XY eksenlerindeki hareketi Şekil 5.20 ve 21’de gösterilmektedir. Benzer şekilde Z eksenindeki hareketinde istenen yönde sabit kalması HMC5883L gibi bir sensör yardımı ile sağlanabilir.



Şekil 5.20. 3-UPS,1-RU robotun sabit platformunun X ekseninde hareket ettirilmesi



Şekil 5.21. 3-UPS,1-RU robotun sabit platformunun Y ekseninde hareket ettirilmesi

6. PARALEL ROBOTLARDA YENİ KULLANIM ALANI ÖNERMELERİ

Bu tez çalışmasında tasarlanan 3-UPS,1-RU paralel robotun uygulaması olarak endsütriyel joystick ile uygulaması, kullanılan ADXL345 ve HMC5883L sensörü ile sensör uygulaması ve paralel mod uygulaması gerçekleştirilmiştir. Kullanım alanlarına örnekler verilmiştir. Bu bölümde literatürde olmayan telekomünikasyon sektöründe kullanım önerileri ve nasıl gerçekleştirileceği yönünde bilgi verilmiştir.

İlk olarak paralel robotların otomatik mekanik anten tilt aparatı olarak kullanılmasından bahsedilmiştir. İkinci bir kullanım alanı olarak hareketli baz istasyonlarında radyo link haberleşmesi probleminin giderilmesinde önerilmiştir.

6.1. Paralel Robotların Anten Tilt Aparatı Olarak Kullanılması

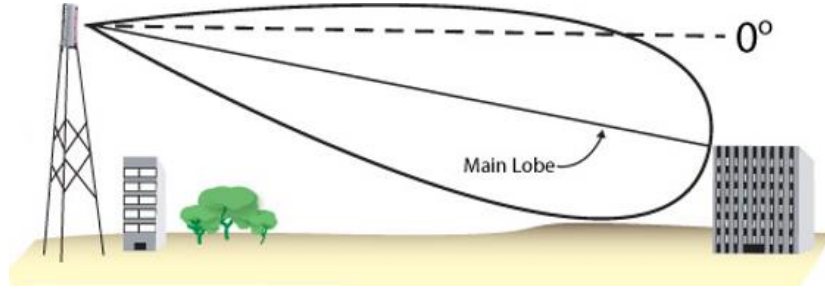
Telekomünikasyon sektöründe geline kablosuz haberleşmede yaşanan gelişmeler ve ulaşılan yüksek hızlar kablosuz haberleşmeyi en etkin haberleşme aracı haline getirmiştir. Şekil 6.1’de kablosuz haberleşme teknolojilerinin gelişimi gösterilmiştir. Gelişen teknoloji ile birlikte LTE teknolojisi ile ülkemizde, Turkcell operatörü 900Mbps hızlarına erişmiştir.



Şekil 6.1. Kablosuz haberleşme gelişimi [37]

Şekil 6.2’de görüldüğü gibi kablosuz haberleşme için baz istasyonları ile kapsama sağlanır. Kapsama altına alınan alandaki mobil istasyonlar ile haberleşme çift yönlü şekilde yapılır. Kapsama için belirli özelliklerdeki kazancı yüksek antenler kullanılır.

Omni antenler dışındaki antenlerin belirli bir yönelimde ışıma paterni bulunmaktadır. Bu ışıma paterninin en yoğun olduğu doğrultuya ana lob denir. Kapsanmak istenen alanın özelliklerine göre antenin ışıma paterni optimize edilerek ilgili alana yönlendirilir. Bu şekilde bir baz istasyonu hücre denilen yapılara bölünerek genelde 120 derecelik aralıklarla 3 yöne bakmak kaydıyla 3 sektöre bölünürler. Sektörel bölünüm ortam koşullarına göre farklı açılarda, farklı sayıda ve kapasitede olabilir.



Şekil 6.2. Baz istasyonu kapsamı [38]

Anten yönünün ayarlanması 2 farklı yöntem ile sağlanır. Bunlardan ilki Şekil 6.3'te görünen mekanik tilt yöntemidir. Bu yöntem ile antenin ışıma paterni x ekseninde belirli bir açıda döndürülerek hedeflenen noktaya yönlendirilir. Kolay bir yöntem olmasına rağmen kişiye bağlı bir yöntemdir. Herhangi bir değişim durumunda ilgili çalışanın antene ulaşması ve manuel operasyon yapması gerekmektedir. Antenin ulaşılması zor olduğu durumlarda, yüksek yerlerde, kulede olması durumunda anten eğim operasyonunu oldukça zorlaştırmaktadır. Tek ekseninde paternin ayarlanabilir olması diğer bir dezavantajdır.



Şekil 6.3. Mekanik tilt [39]

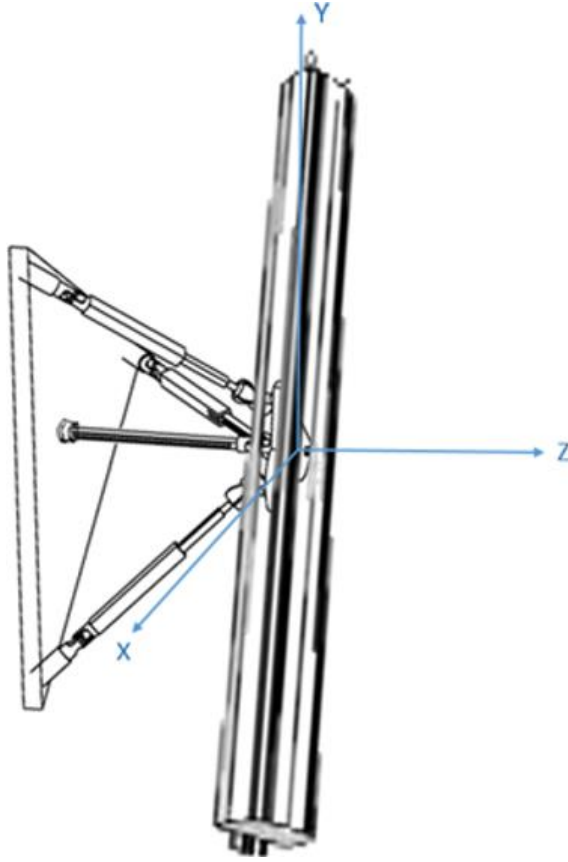
İkinci bir yöntem ise elektrik tilt yöntemidir. Bu yöntem sadece elektrik tilt uyumlu antenler ile çalışmaktadır. Şekil 6.4’te görüldüğü üzere uyumlu antenin altına bağlanan elektronik bir tilt aparatı ile mümkün kılınmaktadır. Mekanik tilt yöntemine benzer şekilde sadece x ekseninde tilt olanağı sağlar. Mekanik tilte avantajı ise herhangi bir yerinde operasyon gerektirmeden uzaktan tilt verilebilmesidir.



Şekil 6.4. Elektrik tilt [40]

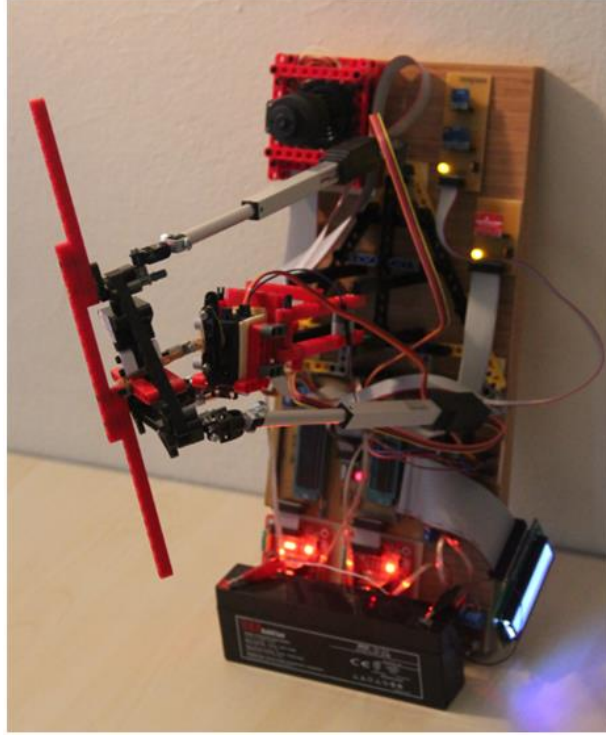
Günümüzde Ericsson firması tarafından geliştirilen akıllı self-optimize telekomünikasyon cihazlarında yoğun trafiğe doğru antenlere tilt verme gibi bir eksenenden fazla ekseninde hareket gerektiren çalışmalar mevcuttur. Bu tez çalışmasında gerçekleştirilen paralel robotun veya genel olarak anlatılan paralel robotların diğer bir kullanım alanı olarak kablosuz haberleşme sektöründe kullanılan anten tilt operasyonu önerilmiştir. Sebebi ise paralel robotlar ile sağlam bir yapı kurulması ve hassas hareket olanaklarının olması tilt operasyonlarında paralel robotların kullanımını mümkün kılmaktadır. Özellikle gelecek teknolojiler olan self-optimize şebekelerde tilt operasyonlarının sayısı ve hassasiyeti bir hayli önem kazanacaktır. Bu yüzden hem kişi bazlı hem de otonom tilt operasyonlarında birden fazla ekseninde tilt ihtiyacını karşılayabilecektir. Önerilen paralel mekanizmalı yapı Şekil 6.5’te gösterilmiştir. Şekil 6.5’te temsili olarak bu tez çalışmasındaki yapı kullanılmasına rağmen genel olarak paralel robotlardaki sistemler ile bu şekilde bir yapı kurulması mümkündür. Mevcuttaki tilt aparatlarına göre 3 ekseninde hareket imkanı tanınması akıllı şebekelerde bu yapının kullanılmasına ve şebekede 3 boyutlu optimizasyon imkanı tanımaktadır.

Bunun yanında sisteme ilişkilendirilecek bir haberleşme modülü ile tamamen uzaktan çalıştırılabilir geri beslemeli bir sistem yapılması mümkündür.

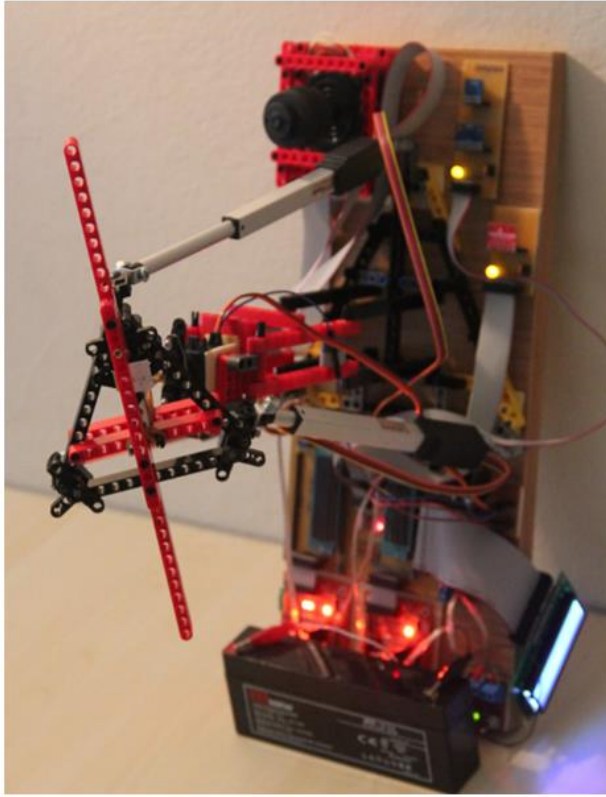


Şekil 6.5. Paralel mekanizmalı anten tilt aparatı

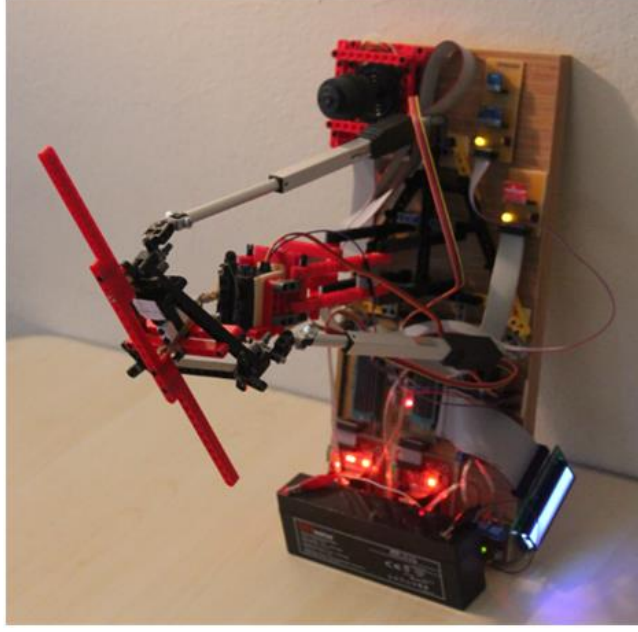
Sırasıya Şekil 6.6, 6.7 ve 6.8’de anten yön değişimi için 3-UPS, 1-RU robotun kullanımı gösterilmiştir. Hareketli platforma bağlı olan kırmızı Lego anteni simgelemektedir. Bu sayede anten, uzaktan yüksek hassasiyetle, X,Y,Z eksenleri etrafında kararlı bir yön değişimi hareketi yaptırılabilir.



