



**YÖRÜNGE VE HAREKET
KONTROLLÜ MOBİL ROBOT
TASARIMI**

ABİDİN CAN

Yüksek Lisans Tezi

Elektronik ve Haberleşme Mühendisliği

Anabilim Dalı

Danışman: Prof. Dr. Hafız ALİSOY

2019

T.C.

TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

YÖRÜNGE VE HAREKET KONTROLLÜ MOBİL ROBOT TASARIMI

ABİDİN CAN

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANA BİLİM DALI

DANIŞMAN: PROF. DR. HAFIZ ALİSOY

TEKİRDAĞ-2019

Her Hakkı Saklıdır

Prof. Dr. Hafız ALİSOY danışmanlığında, Abidin CAN tarafından hazırlanan “Yörünge ve Hareket Kontrollü Mobil Robot Tasarımı” isimli bu çalışma aşağıdaki jüri tarafından Elektronik ve Haberleşme Mühendisliği Anabilim Dalı’nda Yüksek Lisans tezi olarak oy birliği ile kabul edilmiştir.

Jüri Başkanı: Prof. Dr. Hafız ALİSOY

İmza:

Üye: Doç. Dr. Şuayb Çağrı YENER

İmza:

Üye: Doç. Dr. Reşat MUTLU

İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu adına

Doç. Dr. Bahar UYMAZ

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

YÖRÜNGE VE HAREKET KONTROLLÜ MOBİL ROBOT TASARIMI

Abidin CAN

Tekirdağ Namık Kemal Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Hafız ALİSOY

Günümüzde robotlar hayatımızın her alanında yer almaya başlamıştır. Önümüzdeki yıllarda yapay zekanın gelişmesiyle otonom robot teknolojisi daha da önemli hale gelecektir. Gelişmiş ülkeler robot çalışmalarını çok fazla önemsemekte uluslararası robot yarışmaları düzenlemektedir. Ülkemizin bu teknolojinin içinde olması için öğrencilerimizin küçük yaşlarda robotik ve programlama ile tanışması büyük önem taşımaktadır. Bu amaçla bazı okullarımız robot yarışmaları düzenlemekte, STEM ve MAKER etkinlikleri ile öğrencilerimize robotik tanıtılmaktadır. Ayrıca 2018’de Teknofest proje yarışması düzenlenerek ülke çapında çok büyük bir etkinlik yapılmıştır. Robotik faaliyetlerin çok hızlı gelişmesi dikkate alınarak oluşturulan bu tez çalışmasında mobil robotların yörünge ve hareket kontrolü incelenmiştir. Sırasıyla mobil robotların kinematik ve dinamik sistem modelleri, kontrol sistemleri türleri, geri besleme, kontrol sistemlerinin zaman bölgesinde analizi ve PID kontrol sistemi açıklanmıştır. Tez çalışmasında test amacıyla kullanılan First Robotics Competition robotunun mekanik, elektronik ve yazılım bileşenleri anlatıldıktan sonra robotun elektronik ve yazılım sistemi verilmiştir. Robottaki lift sisteminin hareketi ve robotun düz bir yörüngede gitmesi için PID kontrolör tasarımı gerçekleştirilerek kontrol parametreleri gerçek zamanlı konum ve açı grafikleri çizdirilerek belirlenmiştir. Ayrıca mobil robot kontrol tasarımında dikkat edilecek noktalar açıklanmıştır.

Anahtar Kelimeler: Mobil Robot Bileşenleri, Mobil Robot Tasarımı, PID Kontrolör

2019, 90 sayfa

ABSTRACT

MSc. Thesis

ORBITAL AND MOTION CONTROL OF MOBILE ROBOTS

Abidin CAN

Tekirdag Namık Kemal University
Graduate School of Natural and Applied Sciences
Department of Electronics and Communications Engineering

Supervisor : Prof. Dr. Hafız ALİSOY

Today, robots are started to take place in every aspect of our lives. In the coming years along with the development of artificial intelligence, autonomous robot technology will become even more important. Developed countries are cared about robotics and organizing international robotics competitions. In order for our country to be in this technology, it is very important for our students to meet with robotics and programming at an early age. For this purpose, some of our schools organize robot competitions, and robotics are introduced to our students with STEM and MAKER activities. Furthermore, Teknofest project competition was organized in 2018 and a big event was held throughout the country. In this thesis study, which is formed considering the rapid development of robotic activities, orbital and motion control of mobile robots are examined. The kinematic and dynamic system models of the robots are explained, the types of control systems, feedback, control systems are described in the time frame and the PID controller is explained. In the thesis study, the mechanical, electronic and software components of the First Robotics Competition robot is explained and the robot's electronic and software system is given. The PID controller design is designed to move the elevator system and advance the robot in a straight orbit. Control parameters is determined by real time position and angle graphs. Also, the points to be considered in robot control design is explained.

Keywords: Mobile Robot Components, Mobile Robot Design, PID Controller

2019, 90 pages

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER.....	iii
ÇİZELGE DİZİNİ.....	vi
ŞEKİL DİZİNİ.....	vii
SİMGELER VE KISALTMALAR DİZİNİ.....	ix
ÖNSÖZ	xi
1. GİRİŞ	1
2. KURAMSAL TEMELLER	6
2.1. Mobil Robotun Kinematik Modeli	6
2.2. Mobil Robotun Dinamik Modeli	9
2.3. Hareketin Tekerleğe Aktarım Modeli.....	12
2.4. Dinamik Robot Sisteminin Bilgisayar Modeli	12
3. KONTROL SİSTEMLERİ	13
3.1. Kontrol Sistemi Türleri.....	13
3.2. Geri Besleme	14
3.3. Kontrol Sistemlerinin Zaman Bölgesinde Analizi.....	15
3.4. PID Kontrolör	17
3.5. PID Parametrelerini Ayarlama Metotları.....	20
4. MOBİL ROBOT BİLEŞENLERİ ve ROBOT TASARIMI	23
4.1. Mekanik Bileşenler	23
4.1.1. Robot şasesi	23
4.1.2. Motor dişli kutusu.....	23
4.1.3. Robot lift ve robot kol sistemi	24
4.2. Robotun Elektronik Bileşenleri	25
4.2.1. RoboRIO	26
4.2.2. Güç dağıtım paneli (PDP).....	28
4.2.3. Voltaj regülatör modülü(VRM).....	28
4.2.4. Sigortalar.....	30
4.2.5. CIM motor	30
4.2.6. BAG motor	31
4.2.7. Snow blower motor.....	32
4.2.8. TALON SRX motor sürücüsü	33