Şekil 6.6. 3-UPS,1RU robotun anten yön deęişimi için kullanılması



Şekil 6.7. 3-UPS,1RU robotun anten yön deęişimi için kullanılması



Şekil 6.8. 3-UPS,1RU robotun anten yön deęişimi için kullanılması

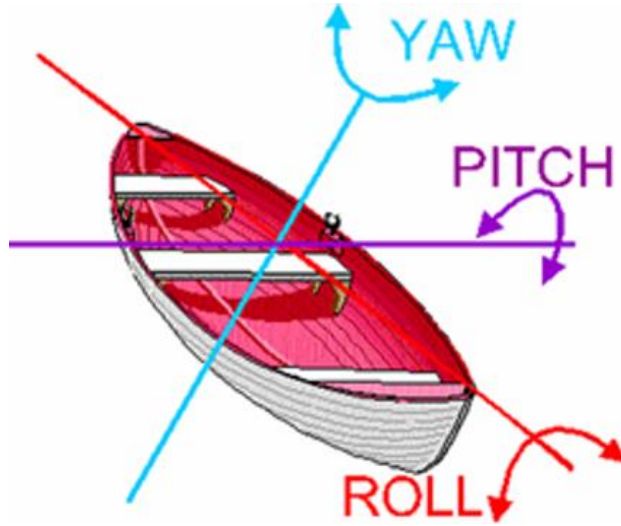
6.2. Paralel Robotların Hareketli Baz İstasyonlarında Kullanılması

Kablosuz iletişimde baz istasyonu ile mobil istasyon arasındaki veri transferinden sonra, baz istasyonundan ilgili santrallere doęru bir haberleşme trafięi mevcuttur. Bu veri transferi HDSL, fiber veya radyolink yolları ile saęlanır. Fiberle iletişim çok yüksek veri hızı saęlamasına raęmen büyük altyapı yatırımları gerektirir. HDSL devreleri kiralık devreler olduęu için yüksek kira bedelleri mevcuttur. Operatörler için mümkün olduęunca saha özelliklerine baęlı olarak sahaların haberleşmesinin büyük kısmı radyolinkler ile saęlanır. Radyolinkler davul anten kullanan ışımaya paterni dar fakat yüksek mesafeli haberleşme sistemleridir. Bu sistemler sayesinde kullanıcılardan baz istasyonuna ulaşan veri, radyolinkler ile ilgili toplama merkezlerinden santrallere taşınırlar. Radyolink sistemlerde en büyük problem doğrudan görüş LOS gerektirmesidir. Mevcut görüşün kesilmesi halinde haberleşme duracaktır.

Kablosuz haberleşmede dięer bir problem göl ve denizlerin kapsanmasıdır. Denizlerdeki su tabakası gönderilen sinyalleri adeta bir yansıtıcı gibi yansıttığı için uzun mesafeli kapsamalarda yansıyan sinyaller girişim oluştururlar ve çok uzun mesafe giderler. Bu yüzden denizlerde düşük çıkış gücünde sahalar ile kapsanmalıdır. Bu durum deniz içerisinden kapsama ihtiyacı doğurur.

Mevcut telekomünikasyon sektöründe deniz kapsamaları için çeşitli çalışmalar mevcuttur. Bunlardan biri Turkcell operatörünün gerçekleştirdiği batık saha uygulamasıdır. Batık saha uygulaması mevcut bir teknenin belirli bir derece batırılarak içerisine bir baz istasyonu eklenmesi ile gerçekleştirilir. Kapsama sağlayan bu baz istasyonu sinyali karaya radyolink üzerinden göndermektedir. Fakat yukarıda bahsedilen radyolinklerin doğrudan görüş açısı gerektirmesi nedeni ile denizlerde dalga sarsıntılarından dolayı çalışma verimli olmamıştır. Sarsılan tekneden gönderilen radyolink sinyalleri karadaki karşılığına ulaşamadığı için haberleşme kesintiye uğramaktadır.

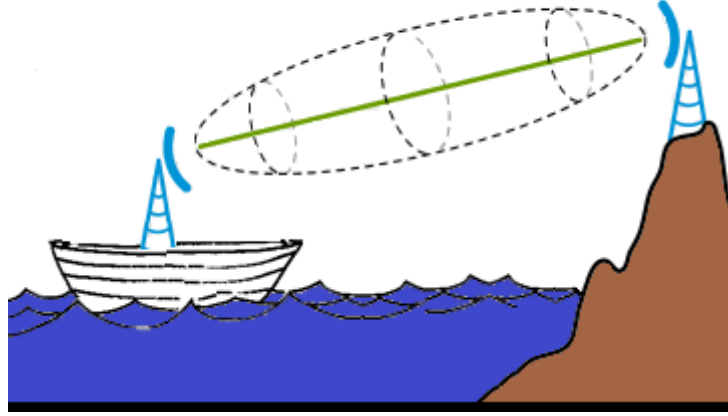
Suda savrulan bir bot tipik Roll-Picth-Yaw açı setinde hareket yapmaktadır. Bu hareketlilik Roll-Pitch-Yaw açı setinde Şekil 6.9’da gösterildiği gibi üç eksen etrafında dönmek suretiyle gerçekleşir. Tez çalışması kapsamında çalışılan 3-UPS, 1-RU robotun hareketli platformunun hareket eksenleri ile aynı mantıktadır. Bota ilâştirilecek sensörler vasıtasıyla algılanacak olan açı değişimi, bot üzerine kurulacak olan bir paralel mekanizma ile sıfırlanabilir. Dalgaların hareketinden kaynaklı doğal hareketin hızı ile eşit hızda bir paralel mekanizma ile dalgaların oluşturduğu etki giderilebilir.



Şekil 6.9. Sudaki botun savrulma eksenleri [41]

Şekil 6.10’da suda bulunan bot üzerindeki radyolink ile karada bulunan radyolinkin haberleşmesi gösterilmiştir. Buradaki problem sudaki düzensiz hareketten kaynaklı radyolinkin bakış açısının sürekli değişmesidir. Sudaki düzensiz hareketlerden dolayı sudaki linkin bakış açısı sürekli değişmektedir. Karada sabitlenen radyolinklerde böyle

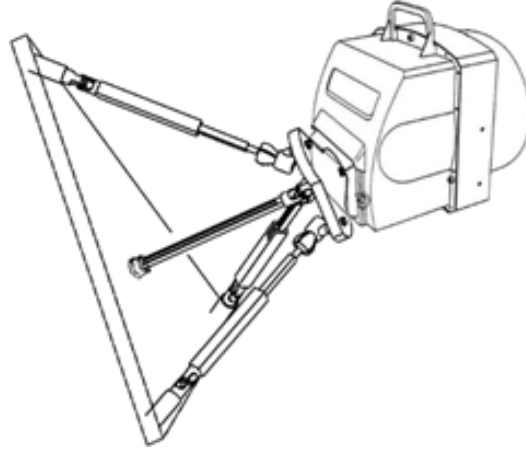
bir problem bulunmamaktadır. Bu düzensiz dalga hareketleri Şekil 6.9’da gösterilen Roll-Pitch-Yaw açı setinde yapılan düzensiz hareketlerdir. Eğer bu hareketlerin üç eksendeki bileşkesinin tam zıttı yönünde radyolink gerçek zamanlı yönlendirilebilirse radyolinkin bakış açısının daima sabit kalması sağlanabilir.



Şekil 6.10. Su ve kara arası radyolink haberleşmesi

Yukarda bahsi geçen problem için çözüm önerisi olarak Roll-Pitch-Yaw açı setinde hareket gereksinimini karşılayacak bir paralel robot önerilebilir. Sistemin tepki süresi dalgaların düzensiz hareketine gerçek zamanlı cevap verebilecek düzeyde kurulacak bir paralel robot sistemi ile radyolinkin bakış açısının daima sabit kalması sağlanabilir.

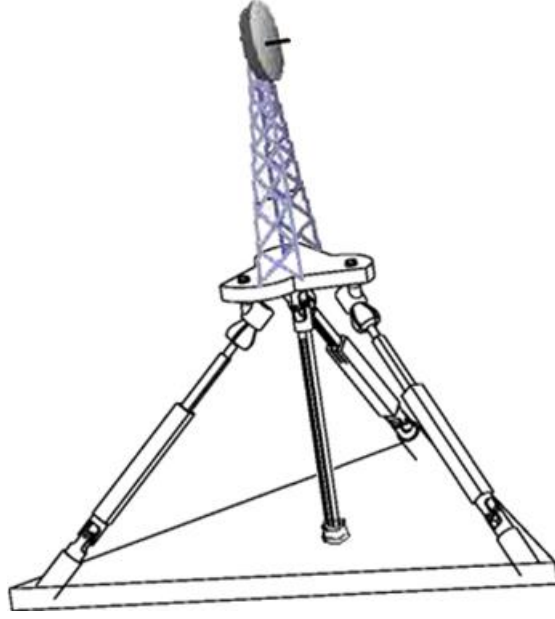
Sudaki düzensiz hareketlerin ivme sensörü ve dijital pusula gibi sensörler ile hassasiyetle algılanması yolu ile problemin çözümü mümkün kılınır. Çevresel değişkenlerin algılanmasının ardından paralel robot ile problemin çözümü iki farklı şekilde sağlanabilir. Bunlardan ilki Bölüm 6.1’de anlatıldığı şekilde radyo anteninin paralel bir mekanizma üzerine sabitlenerek anten yönünün sürekli radyo karşılığı olarak belirlenen antene bakması sağlanabilir. Şekil 6.11’de bu yapı gösterilmiştir. Böylece hangi yönde belirsiz hareket gelirse gelsin sensörler ile algılanan açı setindeki değişiklikler paralel mekanizma ile giderilerek antenlerin sürekli istenilen yöneline bakması sağlanabilir.



Şekil 6.11. Paralel mekanizmaların radyo link antenleri ile kullanılması

Diğer bir çözüm ise paralel platformun Şekil 6.12’de gösterildiği gibi doğrudan antene uygulanmasından ziyade antenin bağlı bulunduğu pole uygulanması ile gerçekleştirilir. Sensörler vasıtası ile gerçek zamanlı olarak dalgaların teknede oluşturduğu Roll-Pitch-Yaw açısı setindeki dönme hesaplanarak üst platformu sürekli olarak Roll ve Pitch açısı setinde dönmesi ile deniz seviyesine paralel olarak tutulur. Dünyanın doğal manyetik alanından faydalanarak çalışan bir dijital pusula ile dalgaların Yaw açısı setinde meydana gelen dönme algılanarak z eksenindeki dönme tölere edilerek radyolink istenilen doğrultuya yönelimi konumlandırılabilir.

Bahsedilen 2 farklı çözüm yöntemi de tez kapsamında anlatılan kinematik hesaplamaları, yörünge planlaması ve uygulaması ile çözülebilir. Çözümler tez çalışmasında gerçekleştirilen 3-UPS,1-RU robotun kinematik ve yörünge planlaması çözümlerine göre anlatılmasına rağmen genel olarak paralel robotların kullanılması vasıtasıyla çözümlenebilir. Bu tez çalışması kapsamında gerçekleştirilen 3-UPS,1-RU robotun hareketli zeminde sabit platform olarak kullanılması Şekil 6.13’te gösterilmiştir. Robotun sabit platformu hangi açıda tutulursa tutulsun, üst hareketli platform sensörler vasıtasıyla algılanan değişimi sıfırlar. Bu sayede üst platform her zaman istenilen eğimde ve yönde kalır.