4.2.9. SPARK motor sürücüsü.....	34
4.2.10. Access point.....	35
4.2.11. Akü	36
4.2.12. Pnömatik kontrol modülü(PCM)	36
4.2.13. Robot sinyal lambası.....	37
4.2.14. Enkoder.....	37
4.2.15. Sınır anahtarı.....	38
4.2.16. JİROSKOP.....	38
4.2.17. Kamera.....	39
4.3. Yazılım Bileşenleri	39
4.3.1. LabVIEW FRC	39
4.3.2. FRC Update Suite	40
4.3.3. WPI robotik kütüphanesi	40
4.3.4. Visual Studio Code IDE (C ++ \ Java)	40
4.3.5. FRC Radio Configuration Utility	40
4.3.6. FRC roboRIO Imaging Tool.....	40
4.3.7. FRC Driver Station	41
4.3.8. FRC LabVIEW Dashboard.....	41
4.3.9. Smart Dashboard	41
4.3.10. Shuffleboard	42
4.4. Robot Elektronik Sistemi.....	42
4.5. Mobil Robot Sistemi Yazılımı.....	45
4.5.1. Robotun otonom kontrolü.....	45
4.5.2. Robotun sürücü kontrolü	46
4.6. Mobil Robot Lift Sistemin Konum Bilgisini Elde Etme	47
4.6.1. Enkoder ile yön bilgisi elde etme	48
4.6.2. Enkoder ile Hız Bilgisini Elde Etme	49
4.6.3. Enkoderler ile mesafe bilgisi elde etme.....	49
4.6.4. Enkoder Çözünürlüğü	49
4.6.5. Enkoder Çeşitleri	51
4.6.6. Mobil robotik sistemde enkoder ayarları.....	51
5. ROBOTİK SİSTEM YÖRÜNGE VE HAREKET KONTROLÜ DENEYLERİ	54
5.1. Mobil Robotun Lift Sistemini Konumlandırma	54
5.1.1. Lift sistemi P kontrolör tasarımı	56
5.1.2. Lift sistemi PD kontrolör tasarımı	59
5.1.3. Lift sistemi PI kontrolör tasarımı.....	60
5.1.4. Lift sistemi PID kontrolör tasarımı.....	61

5.1.5. Mobil robot lift sisteminde akü gerilimin PID kontrolör üzerine etkileri	66
5.1.6. Ziegler - Nichols metodunun uygulanması.....	67
5.2. Robotu Düz Bir Yörüngede Hareket Ettirmek için PID Kontrolör Tasarımı	69
5.2.1. Robotun düz yörüngede hareketi için PID kontrolör kodları	69
5.2.2. Robotun düz yörüngede hareketine ait PID kontrolör etkileri.....	70
5.2.3. Robotun düz yörüngeden çıkarılmasına PID kontrolör cevabı.....	71
6. SONUÇ ve ÖNERİLER	74
7. KAYNAKÇA.....	76
EK 1 NI roboRIO BLOK DİYAGRAMI	80
EK 2 NI roboRIO GİRİŞ ÇIKIŞ PORTLARI.....	81
EK 3 ACCESS POINT KONFIGÜRASYON AYARLARI	83
EK 4 ROBORIO IMAGE İŞLEMLERİ	84
EK 5 ROBOT KODU	85
EK 6 ROBOT SÜRÜCÜ KONTROLÜ AKIŞ DİYAGRAMI.....	91
ÖZGEÇMİŞ	92

ÇİZELGE DİZİNİ

Sayfa

Çizelge 3.1. PID kontrolör varyasyonları.....	17
Çizelge 3.2. PID parametrelerinin sistem dinamikleri üzerine etkileri	20
Çizelge 3.3. Ziegler-Nichols ayar tablosu (Cao 2019)	22
Çizelge 4.1. RoboRIO PWM çıkışlarına göre robotun hareketleri.....	43
Çizelge 4.2. Mobil robot lift sistemi konum bilgisi hesaplamaları	53
Çizelge 5.1. Lift sistemi PID kontrolör kodu	56
Çizelge 5.2. Ziegler-Nichols metodu ile PID parametrelerinin belirlenmesi	68
Çizelge 5.3: Robotun düz yörüngede gitmesi için PID kontrolör kodu	70



ŞEKİL DİZİNİ

Sayfa

Şekil 1.1. ASIMO robotu (Honda 2019)	1
Şekil 2.1. Mobil robot modelinin yapısı	6
Şekil 2.2. Bir mobil robotun kinematik şeması	7
Şekil 2.3. Şasenin doğrusal ve açısal hızlarıyla kontrol edilen bir mobil robotun kinematik modeli	8
Şekil 2.4. Robotun tekerleklerinin açısal hızını kontrol eden kinematik model.....	9
Şekil 2.5. Robotun dinamik modeli	11
Şekil 2.6. Hareketin robot tekerleğine aktarımın benzetim modeli	12
Şekil 2.7. Bir mobil robotun genel benzetim modelinin şematik gösterimi	12
Şekil 3.1. Açık çevrim kontrol sisteminin öğeleri (Kuo 1999)	13
Şekil 3.2. Kapalı çevrim boşta hız kontrol sistemi (Kuo 1999)	14
Şekil 3.3. Geri beslemeli bir kontrol sisteminin temel blok diyagramı (Kuo 1999)	14
Şekil 3.4. Birim basamak yanıtı	16
Şekil 3.5. Geri beslemeli bir sistemde PID kontrolör blok diyagramı	17
Şekil 3.6. Oransal(P) kontrolör.....	18
Şekil 3.7. PI kontrolör blok diyagramı	19
Şekil 3.8. PD Kontrolör blok diyagramı.....	19
Şekil 3.9. PID kontrolör blok diyagramı	20
Şekil 4.1. Robot şasesi (AndyMark 2019).....	23
Şekil 4.2. Motor dişli kutusu (AndyMark 2019a)	24
Şekil 4.3. Mobil robot sistemi	24
Şekil 4.4. Mobil robot mekanik bileşenleri	25
Şekil 4.5. Roborio denetleyicisi (National Instruments 2019a)	26
Şekil 4.6. Toplam akım-gerilim eğrisi (WPI 2019).....	27
Şekil 4.7. Güç dağıtım paneli (CTR Electronics 2019).....	28
Şekil 4.8. Voltaj regülatör modülü (CTR Electronics 2019a).....	29
Şekil 4.9. VRM modülü giriş-çıkış gerilimi eğrisi (CTR Electronics 2019a).....	29
Şekil 4.10. Sigorta çeşitleri (AndyMark 2019b)	30
Şekil 4.11. PDP sigorta bağlantısı (CTR Electronics 2019).....	30
Şekil 4.12. CIM motor (AndyMark 2019c).....	31
Şekil 4.13. CIM motor tork-hız grafiği (AndyMark 2019c)	31
Şekil 4.14. BAG motor (VEX Robotics 2019).....	32
Şekil 4.15. BAG motor eğrileri (VEX Robotics 2019)	32
Şekil 4.16. Snow Blower motor (AndyMark 2019d)	32
Şekil 4.17. Talon motor sürücü bağlantıları (CTR Electronics 2019b).....	33
Şekil 4.18. Mini CIM motorda PWM - Hız grafiği(CTR Electronics 2019b)	33
Şekil 4.19. Talon motor sürücü PWM kablo bağlantısı (CTR Electronics 2019b).....	34
Şekil 4.20. Talon motor sürücü CAN kablo bağlantıları (CTR Electronics 2019b)	34
Şekil 4.21. SPARK motor sürücü bağlantıları (REV Robotics 2019).....	35
Şekil 4.22. Access Point (AndyMark 2019e)	35
Şekil 4.23. 12V/18Ah Akü (AndyMark 2019f).....	36
Şekil 4.24. Pnömatik kontrol modülü (CTR Electronics 2019c).....	36
Şekil 4.25. Robot sinyal lambası (Anonim 2019)	37
Şekil 4.26. CIMcoder (AndyMark 2019g)	37
Şekil 4.27. Sınır anahtarı (Honeywell 2019)	38
Şekil 4.28. Jiroskop (AndyMark 2019h)	39