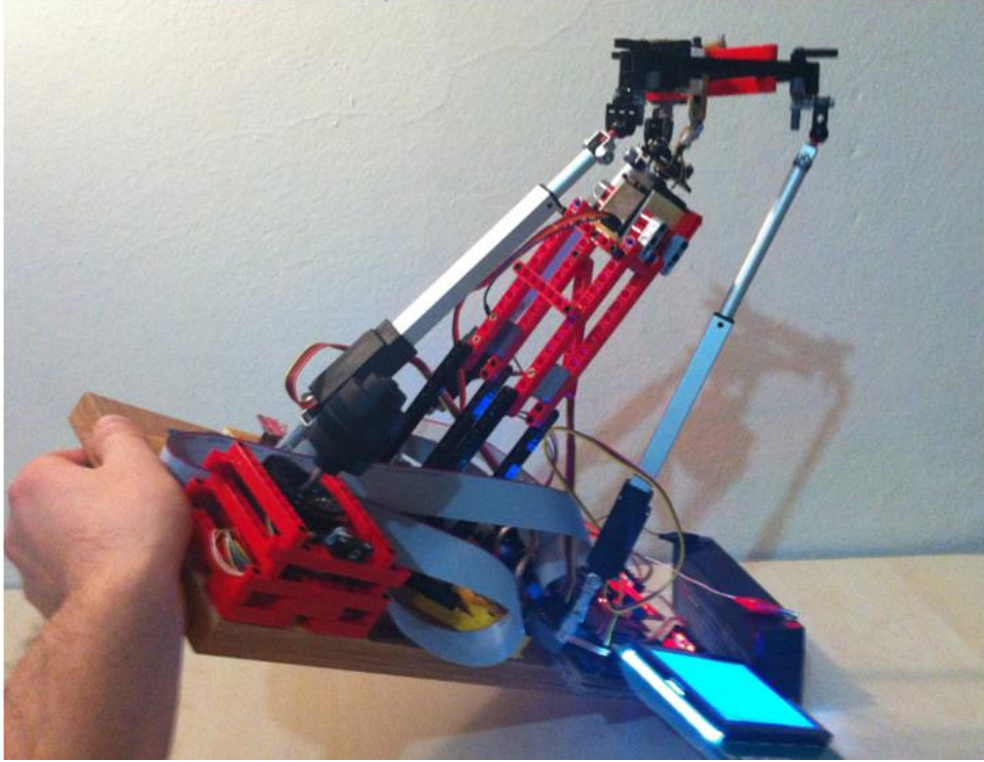


Şekil 6.12. Radyo link sistemlerinin paralel mekanizmalar ile sabitlenmesi

Çözümlerin temeli, tez çalışmasında uygulanan 3-UPS,1-RU robotun çözümünde kullanılan ivme sensörü ve dijital pusula ile doğadaki düzensiz hareketlerin gerçek zamanlı algılanarak, çözümlenmesi ve robotun gerçek zamanlı tepki vermesine dayanmaktadır. İvme sensörü ile algılanan Roll ve Pitch açı setlerindeki değişim ile robotun hareketli platformu deniz seviyesine paralel kalması sağlanmaktadır. Yaw açı setinde meydana gelen dönme ise dijital bir pusula vasıtası ile algılanarak, ivme sensöründen gelen bilgi ile birleştirilerek kinematik denklemlerinin girdisini oluşturur. Roll-Pitch-Yaw açı setindeki değişikliklere göre çözümlenen kinematik denklemine göre robotun yapısına göre uygulanacak olan yörünge planlaması çözümlenmesi gerektirmektedir. Yörünge planlamasının çözümlenmesinin ardından robot ulaşılabilecek konumuna doğru harekete geçmelidir. Bu işlemlerin gerçek zamanlı olması büyük önem taşımaktadır. Dalgaların hızından doğan robotun hareket hızı gereksinimi karşılanmalıdır. Aksi takdirde radyolinkin bakış açısı değişebileceğinden dolayı haberleşmede kesintiler meydana gelecektir. Yapılacak çalışmada dalgalardan kaynaklı maksimum hız gereksinimi hesaplanmalı ve buna göre malzeme seçimi yapılmalıdır.

Robottaki kinematik, yörünge planı çözümlenmelerinin yapılması ve aynı anda hareket halindeyken sürekli bir geri besleme olmak zorundadır. Bunun nedeni de dalgaların

düzensiz bir hareket oluşturmaktadır. Düzensiz hareket değişimi sürekli sensörler tarafından dinlenmeli ve belirlenen eşik değerindeki hassasiyet aşıldığında robotun kinematik, yörünge planlaması ve hareket algoritmaları tekrar hesaplanmalıdır. Bu tez kapsamında yapılan 3-UPS,1-RU robotta sürekli olarak kinematik hesaplamalarında, yörünge planlamasında ve hareket basamaklarında 10 ms'de bir sensörler dinlenmektedir. 10 ms içerisinde gerçekleşen bir hareket eşik değerini aşmışsa bahsedilen hareket süreci kinematik hesaplamalarından itibaren tekrarlanır. Benzer şekilde dalgalardaki hareket hızı ve tipi incelenerek geri besleme periyodu uygun olarak belirlenmelidir.



Şekil 6.13. 3-UPS, 1-RU robotun hareketli ortamda sabit platform olarak kullanılması

7. SONUÇLAR ve ÖNERİLER

Bu tezde çalışmasında paralel robot kinematiği teorik çözümleri incelenmiştir. Kinematik çözümü ile birlikte gelen ulaşılacak pozisyon konumuna yörünge planlaması yapılmıştır. Teorik çözümlerin tutarlılıkları yapılan paralel robot uygulaması ile denenmiştir. Deneysel uygulamada ilk olarak Beji ve Pascal modeli paralel robot denenmiştir. Robotun 3 aktif prizmatik eklem ile 3 adet pasif universal eklemi ve 3 adet pasif küresel eklemi kontrol etmekte yetersiz kaldığı saptanmıştır. Pratikte yeterli seviyede kararlı bulunmayan bu yapıya yeni bir yapı önerilmiştir. Bu yapı ile çalışmada teorik olarak anlatılan kinematik çözümleri ve yörünge planlamalarının deneysel uygulamada çözümleri gerçekleştirilmiş ve kararlı bir paralel robot hareketi sağlanmıştır. Robot hareketi için bağımlı yörünge algoritması geliştirilmiş ve robotun bütün kollarının yörünge planı ve birbirleri ile uyumu sağlanmıştır.

Paralel robotların geniş uygulama alanlarından bahsedilmiş ve deneysel uygulamada örnek bir uygulama yapılmıştır. Paralel robot taşıyacak ve mobilitesini sağlayacak bir mobil platform tasarlanmıştır. Tasarlanan mobil platform ile paralel robotun üst platformunun doğal engellerden bağımsız bir şekilde deniz seviyesine paralel kalması sağlanmış, teorik bilgilerde bahsedilen kinematik ve yörünge planlarının kararlı bir şekilde uygulanması sağlanmıştır. Bu şekilde rampa olarak kullanılabilen, kimyasal taşımacılıkta taşınan maddelerin hareket etmemesi için kullanılabilen bir mobil paralel robot simülasyonu yapılmıştır.

Literatürdeki kullanımlarına ek olarak tasarlanan paralel robotun yeni kullanım alanları önerilmiştir. Önerilen kullanım alanlarından ilki GSM, UMTS ve LTE sektöründe anten ışınma doğrultusu optimizasyonunda, antenlerin mekaniksel olarak tilt verilmesi çözümüdür. Bu şekilde haberleşme sektöründe büyük önem taşıyan tilt optimizasyonunda otomatik self-optimize sistemler tasarlanabilir. Diğer önerilen kullanım alanı ise batık saha olarak adlandırılan deniz üzerinde bulunan haberleşme baz istasyonlarının denizde dalga hareketlerinden dolayı radyolarının haberleşme

problemine çözüm önerisi olarak geliştirilmiştir. Doğal deniz dalgalarının x,y,z (Roll-Pitch-Yaw) açı setinde oluşturacağı hareket problemi, sensörler vasıtası ile algılanarak bot veya tekne üzerindeki baz istasyonunun radyosunun karadaki karşılığı ile doğrudan görüş açısı alması sağlanılarak çözümlenebilir. Bu çözüm radyonun bağlı olduğu direğe uygulanabileceği gibi tilt verme mantığı ile radyo antenine doğrudan uygulanabilir.

KAYNAKLAR

- [1] Moment Expo, Nostalji, <http://www.moment-expo.com/m-oden-gunumuze-robot-tarihi>, (Ziyaret tarihi: 21 01 2014).
- [2] <http://shop.lego.com/en-US/LEGO-MINDSTORMS-NXT-2-0-8547>, (Ziyaret tarihi: 07 03 2014).
- [3] <http://mrrobot.eu/shop/lego-mindstorms-nxt-2-0-8547-robot-toy/>, (Ziyaret tarihi: 07 03 2014).
- [4] <http://www.diseno-art.com/encyclopedia/archive/ASIMO.html>, (Ziyaret tarihi: 07 03 2014).
- [5] Types of Robots, <http://prime.jsc.nasa.gov/ROV/types.html>, (Ziyaret tarihi: 21 04 2014).
- [6] PKM Tricept, <http://www.pkmtricept.com/productos/index.php?id=en&Nproduct=1240413378>, (Ziyaret tarihi: 07 03 2014).
- [7] O'Brien T., Owsome Lego Robots Arm Recalls Origin of Skynet, <http://www.switched.com/2010/03/05/awesome-lego-robot-arm-recalls-origin-of-skynet/>, (Ziyaret tarihi: 07 03 2014).
- [8] Merlet J., *Parallel robots*, 2nd ed., Springer, Netherlands, 2006.
- [9] ABB Robotics Industrial Robots IRB 360, <http://new.abb.com/products/robotics/industrial-robots/irb-360> (Ziyaret tarihi: 07 03 2014).
- [10] <http://www.youtube.com/watch?v=7VxCl6w3HS0> (Ziyaret tarihi: 07 03 2014)
- [11] Anlı E., Alp H., Yurt S. N., Özkol İ., Paralel Mekanizmaların Kinematığı, *Havacılık ve Uzay Teknolojileri Dergisi*, 2005, **2**, 19-36.
- [12] Parallelic, The Paralel Mechanism Information Center, <http://www.parallelic.org/Reviews/Review007p.html>, (Ziyaret tarihi: 22 01 2014).
- [13] İner B., Stewart Platform Benzetim ve Eniyileme Yazılımının Gerçekleştirilmesi, Doktora Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2013, 335442.

- [14] Po-Er H., Yeh-Liang H., Jun-Ming L., Tsai J.-H., Yi-Shin C., iRW: An Intelligent Robotic Wheelchair Integrated with Advanced Robotic and Telehealth Solutions, *Asia Pacific eCare and TeleCare Congress*, Hong Kong, China, 16-19 June 2011.
- [15] Yangmin L, Qingsong X., Kinematic analysis of a 3-PRS parallel manipulator, *Science Direct*, 2007, **23**, 395-408.
- [16] Küçük S., Bingül Z., Robot Sistemlerinde Kinematik Yöntemlerin Karşılaştırılması, *Polytechnic*, 2004, **7**, 107-117.
- [17] Küçük S., Bingül Z., *Robot Kinematiği*, 1st ed., Birsen Yayın Evi, İstanbul, 2009.
- [18] Küçük S., *Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization*, 1st ed., Intech, Croatia, 2012.
- [19] http://projetromeo.com/romeo-documentation/hardware_romeo_joint.html, (Ziyaret tarihi: 26 1 2014).
- [20] Khalil W., İbrahim O., Kinematic and Dynamic Modelling of 3-RPS Parallel Manipulator, *12th IFTOMM World Congress*, Besançon, France, 17-20 June 2007.
- [21] Tombul T. S., Sarıtaş M., Beş Eksenli Bir Edubot Robot Kolunda Ters Kinematik Hesaplamalar ve Yörünge Planlaması, *Gazi Üniv. Müh. Mim. Fak. Der.*, 2003, **18**, 145-167.
- [22] <http://www.kartalotomasyon.com.tr/Pro-3-Eksenli-joystick>, (Ziyaret tarihi: 07 03 2014).
- [23] Analog Devices, http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf, (Ziyaret tarihi: 07 03 2014).
- [24] http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf, (Ziyaret tarihi: 07 03 2014).
- [25] <http://www.lcd-module.de/eng/pdf/zubehoer/t6963.pdf>, (Ziyaret tarihi: 07 03 2014).
- [26] Sparkfun, Dual Bridge Full Bridge Driver, https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf, (Ziyaret tarihi: 07 03 2013).
- [27] Microchip, <http://ww1.microchip.com/downloads/en/DeviceDoc/39761b.pdf>, (Ziyaret tarihi: 07 03 2014).