Şekil 4.29. Microsoft Lifecam HD-3000 kamera (Microsoft 2019).....	39
Şekil 4.30. FRC Driver Station (WPI 2019c).....	41
Şekil 4.31. Mobil robot elektronik sistemi	44
Şekil 4.32. Robot otonom kontrol akış diyagramı.....	45
Şekil 4.33. Microsoft Xbox kablolu oyun kumandası (Anonim 2019a)	46
Şekil 4.34. Enkoder ile çıkış darbelerinin üretilmesi (Cui 2019).....	48
Şekil 4.35. Şaftın saat yönünün tersinde dönmesi (Cui 2019)	48
Şekil 4.36. Şaftın saat yönünde dönmesi (Cui 2019)	48
Şekil 4.37. Farklı ölçek faktörleri ile darbe sayımı	50
Şekil 4.38. Devir başına gerekli darbe sayısı ve darbe derecesi (Cui 2019a).....	51
Şekil 5.1. Lift sistemi için kontrol sistemi blok diyagramı	54
Şekil 5.2. Kumanda ile lift sisteminin kontrolü	55
Şekil 5.3. P Kontrol konum-zaman eğrisi ($K_p=0,05$).....	57
Şekil 5.4. P kontrol Hata-zaman eğrisi ($K_p=0,05$)	58
Şekil 5.5. P kontrol konum-zaman eğrisi ($K_p=0,25$).....	58
Şekil 5.6. P kontrol konum-zaman eğrisi ($K_p=1$).....	59
Şekil 5.7. PD kontrol konum-zaman eğrisi.....	60
Şekil 5.8. PI kontrol konum-zaman eğrisi	60
Şekil 5.9. P kontrol konum-zaman eğrisi ($K_p=0,5$).....	61
Şekil 5.10. PI kontrol konum-zaman eğrisi ($K_p=0,5$; $K_i=0,5$ ve $K_d=0$)	62
Şekil 5.11. PID kontrol konum-zaman eğrisi($K_p=0,5$; $K_i=0,5$ ve $K_d=0,05$)	62
Şekil 5.12. PID kontrol hata-zaman eğrisi($K_p=0,5$; $K_i=0,5$ ve $K_d=0,05$)	63
Şekil 5.13. PID kontrolör çıkış eğrileri ($K_p=0,5$; $K_i=0,5$ ve $K_d=0,05$) (a) oransal kontrol çıkışı (b) integral kontrol çıkışı (c) türev kontrol çıkışı (d) PID kontrolör çıkışı.....	63
Şekil 5.14. PID kontrol konum-zaman eğrisi (~100s analiz)	64
Şekil 5.15. PID kontrol hata-zaman eğrisi(~100s analiz).....	65
Şekil 5.16. PID kontrolör çıkış eğrileri(~100s analiz) (a) oransal kontrol çıkışı (b) integral kontrol çıkışı (c) türev kontrol çıkışı (d) PID kontrolör çıkışı.....	65
Şekil 5.17. Akü geriliminin lift sistemi üzerine etkileri	66
Şekil 5.18. Akü geriliminin 11,5V seviyesine düşmesi durumunda lift sisteminin konumu ...	66
Şekil 5.19. Akü geriliminin 11V seviyesine düşmesi durumunda lift sisteminin konumu	67
Şekil 5.20. Ziegler-Nichols Metodu ile lift sisteminin konum-zaman eğrisi	68
Şekil 5.21. Yörünge kontrol sisteminin blok diyagramı.....	69
Şekil 5.22. Robotun düz yörüngede hareketine ait PID kontrolör etkileri (a) aç-zaman grafığı (b) hata-zaman grafığı (c) oransal kontrol çıkışı (d) integral kontrol çıkışı (e) türev kontrol çıkışı.....	71
Şekil 5.23. Robotun düz yörüngeden çıkarılmasıyla PID kontrolör cevabı(a) aç-zaman grafığı (b) hata-zaman grafığı (c) oransal kontrol çıkışı (d) integral kontrol çıkışı (e) türev kontrol çıkışı	72

SİMGELER VE KISALTMALAR DİZİNİ

A	:	Akım şiddeti (Amper)
AC	:	Alternatif akım (Alternative current)
Ah	:	Amper saat (Ampere hours)
bps	:	Saniyedeki bit sayısı (Bits-per-second)
C	:	Üretilen pals sayısı (Count)
CAN	:	Kontrol alan ağı veri yolu (Controller Area Network Bus)
CPR	:	Devir başına üretilen ölçüm sayısı (Counts per Revolution)
dBm	:	Decibel miliwatt
DC	:	Doğru akım (Direct Current)
DIO	:	Dijital giriş çıkış pinleri (Digital Input Output)
Dout	:	Türev kontrolör çıkışı (Derivative Output)
D-PAD	:	Yön Düğmesi (Directional Pad)
DPP	:	Darbe başına alınan mesafe (Distance Per Pulse)
DRC	:	DARPA robotik yarışması (DARPA Robotics Challenge)
F	:	Lineer yer değiştirme kuvveti
FIRST	:	For Inspiration and Recognition of Science and Technology
FPGA	:	Alan programlanabilir kapı dizisi (Field Programmable Gate Arrays)
fps	:	Saniyedeki görüntü sayısı (Frame Per Second)
FRC	:	First robot yarışması (First Robotics Competition)
Hz	:	Hertz
I2C	:	I2C haberleşme protokolü (Inter-Integrated Circuit)
Iout	:	İntegral kontrolör çıkışı (Integral Output)
Kd	:	Türevsel kontrolör kazancı (Derivative Controller Gain)
Ki	:	İntegral kontrolör kazancı (Integral Controller Gain)
Kp	:	Oransal kontrolör kazancı (Proportional Controller Gain)
LAN	:	Yerel alan ağı (Local Area Network)
LB	:	Tuş takımı sol yassı düğme (Left bumper)
LT	:	Tuş takımı sol tetik düğmesi (Left trigger)
MB	:	Mega byte
M_0	:	Hareketli robotun açıl momentumu
Nm	:	Newton metre (Newton meter)
PCM	:	Pünomatik kontrol modülü (Pneumatic Control Module)
PDP	:	Güç dağıtım paneli (Power Distribution Panel)
PID	:	Oransal-İntegral-Türevsel kontrolör (Proportional-Integral-Derivative)
PPR	:	Devir başına gerekli darbe sayısı (Pulses per revolution)
Pout	:	Oransal kontrolör çıkışı (Proportional Output)
PWM	:	Pünomatik kontrol modülü (Pneumatic Control Module)
RB	:	Tuş takımı sağ yassı düğme (Right bumper)
RPM	:	Dakikadaki devir sayısı (Revelution Per Minute)
RSL	:	Robot sinyal lambası (Robot Signal Light)
RT	:	Tuş takımı sağ tetik düğmesi (Right trigger)
s	:	Saniye (second)

SF	:	Ölçek faktörü (ScaleFactor)
S/s	:	Saniyedeki örnek sayısı (Samples per second)
SPI	:	Seri çevresel arayüz haberleşmesi ((Serial Peripheral Interface)
USB	:	Evrensel seri veriyolu (Universal Serial Bus)
V	:	Gerilim (Volt)
VRM	:	Voltaj regülatör modülü (Voltage Regulator Module)
W	:	Güç (Watt)
WAN	:	Geniş alan ağı (Wide Area Network)
WPI	:	Worcester Politeknik Enstitüsü (Worcester Polytechnic Institute)
Ω	:	Direnç (Ohm)
ω	:	Açısal Hız
ω_L	:	sol tekerleğin dönüş hızı
ω_R	:	sağ tekerleğin dönüş hızı
$^{\circ}/s$	:	Derece/Saniye
θ	:	Mobil robotun küresel koordinat sistemindeki x eksenine göre hareket yönü

ÖNSÖZ

Bu yüksek lisans çalışmasında mobil robotun bileşenleri açıklanarak otonom ve uzaktan kumanda ile kontrol edilebilen First Robotics Competition robotunun kontrol sistemi incelenmiştir. Robotun yörünge ve hareket kontrolü için PID kontrolör tasarımı yapılarak kontrol etkileri değerlendirilmiştir.

Çalışmam sürecinde her zaman destek veren, yol gösteren ve bilgi birikimini aktaran danışmanım Prof. Dr. Hafız Alisoy'a, First Robotics Competition yarışmasını Türkiye'ye getiren Fikret Yüksel Vakfına, robotun mekanik tasarımını yapan CNC İleri Teknoloji firmasına, ana mentörlüğünü yaptığım robot takımımdaki tüm öğretmen arkadaşlarıma ve öğrencilerime teşekkürü bir borç bilirim.

Son olarak en büyük teşekkür, her zaman yanımda olup destek veren, değerli eşime.

Mayıs 2019

Abidin CAN

1. GİRİŞ

1986'da Honda mühendisleri insan gibi yürüyen robot yapmak için çalışmaya başlamışlardır. İlk modellerde(E1,E2,E3) bir insanın yürüyüşünü örnek alarak robot bacaklar geliştirmiştir. Bir sonraki model (E4, E5, E6) robotlarında kararlı yürüyüş ve merdiven çıkma gibi çalışmalar yapmışlardır. Sonrasında ise dengeli ve işlevsel özellikler için robota baş, gövde ve kol gibi özellikler eklemişlerdir. Honda'nın ilk insansı robotu P1'dir. P2 robotu ise geliştirilmiş yürüme, merdivenlerden çıkıp inme ve kablosuz otomatik hareketler yapma gibi özelliklere sahiptir. Honda robotik çalışmalarının sonunda ASIMO ismini verdiği insansı robotu geliştirmiştir. ASIMO koşabilmekte, eğimli yüzeylerde yürüyebilmekte, merdivenlerden çıkabilmekte, nesneleri bir yerden bir yere koyabilmektedir. ASIMO temel seviyede sesli komutlara cevap verebilmektedir. ASIMO, görüntü işleme özelliği sayesinde insanları yüzlerinden tanıyabilmekte, ortamı haritalandırarak nesnelerin konumlarını kaydedebilmektedir. Honda gelecek planlarında ASIMO ile insanların ihtiyaçlarına cevap vermek, yaşlılara ve engellilere önemli derecede yardımcı olmak vardır. Ayrıca yangına müdahale ve kimyasal tehlike durumlarında temizlik gibi katkılar sağlama amacındadır (Honda 2019).

Honda ASIMO dışında robotik çalışmalarda yapmaktadır. Örneğin zayıf bacak kasları olup yürümekte zorlanan kişiler için tasarladığı cihazda bir motor ileri geri hareket ederek bacakların kaldırılmasına yardımcı olmaktadır. Bu cihaz ile kullanıcın adımları uzatarak daha uzun yol kaydedebilmektedir. Honda ikinci bir deneysel çalışmasında insanların yürürken, merdivenlerden inip çıkarken, çömelme durumunda bacaklarındaki yükü azaltmak ve vücuda destek için bir cihaz daha geliştirmiştir. Bu cihaz ile insanlar daha az güç sarf ederek daha az yorulmaktadır (Honda 2019a).