- [28] Microchip, <http://ww1.microchip.com/downloads/en/DeviceDoc/70290F.pdf>, (Ziyaret tarihi: 07 03 2014).
- [29] Çiçek S., *CCS C ile PIC Programlama*, 5th ed., Altaş Yayınevi, İstanbul, 2009.
- [30] Bolat B., I2C Haberleşmesi, <http://berkbolat.com/i2c-haberlesmesi-2/>, (Ziyaret tarihi: 26 01 2014).
- [31] Wikipedia, the free encyclopedia, <http://en.wikipedia.org/wiki/I%C2%B2C>, (Ziyaret tarihi: 26 01 2014).
- [32] <http://rebrickable.com/parts/61903/Technic-Universal-Joint-3L-%5BComplete-Assembly%5D>, (Ziyaret tarihi: 07 03 2014).
- [33] <http://www.bricklink.com/messageThread.asp?ID=159411>, (Ziyaret tarihi: 07 03 2014).
- [34] Firgelli Technologies Inc., <http://www.firgelli.com/products.php?id=42>, (Ziyaret tarihi: 07 03 2014).
- [35] <https://www.mr-rcworld.co.uk/index.php?productID=1027>, (Ziyaret tarihi: 07 03 2014).
- [36] Arduino, <http://forum.arduino.cc/index.php/topic>, (Ziyaret tarihi: 07 03 2014).
- [37] Long Term Evolution, Teknik Elektro UGM, <http://suatmimurnani.wordpress.com/tag/long-term-evolution/>, (Ziyaret tarihi: 07 03 2014).
- [38] <http://www.proxim.com/products/knowledge-center/calculations/calculations-antenna-downtilt>, (Ziyaret tarihi: 07 03 2014).
- [39] ECLE Communication, <http://www.ecvv.com/product/2340236.html>, (Ziyaret tarihi: 07 03 2014).
- [40] Alpha Wireless Ltd., <http://www.alphaantennas.com/aisg.asp>, (Ziyaret tarihi: 07 03 2014).
- [41] <http://www.boatsafe.com/nauticalknowhow/captpat1.htm>, (Ziyaret tarihi: 16 02 2014).

EKLER



Miniature Linear Motion Series · L16

Firgelli Technologies' unique line of Miniature Linear Actuators enables a new generation of motion-enabled product designs, with capabilities that have never before been combined in a device of this size. These linear actuators are a superior alternative to designing your own push/pull mechanisms.

The L16 actuators are complete, self contained linear motion devices with position feedback for sophisticated position control capabilities, or end of stroke limit switches for simple two position automation. Driving them couldn't be easier, simply apply a DC voltage to extend the actuator, and reverse the polarity to retract it. Several gear ratio's are available to give you varied speed/force configurations.

L16 Specifications

Gearing Option	35:1	63:1	150:1
Peak Power Point	50N @16mm/s	75N @10mm/s	175N @4mm/s
Peak Efficiency Point	24N @24mm/s	38N @15mm/s	75N @7mm/s
Max Speed (no load)	32mm/s	20mm/s	8mm/s
Max Force (lifted)	50N	100N	200N
Back Drive Force	31N	46N	102N
Stroke Option	50mm	100mm	140mm
Mass	56g	74g	84g
Positional Accuracy	0.3mm	0.4mm	0.5mm
Max Side Load (extended)	40N	30N	20N
Feedback Potentiometer	9kΩ±30%	18kΩ±30%	25kΩ±30%
Electrical Stroke	48mm	98mm	138mm
Input Voltage	0-15 VDC. Rated at 12VDC.		
Stall Current	650mA @ 12V		
Operating Temperature	-10°C to +50°C		
Lifetime @ Peak Eff. Pt.	20,000 strokes, 20% Duty Cycle		
Audible Noise	57dB @ 45cm		
Ingress Protection	IP-54		
Mechanical Backlash	0.2mm		
Limit Switches	Max. Current Leakage: 8uA		

Applications

- Robotics
- Consumer appliances
- Toys
- RC vehicles
- Automotive
- Industrial Automation

Basis of Operation

The L16 is designed to push or pull a load along its full stroke length. The speed of travel is determined by the load applied. (See the Load Curves). When power is removed the actuator will hold its position, unless the applied load exceeds the backdrive force. Stalling the actuator for short periods will not cause damage, however repeated stalling will shorten the life of the actuator.

Ordering

Small quantity orders can be placed directly online at www.firgelli.com. Each actuator ships with two mounting brackets and #8-32 mounting hardware. The cable length is approximately 300mm and connector is a 0.1" pitch female socket connector.



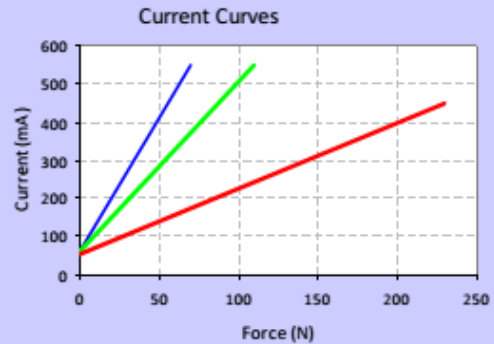
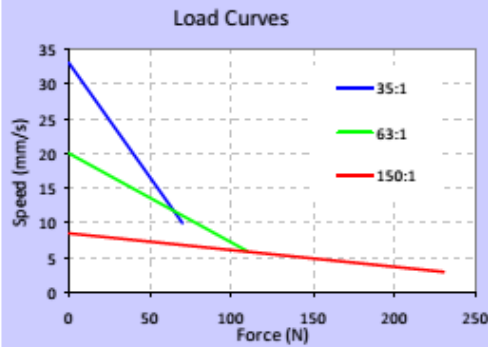
Firgelli Technologies Inc.

1550 Mileva Lane
 Victoria, BC V8N 2V5
 Canada

1 (206) 347-9684 phone
 1 (888) 225-9198 toll-free
 1 (206) 347-9684 fax

sales@firgelli.com
 www.firgelli.com

L16 Specifications



Model Selection

The L16 has 3 configuration choices: Stroke, Gear Ratio and Controller. L16 options are identified according to the following model numbering scheme:

L16-SS-GG-VV-C

feature	options
SS: Stroke	50, 100, 140 (mm)
GG: Gear reduction ratio (refer to load curves above)	35, 63, 150 (lower ratios are faster but push less force, and vice versa)
VV: Voltage	12 (DC volts)
C: Controller	P Potentiometer Feedback S Limit Switches

L16 Controller Options

Option S – End of Stroke Limit Switches

WIRING: (see next page for pin numbering)

- 1 (red) - Actuator Motor Power
- 2 (black) - Actuator Motor Power

The –S actuators have limit switches that will turn off power to the motor when the actuator reaches within 1mm of the end of stroke. Internal diodes allow the actuator to reverse away from the limit switch. The limit switches cannot be moved. While voltage is applied to the motor power pins the actuator extends. Reverse the polarity and the actuator retracts. This can be accomplished manually with a DPDT switch or relay, or using an H-Bridge circuit. The –S model cannot be used with the CIB control board.

Option P – Potentiometer Position Feedback

WIRING: (see next page for pin numbering)

- 1 (orange) – Feedback Potentiometer negative reference rail
- 2 (purple) – Feedback Potentiometer wiper
- 3 (red) – Actuator Motor Power
- 4 (black) – Actuator Motor Power
- 5 (yellow) – Feedback Potentiometer positive reference rail

The –P actuators have no built in controller, but do provide an analog position feedback signal that can be input to an external controller. While voltage is applied to the motor power pins (3 & 4) the actuator extends. Reverse the polarity and the actuator retracts. This can be accomplished manually with a DPDT switch or relay, or using an H-Bridge circuit. Position of the actuator stroke can be monitored by providing any stable low and high reference voltage on pins 1 & 5, then reading the position signal on pin 2. The voltage on pin 2 will vary linearly between the two reference voltages in proportion to the position of the actuator stroke.

The L16 –P actuators can be used as a linear servo by connecting the actuator to a microcontroller such as the CIB control board offered by Firgelli. This control board reads the position signal from the L16, compares it with your input control signal then commands the actuator to move via an on-board H-bridge circuit. The CIB allows any one of the following control inputs: Analog 0-5V or 4-20mA, or Digital 0-5V PWM or 1-2ms standard RC. The RC input effectively transforms your L16 into a linear servo which is a direct replacement for any common hobby servo used in RC toys and robotics. Refer to the CIB datasheet for more details.

Special Notes:

Currently in development are the RC versions of the L16 .

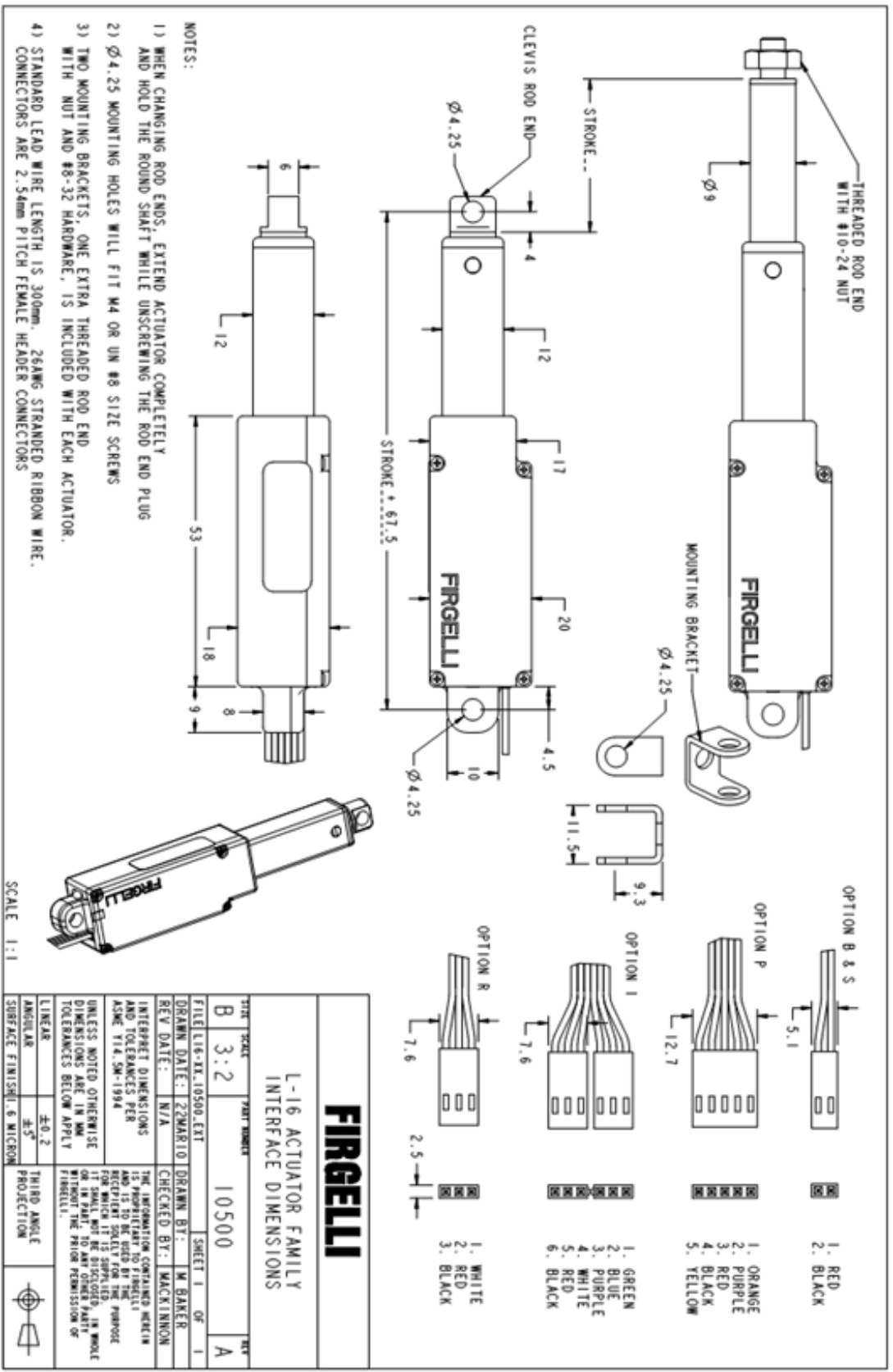


Firgelli Technologies Inc.