Şekil 1.1. ASIMO robotu (Honda 2019)

Günümüzde robotik ile çalışmalarda Boston Dynamics firması ön plana çıkmaktadır. Massachusetts Teknoloji Enstitüsü'nün desteği ile hayvanlar gibi koşabilen manevra yapabilen robotlar geliştirilmiştir. Dinamik ve denge prensiplerini yeni teknolojilerle birleştirip kusursuz robotlar yapma amacı vardır. Bu robotlardan Handle lojistik işlemler için geliştirilmiştir ve robot kolu sayesinde paketleme yapabilmektedir. Atlas robotu ise insansı bir robottur ve engebeli arazilerde dengeli bir şekilde yol alabilmektedir. Sarsılması durumunda kendini dengeye getirebilmekte ve düşerse ayağa kalkabilmektedir. WildCat robotu saatte 32 km yol yapabilen dünyanın en hızlı 4 bacaklı robotudur. Robot içerdiği çeşitli sensörler ile yüksek hızlarda dengesini koruyabilmektedir (Boston Dynamics 2019).

AKINROBOTICS firması yerli sermaye ile kurulmuş bir firmadır ve özellikle hizmet sektörü için robotlar yapmaktadır. Bu robotlar insan ergonomisine uygun olmayan monoton işleri yaparak insan yaşamına katkı sağlayabilecektir. ADA isimli robotu yiyecek içecek servisi yapma, yürüme, konuşma, görme, engel tanıma gibi yetenekleri olan yapay zekaya sahip bir robottur (Akın 2016).

Robotik çalışmalar hızla yaygınlaşırken birçok robot yarışması da düzenlenmektedir. Amerika Birleşik Devletleri Savunma Bakanlığı İleri Araştırma Projeleri Ajansı (DARPA) DARPA Robotics Challenge (DRC) yarışması düzenlemiş ve bu yarışmada robotlar ortamdaki el aletlerini kullanarak karmaşık görevleri yerine getirmişlerdir. Bu yarışmanın amacını DARPA robotik gelişimi hızlandırmak ve insanlar için tehlikeli işleri (örneğin 2011'de Japonya Fukushima' daki gibi) robotların yaparak insan yaşamına katkı sağlamak olarak belirlemiştir (DARPA 2019).

FIRST (For Inspiration and Recognition of Science and Technology) bilim ve teknolojileri alanlarına gençlerin ilgisini arttırmak ve teşvik etmek amacıyla 1989 yılında Daen Kamen tarafından kurulmuştur. Kar amacı gütmeyen bu vakıf birçok büyük firma sponsorluğunda farklı yaş grupları için yarışmalar düzenlemektedir. Bu yarışmalarda öğrenciler robot yaparak fen, teknoloji, matematik ve mühendislik becerilerini geliştirmektedir. Ayrıca bir robottan daha fazlasını veren bu yarışmalar öğrencilerin yaşam becerilerini arttırarak kendilerine güven duymalarını amaçlamaktadır. FIRST, yarışmaya katılan öğrencilere birçok üniversitede burs imkanı da verilmektedir. FIRST danışmalarından Dr. Woodie Flowers Gracious Professionalism(Duyarlı profesyonellik) ismini verdiği terim FIRST' ün bir parçasıdır. Duyarlı profesyonellik yüksek kaliteli çalışmayı teşvik etmekte aynı

zamanda rekabet ederken yardımlaşmayı, başkalarına değer vermeyi, topluma ve bireylere saygılı olarak daha anlamlı bir yaşam sürmeyi ifade etmektedir (FIRST 2019).

First Robotics Competition(FRC) FIRST tarafından lise öğrencileri için 1992’den beri düzenlenen bir robot yarışmasıdır. Bu yarışmada öğrenciler gerçek mühendislik deneyimlerini edinmektedir. Nisan 2019 itibari ile yaklaşık 8000 robot takımı bulunmaktadır. Her yıl ocak ayında açıklanan yeni temaya göre robot takımları 6 haftalık süreçte robotlarını tasarlayıp bölgesel yarışmalara katılmaktadır. Bölgesel yarışmalar ABD dışında Kanada, İsrail, Meksika, Avustralya ve Türkiye’ de yapılmaktadır. Bölgesel turnuvalarda başarılı olan takımlar ABD’ de yapılan şampiyonaya katılmaktadır (FIRST 2019a).

Pew Araştırma Merkezi tarafından 2014’ te yayınlanan “2025’ te Dijital Yaşam” anketine katılan uzmanların %48’si, hem mavi hem de beyaz yakalıların işlerini robotların alacağını öngörmüşlerdir. Uzmanların %52’si ise şuan ki birçok işi robotların alacağını kabul etmekte ancak bu teknolojik gelişmelerin insanlara yeni iş fırsatları sağlayacağını ifade etmişlerdir. Üretimde ve teknolojiideki bu gelişmelere insanların uyum sağlamasının önemli olduğu ankette belirtilmiştir (Smith 2014).

Robotik çalışmaları ve kontrol sistemlerini anlamak için endüstri devrimlerinin tarihi gelişimini incelemekte fayda vardır. Endüstri 1.0 olarak adlandırılan ilk sanayi devriminde üretimde makineleşmeye ilk defa geçilmiştir. Bu devrimdeki makinelerin gelişiminde buhar gücü etkili olmuştur. Sanayileşme ile birlikte üretim süreçleri hızlanmış ve insanların yaşamları da farklılaşmıştır. Endüstri 2.0 ile elektrik kullanılarak sanayide seri üretime geçilmiştir. Henry Ford seri üretim yapan araç fabrikası kurmuştur. Bu dönemde ulaşımın gelişmesi ile hammaddeye ulaşım kolaylaşmış ve yeni pazarlara açılımlar başlamıştır. Endüstri 3.0 ile dijital teknolojiler ön plana çıkmış ilk hesap makinesi üretilmiş ve daha sonrasında bilgisayar teknolojileri gelişmiştir. Akıllı telefonlara kadar uzanan bu dönemde bilişim teknoloji firmaları nano boyutta ürünler geliştirmişlerdir. Endüstri 4.0 ise ilk defa Almanya’daki Hannover fuarında tanıtılmıştır. Bilişimin üretim süreçlerini tamamen değiştireceği bu dönem nesnelerin interneti, büyük veri, otonom robotlar, otonom araçlar, 3d yazıcılar, bulut bilişim, siber fiziksel sistemler, akıllı fabrikalar vb. birçok alanda gelişimi tanımlamaktadır. Nesnelerin interneti ile tüm elektronik cihazlar birbiriyle haberleşmeye başlamış ve üretimde robotlar birbiriyle konuşarak tamamen insansız üretim süreçleri gelişmeye başlamıştır (EBSO 2015).

Kontrol sistemleri günlük yaşamdan fabrikalara kadar her alanda kullanımı vardır. Her gün yaşadığımız olaylarda kontrol sistemlerinin etkileri vardır. Fabrikalarda makineler otomatik kontrol edilerek üretimler çok hızlı bir şekilde gerçekleştirilmektedir. Önümüzdeki yıllarda robotik çalışmaların daha da artmasıyla kontrol sistemlerin önemi daha da artacaktır (EBSO 2015).

Robotik bilimi çok disiplinli bir yapıya sahiptir. Bilgisayar bilimleri, yazılım geliştirme, elektrik, elektronik, matematik, makine, fizik, kimya, yapay zeka, insan robot ilişkisi ve insan bilgisayar ilişkisine kadar birçok konu robotik bilimle ilgilidir. Moskova Stratejik Girişimler Ajansının öngörüsüne göre önümüzdeki 15-20 yıl boyunca robotik hızla gelişerek emeğe dayalı işleri yapacak duruma gelecektir. Bunun sonucunda robot programcıları, tasarımcılar, çok fonksiyonlu robotik sistem operatörleri, parça üretimi için kompozit malzeme uzmanları, robotik cihazların bakım onarımı ve montajı uzmanları gibi yeni meslekler ortaya çıkacaktır. Robotik alanında etkin ve motive olmuş profesyoneller için robotik ve mekatronik alanında yeni bir eğitim kavramı gereklidir. Şu anda robotik eğitimi tüm dünyada başlangıç aşamasındadır. Çok yakında küresel bir robotik işgücü piyasası ortaya çıkacaktır. 2011’de Rusya hükümeti tarafından yayınlanan kararnameye göre eğitimin en acil görevinin robotik konusunda gelişmeleri şekillendirme, sürekli eğitim, kişisel gelişim, eleştirel düşünme ve yabancı dil olduğu belirtilmiştir. Rusya, lisans ve yüksek lisans müfredatına robotik dersleri almıştır. Rusya’da robotiğin kökleri 1960’lara dayanmaktadır. Bu yıllarda endüstriyel robotlar geliştirilmiştir. 1972’de robotik araştırmaları devlet politikasının bir parçası olmuştur. Rusya’da 2002’den beri robot yarışmaları düzenlenmektedir. Bu yarışmalara 2014 yılında 10000 öğrenci katılmıştır. Robotik alanda uzmanlara gelecekte çok fazla ihtiyaç olacaktır. Robotik materyallerin çok sık yenilenmesi sebebi ile robotik yeniliklere ayak uydurmak önemlidir. Rusya’da lisans öğrencileri için robotiğe giriş, robotiğin temelleri, kontrol sistemleri ve bilimsel araştırma yöntemleri dersleri verilmektedir. Rusya’daki robot eğitim müfredatının amacı dünyadaki robot endüstrisinin gelişimine katkı sağlamak robot eğitiminde çoklu disiplinler arası yaklaşımı sağlamaktır (Tsoy ve ark. 2017).

Kontrol sistemi tasarımında tasarım süreci performans ihtiyaçlarını açıklayarak başlar. Bunun için sistemin birim basamak yanıtından yararlanılır. Birim basamak yanıtındaki özellikleri kullanarak en kötü senaryoda iyi cevap verecek sistem tasarlamakta fayda vardır. Sistemin tepkisinin bazı değişkenlerle değişmemesi sistemin kararlılığını ifade eder. PID kontrolör endüstride en yaygın kullanılan kontrol algoritmasıdır ve kontrol sistemlerinde

evrensel olarak kabul görmüştür. PID kontrol işleminin popüler olmasının nedeni çok farklı çalışma koşullarında iyi performans vermesi ve kullanımının basit olmasıdır. PID arkasındaki temel düşünce bir sensörle yapılan geri beslemeyi kullanarak oransal, integral ve türev yanıtlarını topladıktan sonra çıkışı vermektir (National Instruments 2019).

PID kontrolör parametreleri temel olarak iki şekilde incelenebilir. İlk şekilde kontrolör parametreleri sabit olarak ayarlanır. Bu durumda kontrol parametreleri sabit olduğu için sistem ihtiyaçlarının değişmesi sonucunda bu kontrolör yapısı sistemin ihtiyacına cevap veremez. İkinci şekilde ise kontrolör parametreleri sistemin ihtiyacına göre değişecek şekilde ayarlanarak sistemin daha etkin çalışması sağlanabilir (Dalcı 2004).

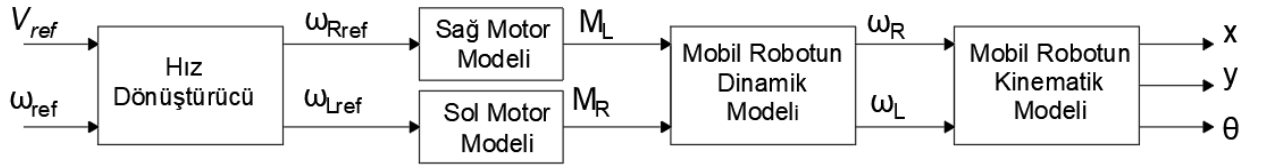
PID kontrolör parametrelerini belirlemek için Ziegler-Nichols ve CHR (Chien-Hrones-Reswick) ayarlama metotları yaygın olarak kullanılmasına rağmen doğrusal olmayan kontrol sistemleri için bu yöntemler çözüm oluşturamamaktadır (Aström ve Höglund 1995).

Bulanık mantık son yıllarda kontrol sistemlerinde kullanımı artmıştır. Zhao, Tomizuka ve Isaka PID kazançlarını bulanık mantıkla belirleyen bir çalışma yapmışlardır. Yapılan bu kontrolörün Ziegler-Nichols ve Kitamori'nin PID kontrolör çalışmalarından daha iyi sonuçlar verdiğini göstermişlerdir (Zhao ve ark. 1993).