1550 Mileva Lane
Victoria, BC V8N 2V5
Canada

1 (206) 347-9684 phone
1 (888) 225-9198 toll-free
1 (206) 347-9684 fax

sales@firgelli.com
www.firgelli.com



NOTES:

1) WHEN CHANGING ROD ENDS, EXTEND ACTUATOR COMPLETELY AND HOLD THE ROUND SHAFT WHILE UNSCREWING THE ROD END PLUG

2) Ø4.25 MOUNTING HOLES WILL FIT M4 OR UN #8 SIZE SCREWS

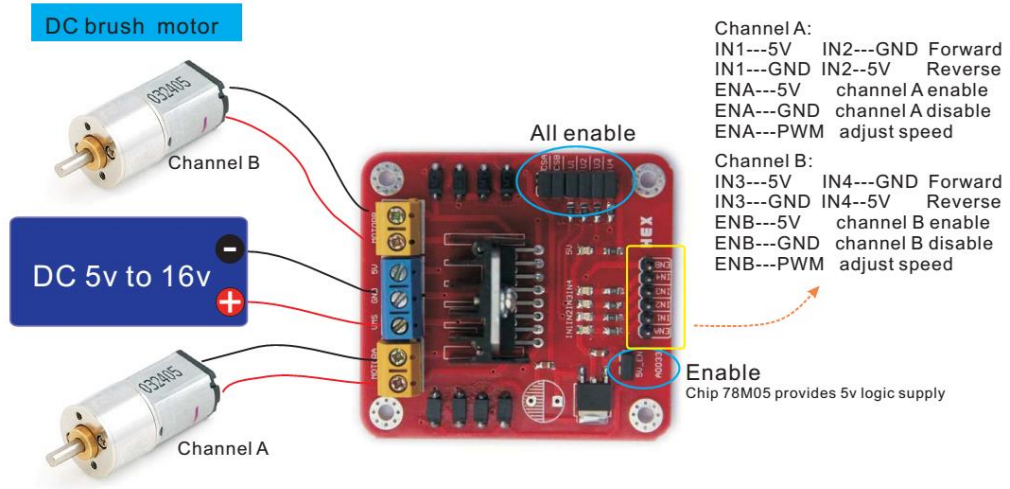
3) TWO MOUNTING BRACKETS, ONE EXTRA THREADED ROD END WITH NUT AND #8-32 HARDWARE, IS INCLUDED WITH EACH ACTUATOR.

4) STANDARD LEAD WIRE LENGTH IS 300mm, 26AWG STRANDED RIBBON WIRE. CONNECTORS ARE 2.54mm PITCH FEMALE HEADER CONNECTORS

FIRGELLI			
L-16 ACTUATOR FAMILY INTERFACE DIMENSIONS			
FILE	L16-IT-10500-EXT	SHEET	1 OF 1
DRAWN DATE:	22/MAR/10	CHECKED BY:	M BAKER
REV DATE:	N/A	INTERPRET DIMENSIONS AND TOLERANCES PER	THE INFORMATION CONTAINED HEREIN IS PROPRIETARY TO FIRGELLI AND IS TO BE USED BY THE PURCHASER ONLY FOR THE PURPOSES SPECIFIED IN THE ORDER. IT SHALL NOT BE DISCLOSED, IN WHOLE OR IN PART, TO ANY OTHER PARTY WITHOUT THE PRIOR PERMISSION OF FIRGELLI.
LINEAR	±0.2	ANGULAR	±5°
SURFACE FINISH	0.6 MICRON	THREAD ANGLE	PROTECTION
TITLE	SCALE	PART NUMBER	REV
B	3:2	10500	A

L298 Dual H-Bridge Motor Driver

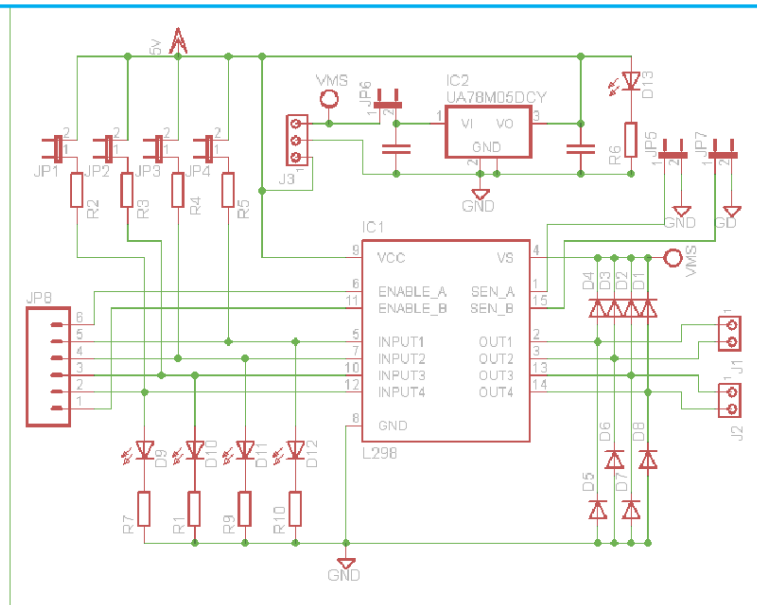
User's Guide



L298 Dual H-Bridge Motor Driver

User's Guide

Schematic:



EK C

```
#include <33FJ32GP202.h>
#device ADC=10 //10 bitlik ADC kullanılacağı belirtiliyor
#fuses FRC
//#use delay(clock=80M,oscillator=10M)
#use delay(clock=7.37M,internal)

//I2C
#use I2C(SLAVE, SCL=PIN_B8, SDA=PIN_B9, ADDRESS=0x14, SLOW=100000)

#include "LCD_FLEX.c"
#include "MATH.h"

#define servo      pin_a4

#define in0_0 pin_a2
#define in0_1 pin_b14
#define in1_0 pin_b13
#define in1_1 pin_b12
#define in2_0 pin_b11
#define in2_1 pin_b10

//GENERAL VARIABLES
unsigned int16 tmp16;

int1 tmr1First=true;
int1 commStarted=false;
unsigned int cntr=0;
byte getData[4];
byte state, incoming;
signed int modeNo,Xrcv,Yrcv,Zrcv;
unsigned int16 servoZ=0;
int sense;
int1 reviseMove;
unsigned int16 modeWaitms;
signed int workingMode;

signed int deltaD0,deltaD1,deltaD2;

//REVERSE KYNEMATICS VARIABLES
float a0[3];
float a1[3];
float a2[3];

float b0[3];
float b1[3];
float b2[3];

float p[3];

float g,h,l;
float armLength;

float q0[3];
float q1[3];
float q2[3];
```

```

float d[3];

float aRotB[3][3];
float Qxk,Qyk,Qzk;

//SET POSITION VARIABLES
signed int d_f[3],d_sudden[3];
signed int dt[3][10];
signed int32 d0[3];
signed int16 df[3];
signed int16 tf;
int step[4];
int1 dMoveDone[4];
int stepCount;

unsigned int distX=0;
unsigned int distY=0;
unsigned int distZ=0;
unsigned int minDistRate=0;

unsigned int rateZtoX;
unsigned int rateZtoY;

unsigned int16 servoWait=0;
unsigned int16 servoCount=0;

signed int Xstop,Ystop,Zstop;
signed int Xdest,Ydest,Zdest;

signed int Xn[10],Yn[10],Zn[10];
int16 servoT[10];

#int_si2c //Interrupt for I2C activity
void sspinterrupt()
{
    state = i2c_isr_state();

    if(state < 0x80) //master is sending data
    {
        incoming = i2c_read();

        if(state == 1)
        {
            getData[0] = incoming;
        }
        else if(state == 2) //first received byte is address
        {
            getData[1] = incoming;
        }
        else if(state == 3) //second received byte is data
        {
            getData[2] = incoming;
        }
        else if(state == 4) //third received byte is data
        {
            getData[3] = incoming;
        }
    }
}

```

```

    }

    if(state == 0x80)                //master is requesting data
    {
        i2c_write(2);
    }

    modeNo=getData[0];
    Xrcv=(getData[1]-90);
    Yrcv=(getData[2]-90);
    Zrcv=(getData[3]-90);
    commStarted=true;
}

#int_timer1
//Servo Duty Cycle
void tmr1Int()
{
    if (tmr1First==true)
    {
        output_high(servo);
        tmr1First=false;
    }
    else
    {
        output_low(servo);
    }
}

#int_timer2
//20ms Servo Period
void tmr2Int()
{
    tmr1First=true;

    cntr++;
    if (cntr==50)
    {
        //output_toggle(prcss33FJ);
        cntr=0;
    }
}

//*****
//*****
//*****PARAMETER
FUNCTIONS*****
*
//*****
//*****
signed int getArmPosition(int dNum)
{
    set_adc_channel(dNum);
    return floor((100.00/1023.00)*read_adc()-49.5);
}

void drawPositionFeedbacks()
{

```

```

//WRITE POSITION FEEDBACK
printf(lcd_putc, "\f");
printf(lcd_putc, "d0:%d ", getArmPosition(0));
printf(lcd_putc, "d1:%d", getArmPosition(1));
lcd_gotoxy(1,2);
printf(lcd_putc, "d2:%d", getArmPosition(5));
}

void printDegrees()
{
printf(lcd_putc, "\f");
printf(lcd_putc, "M:%d ", modeNo);

if (Xrcv==165)
{
printf(lcd_putc, "Xr:N/A");
}
else
{
printf(lcd_putc, "Xr:%d", Xrcv);
}

lcd_gotoxy(1,2);

if (Yrcv==165)
{
printf(lcd_putc, "Yr:N/A ");
}
else
{
printf(lcd_putc, "Yr:%d ", Yrcv);
}

if (Zrcv==165)
{
printf(lcd_putc, "Zr:N/A");
}
else
{
printf(lcd_putc, "Zr:%d", Zrcv);
}
}

//*****
//*****
//*****MOVEMENT
FUNCTIONS*****
**
//*****
//*****
void setReverseKynematics(signed int xDeg, signed int yDeg, signed int zDeg)
{
Qxk=xDeg/57.32;
Qyk=yDeg/57.32;
Qzk=zDeg/57.32;

aRotB[0][0]=cos(Qzk)*cos(Qyk);
aRotB[0][1]=cos(Qzk)*sin(Qyk)*sin(Qxk)-sin(Qzk)*cos(Qxk);
aRotB[0][2]=cos(Qzk)*sin(Qyk)*cos(Qxk)+sin(Qzk)*sin(Qxk);

```



```

aRotB[1][0]=sin(Qzk)*cos(Qyk) ;
aRotB[1][1]=sin(Qzk)*sin(Qyk)*sin(Qxk)+cos(Qzk)*cos(Qxk);
aRotB[1][2]=sin(Qzk)*sin(Qyk)*cos(Qxk)-cos(Qzk)*sin(Qxk);

aRotB[2][0]=-sin(Qyk);
aRotB[2][1]=cos(Qyk)*sin(Qxk);
aRotB[2][2]=cos(Qyk)*cos(Qxk);

q0[0]=P[0] + aRotB[0][0]*b0[0] + aRotB[0][1]*b0[1] + aRotB[0][2]*b0[2];
q0[1]=P[1] + aRotB[1][0]*b0[0] + aRotB[1][1]*b0[1] + aRotB[1][2]*b0[2];
q0[2]=P[2] + aRotB[2][0]*b0[0] + aRotB[2][1]*b0[1] + aRotB[2][2]*b0[2];

q1[0]=P[0] + aRotB[0][0]*b1[0] + aRotB[0][1]*b1[1] + aRotB[0][2]*b1[2];
q1[1]=P[1] + aRotB[1][0]*b1[0] + aRotB[1][1]*b1[1] + aRotB[1][2]*b1[2];
q1[2]=P[2] + aRotB[2][0]*b1[0] + aRotB[2][1]*b1[1] + aRotB[2][2]*b1[2];

q2[0]=P[0] + aRotB[0][0]*b2[0] + aRotB[0][1]*b2[1] + aRotB[0][2]*b2[2];
q2[1]=P[1] + aRotB[1][0]*b2[0] + aRotB[1][1]*b2[1] + aRotB[1][2]*b2[2];
q2[2]=P[2] + aRotB[2][0]*b2[0] + aRotB[2][1]*b2[1] + aRotB[2][2]*b2[2];

d[0]=sqrt((q0[0]-a0[0])*(q0[0]-a0[0]) + (q0[1]-a0[1])*(q0[1]-a0[1]) +
(q0[2]-a0[2])*(q0[2]-a0[2]));
d[1]=sqrt((q1[0]-a1[0])*(q1[0]-a1[0]) + (q1[1]-a1[1])*(q1[1]-a1[1]) +
(q1[2]-a1[2])*(q1[2]-a1[2]));
d[2]=sqrt((q2[0]-a2[0])*(q2[0]-a2[0]) + (q2[1]-a2[1])*(q2[1]-a2[1]) +
(q2[2]-a2[2])*(q2[2]-a2[2]));

deltaD0=floor(d[0]*10+0.5-armLength);
deltaD1=floor(d[1]*10+0.5-armLength);
deltaD2=floor(d[2]*10+0.5-armLength);

Xdest=xDeg;
Ydest=yDeg;
Zdest=zDeg;
}

void setPosition(signed int df0,signed int df1,signed int df2,)
{
    int i,j;

    stepCount=0;

    //DEGREE PLANNING
    for (i=1;i<11;i++)
    {
        Xn[i-1]= (float)Xstop + floor((float)(Xdest-Xstop)/9.00 * ((float)i-
1.00));
        Yn[i-1]= (float)Ystop + floor((float)(Ydest-Ystop)/9.00 * ((float)i-
1.00));
        Zn[i-1]= (float)Zstop + floor((float)(Zdest-Zstop)/9.00 * ((float)i-
1.00));
    }

    //CHECK IF THE MODE CHANGED
    if (workingMode!=modeNo)
    {
        goto skipSetPosition;
    }

    //CHECK IF THE MODE CHANGED

```

```

if (workingMode!=modeNo)
{
    goto skipSetPosition;
}

//CHECK IF THE DESTINATION CHANGED MORE THAN 5 DEGREE
if (workingMode==1)
{
    //ZERO POSITION
}
else if (modeNo==2)
{
    //WORKSPACE
}
else if (modeNo==5)
{
    //IDLE MODE
}
else if (workingMode==4)
{
    //PARALLEL MODE
    if (abs(Xdest-(-1*Xrcv))>sense || abs(Ydest-(-1*Yrcv))>sense)
    {
        Xstop=Xn[stepCount];
        Ystop=Yn[stepCount];
        Zstop=Zn[stepCount];
        reviseMove=true;
        goto skipSetPosition;
    }
}
else
{
    //JOYSTICK MODE
    //SENSOR MODE
    if (abs(Xdest-Xrcv)>sense || abs(Ydest-Yrcv)>sense || abs(Zdest-
Zrcv)>sense)
    {
        Xstop=Xn[stepCount];
        Ystop=Yn[stepCount];
        Zstop=Zn[stepCount];
        reviseMove=true;
        goto skipSetPosition;
    }
}

d0[0]=getArmPosition(0);
d0[1]=getArmPosition(1);
d0[2]=getArmPosition(5);

df[0]=df0;
df[1]=df1;
df[2]=df2;

//FIRGELLI
for (j=0;j<3;j++)
{
    for (i=1;i<11;i++)
    {
        //ORBIT PLANNING

```

```

        dt[j][i-1]=floor((d0[j] + (3*(df[j]-d0[j])*(i*i))/(tf*tf) -
(2*(df[j]-d0[j])*(i*i*i))/(tf*tf*tf)) + 0.5);
    }
}

//SERVO PWM PLANNING
for (i=1;i<11;i++)
{
    servoT[i-1]=970 + floor(7.50*(float)Zn[i-1]);
}

//SERVO SPEED PLANNING
if (Xstop>Xdest)
{distX=Xstop-Xdest;}
else
{distX=Xdest-Xstop;}

if (Ystop>Ydest)
{distY=Ystop-Ydest;}
else
{distY=Ydest-Ystop;}

if (Zstop>Zdest)
{distZ=Zstop-Zdest;}
else
{distZ=Zdest-Zstop;}

if (distX==0)
{distX=1;}
if (distY==0)
{distY=1;}

rateZtoX=floor((float)distZ/(float)distX);
rateZtoY=floor((float)distZ/(float)distY);

if (rateZtoX<rateZtoY)
{minDistRate=rateZtoX;}
else
{minDistRate=rateZtoY;}
servoWait=minDistRate*2;

//MOVEMENT VARIABLES
d_f[0]=floor(df[0]);
d_f[1]=floor(df[1]);
d_f[2]=floor(df[2]);

d_sudden[0]=floor(d0[0]);
d_sudden[1]=floor(d0[1]);
d_sudden[2]=floor(d0[2]);

step[0]=0;
step[1]=0;
step[2]=0;
step[3]=0;

dMoveDone[0]=false;
dMoveDone[1]=false;
dMoveDone[2]=false;
dMoveDone[3]=false;

```

```

servoCount=0;

while(dMoveDone[0]==false || dMoveDone[1]==false || dMoveDone[2]==false ||
dMoveDone[3]==false)
{

//D0*****
*****

//*****
*****

//*****
*****

if (d_sudden[0]!=d_f[0] && dMoveDone[0]==false)
{
if (step[0]<stepCount+1)
{
if (d_f[0] > d_sudden[0])
{
output_high(in0_0);
output_low(in0_1);

if (d_sudden[0]>=dt[0][stepCount])
{
step[0]=stepCount+1;
}
}
else if (d_f[0] < d_sudden[0])
{
output_low(in0_0);
output_high(in0_1);

if (d_sudden[0]<=dt[0][stepCount])
{
step[0]=stepCount+1;
}
}
}
else
{
output_low(in0_0);
output_low(in0_1);
}
}
else
{
output_low(in0_0);
output_low(in0_1);

dMoveDone[0]=true;
step[0]=9;
}

//*****
*****

//*****
*****

```

```

//*****
*****

//D1*****
*****

//*****
*****

//*****
*****
    if (d_sudden[1]!=d_f[1] && dMoveDone[1]==false)
    {
        if (step[1]<stepCount+1)
        {
            if (d_f[1] > d_sudden[1])
            {
                output_high(in1_0);
                output_low(in1_1);

                if (d_sudden[1]>=dt[1][stepCount])
                {
                    step[1]=stepCount+1;
                }
            }
            else if (d_f[1] < d_sudden[1])
            {
                output_low(in1_0);
                output_high(in1_1);

                if (d_sudden[1]<=dt[1][stepCount])
                {
                    step[1]=stepCount+1;
                }
            }
        }
        else
        {
            output_low(in1_0);
            output_low(in1_1);
        }
    }
    else
    {
        output_low(in1_0);
        output_low(in1_1);

        dMoveDone[1]=true;
        step[1]=9;
    }

//*****
*****

//*****
*****

```

```

//*****
*****

//D2*****
*****

//*****
*****

//*****
*****
    if (d_sudden[2]!=d_f[2] && dMoveDone[2]==false)
    {
        if (step[2]<stepCount+1)
        {
            if (d_f[2] > d_sudden[2])
            {
                output_high(in2_0);
                output_low(in2_1);

                if (d_sudden[2]>=dt[2][stepCount])
                {
                    step[2]=stepCount+1;
                }
            }
            else if (d_f[2] < d_sudden[2])
            {
                output_low(in2_0);
                output_high(in2_1);

                if (d_sudden[2]<=dt[2][stepCount])
                {
                    step[2]=stepCount+1;
                }
            }
        }
        else
        {
            output_low(in2_0);
            output_low(in2_1);
        }
    }
    else
    {
        output_low(in2_0);
        output_low(in2_1);

        dMoveDone[2]=true;
        step[2]=9;
    }

//*****
*****

//*****
*****

```

```

//*****
*****

//SERVO*****
*****

//*****
*****

//*****
*****
    if (dMoveDone[3]==false)
    {
        if (step[3]<stepCount+1)
        {
            if (step[3]==9)
            {
                dMoveDone[3]=true;
            }
            else
            {
                servoCount++;
                if (servoCount>=servoWait)
                {
                    step[3]++;
                    setup_timer1(TMR_INTERNAL|TMR_DIV_BY_64,servoT[step[3]]);
                    servoCount=0;
                }
            }
        }
    }
    //output_toggle(prcss33FJ);

//SERVO*****
*****

//*****
*****

//*****
*****

//STEPCOUNT=MIN(STEP[N])
stepCount=step[0];
for (i=0;i<3;i++)
{
    if (step[i+1]<step[i])
    {
        stepCount=step[i+1];
    }
}

//CHECK IF THE MODE CHANGED
if (workingMode!=modeNo)
{
    goto skipSetPosition;
}

```

```

}

//CHECK IF THE DESTINATION CHANGED MORE THAN 5 DEGREE
if (workingMode==1)
{
    //ZERO POSITION
}
else if (modeNo==2)
{
    //WORKSPACE
}
else if (modeNo==5)
{
    //IDLE MODE
}
else if (workingMode==4)
{
    //PARALLEL MODE
    if (abs(Xdest-(-1*Xrcv))>sense || abs(Ydest-(-1*Yrcv))>sense)
    {
        Xstop=Xn[stepCount];
        Ystop=Yn[stepCount];
        Zstop=Zn[stepCount];
        reviseMove=true;
        goto skipSetPosition;
    }
}
else
{
    if (abs(Xdest-Xrcv)>sense || abs(Ydest-Yrcv)>sense || abs(Zdest-
Zrcv)>sense)
    {
        Xstop=Xn[stepCount];
        Ystop=Yn[stepCount];
        Zstop=Zn[stepCount];
        reviseMove=true;
        goto skipSetPosition;
    }
}

d_sudden[0]=getArmPosition(0);
d_sudden[1]=getArmPosition(1);
d_sudden[2]=getArmPosition(5);
}

skipSetPosition:
if (stepCount==9)
{
    Xstop=Xdest;
    Ystop=Ydest;
    Zstop=Zdest;

    output_low(in0_0);
    output_low(in0_1);
    output_low(in1_0);
    output_low(in1_1);
    output_low(in2_0);

```



```

        output_low(in2_1);

        printf(lcd_putc, "\fTamam:%d", stepCount);
    }
    else
    {
        Xstop=Xn[stepCount];
        Xstop=Xn[stepCount];
        Xstop=Xn[stepCount];

        output_low(in0_0);
        output_low(in0_1);
        output_low(in1_0);
        output_low(in1_1);
        output_low(in2_0);
        output_low(in2_1);

        printf(lcd_putc, "\fYarim:%d", stepCount);
    }
}

}

//*****
//*****
//*****MODE
FUNCTIONS*****
//*****
//*****
void modeJoystick()
{
    workingMode=modeNo;
    sense=2;

    printf(lcd_putc, "\fJoystick Mode");
    delay_ms(modeWaitms);

    while (modeNo==0)
    {
        if (abs(Xstop-Xrcv)>sense || abs(Ystop-Yrcv)>sense || abs(Zstop-
Zrcv)>sense)
        {
            repeat:
            reviseMove=false;

            setReverseKynematics(Xrcv,Yrcv,Zrcv);
            setPosition(deltaD0,deltaD1,deltaD2);

            if (reviseMove)
            {
                goto repeat;
            }
        }
        delay_ms(1);
    }
}
}