Bu tez çalışması önümüzdeki yıllarda daha da yaygınlaşacak mobil robot teknolojileri göz önüne alınarak mobil robot sistemlerinin yörünge ve hareket kontrolleri incelenmiştir. PID kontrolör yapılarının lift sisteminin hareketi ve robotun düz yörüngede gitmesinde etkileri gerçek zamanlı test edilmiştir.

2. KURAMSAL TEMELLER

Bu bölümde, diferansiyel aktarmalı iki tekerlekli mobil bir robotun kinematik ve dinamik benzetim modelleri geliştirilmiştir. Bu modelleri kullanarak, farklı yörüngeler boyunca gerçekleştirilen hareketler test edilmiştir. Holonomik olmayan bir kontrol nesnesi olarak geliştirilmiş bir robot modeli şematik olarak Şekil 2.1’de gösterilmiştir (Курганов 2016).



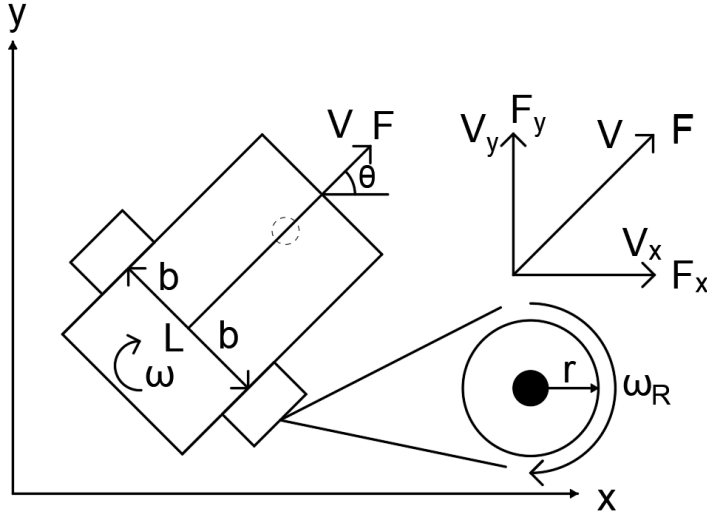
Şekil 2.1. Mobil robot modelinin yapısı

2.1. Mobil Robotun Kinematik Modeli

Hareketli bir nesnenin hareket hızı üzerine belirli sınırlamalar konulması fakat nesnenin pozisyonu ile ilgili herhangi bir kısıtlama olmaması durumlarında, holonomik olmayan mekanik sistemler kullanılır. Bu durumla ilgili en açık örnek, aracın otoparkta ona ayrılan park alanına yaklaşarak paralel park etmesi düşünülebilir. Park yerine paralel yaklaşırken, araç park alanına yana doğru giremez, ancak yine de, aracı ileri geri hareket ettirerek ve tekerlekleri döndürerek, istenen noktaya ulaşmak mümkündür. Basitçe ifade etmek gerekirse, aracın lokalize hareketi sınırlı iken, onun global hareketi sınırsızdır.

Şekil 2.2’de diferansiyel aktarmalı holonom olmayan iki tekerlekli bir robot gösterilmiştir. Bu robotun hareket hızının, kayma olmaması koşuluyla tekerleklerin hareket yönü ile sınırlı olduğu açıktır. Ayrıca bu durumda robotun yana hareket edemeyeceği açıktır. Bununla birlikte, tekerleklerini zıt yönlerde çevirerek, robot, yerinde dönebilir ve böylece gereken konuma hareket etmesine olanak sağlanır. Bu durum, robotun global pozisyonunun sınırsız olduğunu göstermektedir.

Şekil 2.2’ye göre x ve y , robotun ekseninin merkezinin global koordinat sistemine göre konumunu, θ mobil robotun küresel koordinat sistemindeki x eksenine göre hareket yönünü, r robotun tekerleklerinin yarıçapını ve $2b$ (L) tekerlekler arasındaki mesafeyi ifade etmektedir. Son olarak, v ve ω sırasıyla doğrusal ve açısal hızlar olup, ω_L sol tekerleğin dönüş hızı, ω_R ise sağ tekerleğin dönüş hızını ifade etmektedirler (Курганов 2016).



Şekil 2.2. Bir mobil robotun kinematik şeması

Küresel koordinat sistemindeki bu model, iki tekerlekli bir mobil robotun en basit ve minimize temsillerinden biridir. Bu modelde, robot aşağıda tanımlanmış üç serbestlik derecesine sahip bir vektörle temsil edilir.

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (2.1)$$

Yukarıdaki şekilden de anlaşılacağı gibi, bu robot türü üç konum (pozisyonel) serbestlik derecesine sahip olacaktır. Bununla birlikte, mobil bir robot, holonomik olmayan bir kısıtlama nedeniyle anında sadece iki yönde hareket edebilir. Bu sınırlama, robotun eksenine yönüne paralel bir hıza sahip olamaması nedeniyle ortaya çıkmaktadır. Böylece robot, tekerleklerin hareket yönü boyunca V hızında hareket edebilir ve açısal bir ω hızıyla dönebilir. Hareket sırasında kayma olmaksızın, holonomik olmayan bir robot için aşağıdaki matematiksel model oluşturulabilir:

$$\begin{cases} \dot{x} = V \cdot \cos\theta \\ \dot{y} = V \cdot \sin\theta \end{cases} \quad (2.2)$$

Bu ifadelerden hareketle,

$$\frac{\dot{x}}{\cos\theta} = \frac{\dot{y}}{\sin\theta} \quad (2.3)$$

$$\dot{x} \sin\theta = \dot{y} \cos\theta \quad (2.4)$$

denklemlerini elde ederiz.