```

```

void modeZeroPosition()
{
    workingMode=modeNo;
    printf(lcd_putc, "\fZero Position");
    delay_ms(modeWaitms);

    delay_ms(modeWaitms);
    if (modeNo==1)
    {
        setPosition(0,0,0);
    }

    while (modeNo==1)
    {
        //Codes Here
    }
}

void modeWorkSpace()
{
    workingMode=modeNo;
    printf(lcd_putc, "\fWorkSpace Mode");
    delay_ms(modeWaitms);
    printf(lcd_putc, "\fWorkSpace Mode");

    if (modeNo==2)
    {
        signed int deltas[3][73];
        unsigned int16 k;

        for (k=0;k<72;k++)
        {
            deltas[0][k]=floor(35.0*sin((k%360)*10/57.32)+0.5);
            deltas[1][k]=floor(35.0*sin(((k+120)%360)*10/57.32)+0.5);
            deltas[2][k]=floor(35.0*sin(((k+240)%360)*10/57.32)+0.5);
        }

        //WS MOVEMENT1
        for (k=0;k<72;k++)
        {
            setPosition(deltas[0][k],deltas[1][k],deltas[2][k]);
        }

        while (modeNo==2);
    }
}

void modeSensor()
{
    workingMode=modeNo;
    sense=5;

    printf(lcd_putc, "\fSensor Mode");
    delay_ms(modeWaitms);

    while (modeNo==3)
    {
        if (abs(Xstop-Xrcv)>sense || abs(Ystop-Yrcv)>sense || abs(Zstop-
Zrcv)>sense)
        {

```

```

        repeat:
        reviseMove=false;

        setReverseKinematics(Xrcv,Yrcv,Zrcv);
        setPosition(deltaD0,deltaD1,deltaD2);

        if (reviseMove)
        {
            goto repeat;
        }
    }
    delay_ms(1);
}

void modeParallel()
{
    workingMode=modeNo;
    sense=5;

    printf(lcd_putc, "\fParallel Mode");
    delay_ms(modeWaitms);

    while (modeNo==4)
    {
        if (abs(Xstop-Xrcv)>sense || abs(Ystop-Yrcv)>sense)
        {
            repeat:
            reviseMove=false;

            setReverseKinematics((-1*Xrcv),(-1*Yrcv),0);
            setPosition(deltaD0,deltaD1,deltaD2);

            if (reviseMove)
            {
                goto repeat;
            }
        }
        delay_ms(1);
    }
}

void modeIdle()
{
    workingMode=modeNo;
    printf(lcd_putc, "\fIdle Mode");
    delay_ms(modeWaitms);

    while (modeNo==5)
    {
        drawPositionFeedbacks();
        delay_ms(4000);

        //Codes Here
    }
}

void callMode()
{
    if (modeNo==0)

```

```

    {
        modeJoystick();
    }
    else if (modeNo==1)
    {
        modeZeroPosition();
    }
    else if (modeNo==2)
    {
        modeWorkSpace();
    }
    else if (modeNo==3)
    {
        modeSensor();
    }
    else if (modeNo==4)
    {
        modeParallel();
    }
    else if (modeNo==5)
    {
        modeIdle();
    }
}

```

```

//*****
*****
//*****MAIN
FUNCTIONS*****
*****
//*****
*****
void initialConf()
{
    //PIN SETTINGS
    //output_low(prcss33FJ);
    output_low(in0_0);
    output_low(in0_1);
    output_low(in1_0);
    output_low(in1_1);
    output_low(in2_0);
    output_low(in2_1);

    //CONFIGURATIONS
    setup_adc_ports(NO_ANALOGS|VSS_VDD);
    setup_adc(ADC_OFF);

    //PLL 40MIPS (clock=80M,oscillator=7.37M)
    setup_oscillator(OSC_INTERNAL,8000000);
    delay_ms(100);
    setup_oscillator(OSC_INTERNAL,8000000);
    delay_ms(100);
    setup_oscillator(OSC_INTERNAL,8000000);
    delay_ms(100);

    //LCD SETTINGS
    lcd_init();
    delay_ms(20);
}

```

```

printf(lcd_putc, "ACIL");
delay_ms(3000);

//TIMER SETTINGS
//TIMER1 PWM PERIOD
//1ms -70 degree
//setup_timer1(TMR_INTERNAL|TMR_DIV_BY_64,443);
//2ms +70 degree
//setup_timer1(TMR_INTERNAL|TMR_DIV_BY_64,1500);
//1.5ms 0 degree
setup_timer1(TMR_INTERNAL|TMR_DIV_BY_64,970);
//TIMER2 PWM PERIOD 20ms
setup_timer2(TMR_INTERNAL|TMR_DIV_BY_64,12865);

//ANALOG SETTINGS
setup_adc(ADC_OFF); // ADC modülü kapalı
delay_ms(20);
setup_adc(ADC_CLOCK_INTERNAL); // ADC için dahili clock kullanılacak
delay_ms(20);
setup_adc(ADC_CLOCK_DIV_8); // Clock 8'e bölünerek kullanılacak
delay_ms(20);
setup_adc_ports(ALL_ANALOG|VSS_VDD);

//VARIABLES
getData[0]=0;
getData[1]=0;
getData[2]=0;
getData[3]=0;

//KYNEMATIC INITIALS
g=18.5;
h=7;
l=28;
armLength=303;

a0[0]=g;
a0[1]=0;
a0[2]=0;
a1[0]=-g/2;
a1[1]=sqrt(3)*g/2;
a1[2]=0;
a2[0]=-g/2;
a2[1]=-sqrt(3)*g/2;
a2[2]=0;

b0[0]=h;
b0[1]=0;
b0[2]=0;
b1[0]=-h/2;
b1[1]=sqrt(3)*h/2;
b1[2]=0;
b2[0]=-h/2;
b2[1]=-sqrt(3)*h/2;
b2[2]=0;

p[0]=0;
p[1]=0;
p[2]=1;

```

```

//SET POSITION INITIALS
tf=10;

workingMode=5;
modeNo=5;

servoZ=970;
sense=5;
commStarted=false;
reviseMove=false;
modewaitms=4000;

Xstop=0;
Ystop=0;
Zstop=0;

Xdest=0;
Ydest=0;
Zdest=0;

Xrcv=0;
Yrcv=0;
Zrcv=0;

//BASLANGIC POZISYONU Zstop
set_adc_channel(9);
tmp16=read_adc();
if (tmp16<350)
{
    Zstop=floor(((float)read_adc()-(float)350)/1.60);
}
else
{
    Zstop=floor(((float)read_adc()-(float)350)/2.20);
}

servoT[0]=970 + floor(7.50*(float)Zstop);
setup_timer1(TMR_INTERNAL|TMR_DIV_BY_64,servoT[0]);

delay_ms(400);

//INTERRUPT SETTINGS
enable_interrupts(INT_TIMER1);
enable_interrupts(INT_TIMER2);
enable_interrupts(INTR_GLOBAL);

setPosition(0,0,0);

//START COMMUNICATION
enable_interrupts(INT_SI2C); //enable I2C interrupts
}

void main()
{
    initialConf();

    //while(commStarted==false);

    while(true)
    {

```

```
    callMode();  
  }  
}
```

EK D

```
#include <18F4685.h>
#device ADC=10 //10 bitlik ADC kullanılacağı belirtiliyor
#FUSES HS
#FUSES NOWDT //No Watch Dog Timer
#FUSES WDT128 //Watch Dog Timer uses 1:128 Postscale
#FUSES PUT //Power Up Timer
#FUSES NOBROWNOUT //No brownout reset
#FUSES NOPBADEN //PORTB pins are configured as digital I/O on
RESET
#FUSES NOLVP //No low voltage prgming, B3(PIC16) or
B5(PIC18) used for I/O
#FUSES NOXINST //Extended set extension and Indexed
Addressing mode disabled (Legacy mode)

#use delay(clock=20M)

//I2C
#use I2C(MASTER, SCL=PIN_C3, SDA=PIN_C4, FORCE_HW, SLOW=100000)

//LIBRARIES
#include "T6963C.c"
#include "kocaeli_240_128.h"
#include "MATH.h"
#include "ADXL345.c"
#include "HMC5883L.c"
#include "I2C_18F.c"

//PIN DEFINITIONS
#define prcss18f pin_c2
#define modeSelect pin_c0

signed int Xj=0,Yj=0,Zj=0;
signed int qX,qY,qZ;
byte Xsend=0,Ysend=0,Zsend=0;
signed int16 qqZ;
float hz = 0.0;

byte modeNo=5;
unsigned int16 tmr1=33536;
//unsigned int16 tmr1=1036;

float totalX=0.00,totalY=0.00,totalZ=0.00;

byte destDevWrtAdd=0x14;
byte destDevReadAdd=0x15;

byte reqData=0;
byte data[4];

char tmp[]={ "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" };
unsigned int16 tmpInt16=0;

#int_timer1
void tmr1Int()
```



```

{
    output_toggle(prcss18f);
    disable_interrups(int_timer1);

    send_i2c(destDevWrtAdd,modeNo,Xsend,Ysend,Zsend);

    enable_interrups(int_timer1);
    set_timer1(tmr1);
}

signed int getArmPosition(int dNum)
{
    set_adc_channel(dNum);
    return floor((100.00/1023.00)*read_adc()-49.5);
}

void getJoystickPosition()
{
    set_adc_channel(3);
    tmpInt16=read_adc();

    //Xj
    if (tmpInt16==484)
    {
        Xj=0;
    }
    else if (tmpInt16<484)
    {
        Xj= floor(((484-(float)tmpInt16)*50/484)*-1+0.5);
    }
    else if (tmpInt16>484)
    {
        Xj= floor((((float)tmpInt16-484)*50/539)+0.5);
    }
    Xsend=Xj+90;

    set_adc_channel(4);
    tmpInt16=read_adc();
    //Yj
    if (tmpInt16==502)
    {
        Yj=0;
    }
    else if (tmpInt16<502)
    {
        Yj= floor(((502-(float)tmpInt16)*50/502)+0.5);
    }
    else if (tmpInt16>502)
    {
        Yj= floor((((float)tmpInt16-502)*50/521)*-1+0.5);
    }
    Ysend=Yj+90;

    set_adc_channel(5);
    tmpInt16=read_adc();
    //Zj
    if (tmpInt16==524)
    {
        Zj=0;
    }
}

```

```

    }
    else if (tmpInt16<524)
    {
        Zj= floor(((524-(float)tmpInt16)*50/524)*-1+0.5);
    }
    else if (tmpInt16>524)
    {
        Zj= floor((((float)tmpInt16-524)*50/499)+0.5);
    }
    Zsend=Zj+90;
}

long getDegree(int8 msb,int8 lsb)
{
    int16 x;
    long retVal;

    x=adx1345_read(msb);
    x=(x<<8)+adx1345_read(lsb);

    if(x>60000)
    {
        retVal=x-65535;
        retVal=-1*retVal;
        retVal=ceil((retVal*90/136) - 0.5);
        retVal=-1*retVal;
    }
    else
    {
        retVal=x;
        retVal=floor((retVal*90/136) + 0.5);
    }
    return retVal;
}

void imInstall(int boyut_x, int boyut_y, int16 nereden_x, int16 nereden_y)
{
    int sutun,satir,sutun_boy,satir_boy;
    int16 konum;

    konum = 0;

    sutun_boy = boyut_x / 8;
    satir_boy = boyut_y;

    for(satir=0; satir < satir_boy ; satir++)
    {
        for(sutun=0; sutun < sutun_boy; sutun++)
        {
            glcd_pixel8(nereden_x + (sutun * 8), nereden_y + satir,
            resimler[konum]);
            konum++;
        }
    }
}

void drawKocaeliLogo()

```

```

{
    port_b_pullups (TRUE);
    glcd_init(240,128); // Grafik LCD ekran hazırlanıyor
    glcd_WriteByte(1, (LCDDispMode|LCDDisp_TXT|LCDDisp_GRH)); // Grafik LCD
    göre ayarlanıyor
    glcd_WriteByte(1, (LCDDispMode|LCDDisp_TXT|LCDDisp_GRH)); // Grafik LCD
    ekran hem metin hem grafik modunda çalışacak şekilde ayarlanıyor

    imInstall(240,128,0,0); // Giriş resmi çağrılıp ekrana basılıyor. Resimler
    "Resimler.h" dosyasında tek sabit arrayde saklanıyor,
    delay_ms(2000);
}

void drawGrid()
{
    glcd_init(240,128);
    glcd_WriteByte(1, (LCDDispMode|LCDDisp_TXT|LCDDisp_GRH)); // Grafik LCD
    göre ayarlanıyor
    glcd_WriteByte(1, (LCDDispMode|LCDDisp_TXT|LCDDisp_GRH)); // Grafik LCD
    ekran hem metin hem grafik modunda çalışacak şekilde ayarlanıyor

    //1.kolon
    glcd_square(1,1,50,11,1);
    glcd_square(1,13,50,23,1);
    glcd_square(1,25,50,35,1);
    glcd_square(0,0,51,36,1);

    glcd_square(1,42,50,52,1);
    glcd_square(1,54,50,64,1);
    glcd_square(1,66,50,76,1);
    glcd_square(0,41,51,77,1);

    glcd_square(1,84,50,94,1);
    glcd_square(1,96,50,106,1);
    glcd_square(1,108,50,118,1);
    glcd_square(0,83,51,119,1);

    //2. kolon
    glcd_square(56,1,105,11,1);
    glcd_square(56,13,105,23,1);
    glcd_square(56,25,105,35,1);
    glcd_square(55,0,106,36,1);

    glcd_square(56,42,105,52,1);
    glcd_square(56,54,105,64,1);
    glcd_square(56,66,105,76,1);
    glcd_square(55,41,106,77,1);

    glcd_square(56,84,105,94,1);
    glcd_square(56,96,105,106,1);
    glcd_square(56,108,105,118,1);
    glcd_square(55,83,106,119,1);
}

void drawRobot()
{
    glcd_line(130,108,230,108,1,1);
    glcd_line(130,109,230,109,1,1);
    glcd_line(130,108,160,28,1,1);
}

```

```

    glcd_line(230,108,200,28,1,1);
    glcd_line(160,28,200,28,1,1);
    glcd_line(180,28,180,108,1,1);
}

void drawLoading()
{
    glcd_init(240,128);
    glcd_WriteByte(1, (LCDDisplayMode|LCDDisplayMode_XOR)); // Grafik LCD ekran XOR moda göre ayarlanıyor
    glcd_WriteByte(1, (LCDDisplayMode|LCDDisplayMode_TXT|LCDDisplayMode_GRH)); // Grafik LCD ekran hem metin hem grafik modunda çalışacak şekilde ayarlanıyor

    glcd_square(20,100,220,120,1);
    delay_ms(100);

    tmp="PORTLAR TANIMLANIYOR";
    glcd_text57(10,40,tmp,2,1);
    delay_ms(500);

    glcd_box(10,40,239,55,0);
    glcd_box(20,100,60,120,1);

    tmp="I2C AYARLARI";
    glcd_text57(50,40,tmp,2,1);
    tmp="TANIMLANIYOR";
    glcd_text57(50,60,tmp,2,1);
    delay_ms(500);

    glcd_box(50,40,200,75,0);
    glcd_box(61,100,120,120,1);

    tmp="ANALOGLAR PORTLAR";
    glcd_text57(30,40,tmp,2,1);
    tmp="TANIMLANIYOR";
    glcd_text57(50,60,tmp,2,1);
    delay_ms(500);

    glcd_box(30,40,230,75,0);
    glcd_box(121,100,160,120,1);

    tmp="TIMER AYARLARI";
    glcd_text57(40,40,tmp,2,1);
    tmp="TANIMLANIYOR";
    glcd_text57(50,60,tmp,2,1);
    delay_ms(500);

    glcd_box(40,40,220,75,0);
    glcd_box(161,100,200,120,1);

    tmp="KESMELER TANIMLANIYOR";
    glcd_text57(5,40,tmp,2,1);
    delay_ms(500);

    glcd_box(0,40,240,75,0);
    glcd_box(161,100,200,120,1);
    glcd_box(201,100,220,120,1);

    tmp="TANIMLAMALAR";
    glcd_text57(50,40,tmp,2,1);
    tmp="TAMAMLANDI";
}

```

```

        glcd_text57(60,60,tmp,2,1);
        delay_ms(1000);
    }

void drawPositionFeedbacks()
{
    //WRITE POSITION FEEDBACK
    sprintf(tmp,"d0:%d",getArmPosition(0));
    glcd_box(2,43,49,51,0);
    glcd_text57(3,44,tmp,1,1);
    sprintf(tmp,"d1:%d",getArmPosition(1));
    glcd_box(2,55,49,63,0);
    glcd_text57(3,56,tmp,1,1);
    sprintf(tmp,"d2:%d",getArmPosition(2));
    glcd_box(2,67,49,75,0);
    glcd_text57(3,68,tmp,1,1);
}

void drawDegrees()
{
    output_high(prcss18f);
    //DRAW DEGREE

    //READ DEGREE X
    qX=100;
    while (qX>60 || qX<-60)
    {
        qX=getDegree(0x33,0x32)-3;
    }
    //WRITE DEGREE X
    if (qX<51 && qX>-51)
    {
        glcd_box(2,2,49,10,0);
        sprintf(tmp,"Qx:%d",qX);
        glcd_text57(3,3,tmp,1,1);
        Xsend=(qX+90);
    }
    else
    {
        glcd_box(2,2,49,10,0);
        sprintf(tmp,"Qx:N/A");
        glcd_text57(3,3,tmp,1,1);
        Xsend=255;
    }

    //READ DEGREE Y
    qY=100;
    while (qY>60 || qY<-60)
    {
        qY=getDegree(0x35,0x34);
    }
    //WRITE DEGREE Y
    if (qY<51 && qY>-51)
    {
        glcd_box(2,14,49,22,0);
        sprintf(tmp,"Qy:%d",qY);
    }
}

```

```

        glcd_text57(3,15,tmp,1,1);
        Ysend=(qY+90);
    }
    else
    {
        glcd_box(2,14,49,22,0);
        sprintf(tmp,"Qy:N/A");
        glcd_text57(3,15,tmp,1,1);
        Ysend=255;
    }

    //READ DEGREE Z
    if (qX<8 && qX>-8 && qY<8 && qY>-8)
    {
        // -360 +360 arası olduğu için qqZ int16 olarak tanımlanmıştır.
        // qZ ise fonksiyonlarda kullanılan 8 bitlik versiyonudur +50 -50 arası
        hz=HMC5883L_heading(2);
        qqZ=floor(hz + 68.5);
        qqZ=qqZ%360;
        if (qqZ>180)
        {
            qqZ=(360-qqZ)*-1;
        }
    }
    else
    {
        qqZ=0;
    }
    //WRITE DEGREE Z
    if (qqZ<51 && qqZ>-51)
    {
        qZ=floor(qqZ);
        glcd_box(2,26,49,34,0);
        sprintf(tmp,"Qz:%d",qZ);
        glcd_text57(3,27,tmp,1,1);
        Zsend=(qZ+90);
    }
    else
    {
        glcd_box(2,26,49,34,0);
        sprintf(tmp,"Qz:N/A");
        glcd_text57(3,27,tmp,1,1);
        Zsend=255;
    }

    output_low(prcss18f);
}

void drawDegreesJoystick()
{
    getJoystickPosition();

    //WRITE DEGREE
    glcd_box(2,2,49,10,0);
    sprintf(tmp,"Qx:%d",Xj);
    glcd_text57(3,3,tmp,1,1);

    glcd_box(2,14,49,22,0);
    sprintf(tmp,"Qy:%d",Yj);
    glcd_text57(3,15,tmp,1,1);
}

```

```

glcd_box(2,26,49,34,0);
sprintf(tmp,"Qz:%d",Zj);
glcd_text57(3,27,tmp,1,1);
}

```

```

void callMode()
{
    if (modeNo==0)
    {
        glcd_box(120,118,240,128,0);
        tmp="Mode:Joystick Mode";
        glcd_text57(120,118,tmp,1,1);

        delay_ms(400);
    }
    else if (modeNo==1)
    {
        glcd_box(120,118,240,128,0);
        tmp="Mode:Zero Position";
        glcd_text57(120,118,tmp,1,1);

        delay_ms(400);
    }
    else if (modeNo==2)
    {
        glcd_box(120,118,240,128,0);
        tmp="Mode:Work Space";
        glcd_text57(120,118,tmp,1,1);

        delay_ms(400);
    }
    else if (modeNo==3)
    {
        glcd_box(120,118,240,128,0);
        tmp="Mode:Sensor Mode";
        glcd_text57(120,118,tmp,1,1);

        delay_ms(400);
    }
    else if (modeNo==4)
    {
        glcd_box(120,118,240,128,0);
        tmp="Mode:Parallel Mode";
        glcd_text57(120,118,tmp,1,1);

        delay_ms(400);
    }
    else if (modeNo==5)
    {
        glcd_box(120,118,240,128,0);
        tmp="Mode:Idle Mode";
        glcd_text57(120,118,tmp,1,1);

        delay_ms(400);
    }
}
}

```

```

void initialConf()
{
    //GENERAL ADJUSTMENTS
    setup_adc_ports(NO_ANALOGS|VSS_VDD);
    setup_adc(ADC_OFF);
    setup_psp(PSP_DISABLED);
    //setup_spi(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);

    //PIN SETTINGS
    output_low(prcss18f);

    //VARIABLE SETTINGS
    modeNo=5;

    //TIMER SETTINGS
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);
    set_timer1(tmr1);

    //T6963C.c
    port_b_pullups (TRUE);
    glcd_init(240,128);
    glcd_WriteByte(1, (LCDModeSet|LCDMode_XOR)); // Grafik LCD ekran XOR moda
göre ayarlanıyor
    glcd_WriteByte(1, (LCDDispMode|LCDDisp_TXT|LCDDisp_GRH)); // Grafik LCD
ekran hem metin hem grafik modunda çalışacak şekilde ayarlanıyor

    //ANALOG SETTINGS
    setup_adc(adc_clock_internal);
    setup_adc_ports(AN0_TO_AN5);

    //ADXL345 SETTINGS
    adxl345_init();

    //HMC5883L SETTINGS
    HMC5883L_init();

    //VARIABLES
    data[0]=0;
    data[1]=1;
    data[2]=2;
    data[3]=3;

    //INTERRUPT SETTINGS
    enable_interrupts(int_timer1);
}

void main()
{
    initialConf();
}

```



```

//drawKocaeliLogo();
//drawLoading();
drawGrid();
drawRobot();

glcd_box(120,118,240,128,0);
tmp="Bir mode seciniz";
glcd_text57(120,118,tmp,1,1);

//INTERRUPT PERMISSION
enable_interrupts(global);

while(true)
{
    beginPoint:

    if (input(modeSelect)==false)
    {
        modeNo++;
        modeNo=modeNo%6;

        callMode();
        delay_ms(200);
        goto beginPoint;
    }

    if (modeNo==0)
    {
        disable_interrupts(int_timer1);

        drawDegreesJoystick();
        drawPositionFeedbacks();

        enable_interrupts(int_timer1);
        set_timer1(tmr1);
    }
    else
    {
        disable_interrupts(int_timer1);

        drawDegrees();
        drawPositionFeedbacks();

        enable_interrupts(int_timer1);
        set_timer1(tmr1);
    }

    delay_ms(250);
}

```

ÖZGEÇMİŞ

1986 yılında Söke’de doğmuştur. İlk öğrenimini 1992-1997 yılları arasında Fahrettin Uygun Tüzel İlköğretim Okulu’nda, orta ve lise öğrenimini 1997–2004 yılları arasında Söke Hilmi Fırat Anadolu Lisesi’nde tamamlamıştır. 2004 yılında girdiği Karadeniz Teknik Üniversitesi Elektrik Elektronik Mühendisliği Bölümü’nü 2009 yılında bölüm üçüncülüğü ile bitirmiştir. Yüksek lisans eğitimine Kocaeli Üniversitesi Mekatronik Mühendisliği bölümünde devam etmektedir. Aynı zamanda 2009 yılından buyana Turkcell İletişim Hizmetleri A.Ş.’de Rollout ve Kaynak Yönetimi Mühendisi olarak görev almaktadır.