Dolayısıyla, holonomik olmayan bir kayma sistemi aşağıdaki fonksiyonel sınırlamaya sahiptir:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \quad (2.5)$$

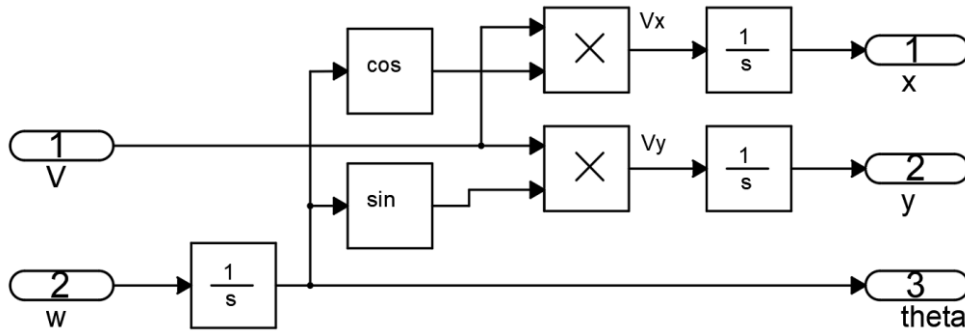
Ek olarak, tekerlek sürtünmesi veya kayma olasılığı gibi dış kuvvetler burada dikkate alınmamıştır. Bu nedenle, robotun yüzeyde kaydığı varsayılabilir.

$$\dot{x}\cos\theta + \dot{y}\sin\theta = v \quad (2.6)$$

Yukarıdaki açıklamalara istinaden, robotun kinematik modeli aşağıdaki gibi olacaktır.

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.7)$$

Bu matematiksel modelin Matlab ortamındaki benzetim devresi Şekil 2.3’de verilmiştir.



Şekil 2.3. Şasenin doğrusal ve açısal hızlarıyla kontrol edilen bir mobil robotun kinematik modeli

Maalesef, robotun V doğrusal ve ω açısal hızları her zaman ölçüm için uygun değildir. Doğrudan tekerleklere monte edilmiş farklı sensörler kullanarak her bir tekerleğin dönme hızını (ω_L ve ω_R) ölçmek çok daha kolaydır ve her bir tekerleğin doğrusal ve açısal hızları ve hızları arasındaki ilişki aşağıdaki gibi ifade edilebilir.

$$v = \frac{\omega_L + \omega_R}{2} r \quad (2.8)$$

$$\omega = \frac{\omega_R - \omega_L}{L} r \quad (2.9)$$

Temel dönüşümlerle tekerleklerin açısal dönme hızlarının mobil robotun doğrusal ve açısal hızlarına bağımlılığı aşağıdaki biçimde belirlenebilir.

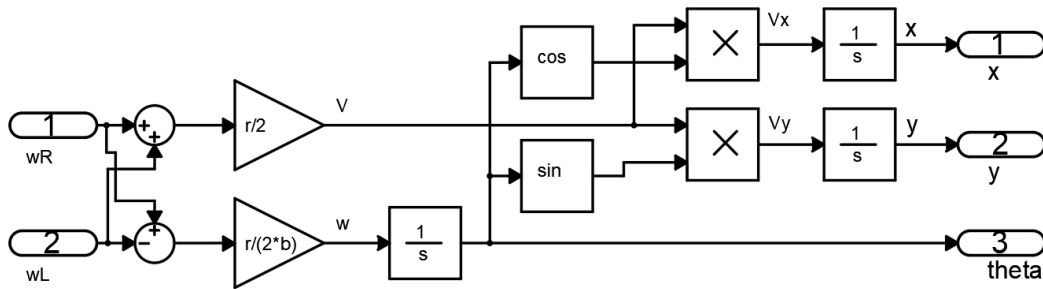
$$\omega_R = \frac{1}{r} \left(v + \frac{\omega L}{2} \right) \quad (2.10)$$

$$\omega_L = \frac{1}{r} \left(v - \frac{\omega L}{2} \right) \quad (2.11)$$

Doğrusal ve açısal hızlar için belirlenmiş ifadeler yukarıdaki denklemde yerine yazılarak, aşağıdaki kinematik matematik modeli elde ederiz:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} r \cos \theta & \frac{1}{2} r \cos \theta \\ \frac{1}{2} r \sin \theta & \frac{1}{2} r \sin \theta \\ -\frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \quad (2.12)$$

Elde edilen kinematik modele dayanarak, iki tekerlekli bir robotun hareket pozisyonunun ve yönünün sadece sol ve sağ tekerleklerin dönme hızına bağlı olduğu sonucuna varabiliriz. Bu modelin MatLab ortamında benzetimi Şekil 2.4’de verilmiştir.



Şekil 2.4. Robotun tekerleklerinin açısal hızını kontrol eden kinematik model

2.2. Mobil Robotun Dinamik Modeli

Kinematik modelin giriş ve çıkış parametreleri, sistem dinamiği gibi önemli bir parametreyi tamamen ihmal eder. Bu nedenle, daha gelişmiş bir tasarım için dinamik bir model elde etmek gerekir (Курганов 2016).

Dinamik bir model oluşturmak için iki yöntem vardır:

a) Sistemi iş ve enerji olarak tanımlayan Lagrange denklemlerini kullanmak. Bu yöntem daha sistematiktir ve kullanılmayan kuvvetleri otomatik olarak ortadan kaldırır.

b) Sistem üzerine etki eden tüm kuvvetler ve momentler açısından sistemi tanımlayan Newton-Euler denklemlerinin kullanılması. Bu yöntem, Newton'un ikinci yasasının doğrudan bir yorumudur.

Varsayalım ki diferansiyel aktarmalı mobil bir robotun kütlesi m ve orta dikey eksen etrafındaki eylemsizlik momenti ise J olsun. Bu durumda, Lagrange denklemlerini kullanarak aşağıdaki matematik modeli elde ederiz.

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} -m.V_y\omega \\ m.V_x\omega \\ 0 \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ M_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F \\ M_\theta \end{bmatrix} \quad (2.13)$$

(2.13) ifadesinde, F ve M_θ sırasıyla lineer yer değiştirme kuvvetini ve hareketli robotun açısal momentumunu temsil eder.

Robotun dönüşleri sırasında, robot üzerine etki eden Coriolis kuvveti ve merkezcil ivme faktörlerinin dikkate alınmasıyla oluşturulan matematik model aşağıdaki gibi olacaktır.

$$m\ddot{x} - m.V_y\omega = F_x \quad (2.14)$$

$$m\ddot{y} + m.V_x\omega = F_y \quad (2.15)$$

$$J \frac{d\omega}{dt} = M_\theta \quad (2.16)$$

Burada; $F_x = F\cos\theta, F_y = F\sin\theta, V_x = V\cos\theta, V_y = V\sin\theta, V = \sqrt{V_x^2 + V_y^2}, M_\theta = (F_R - F_L)$ (2.17)

Doğrusal hız, açısal hız ve genelleştirilmiş hız arasındaki ilişki şöyle ifade edilecektir.

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} \quad (2.18)$$

$$\omega = \dot{\theta} \quad (2.19)$$

Yukarıda elde edilen model daha bilinen bir biçimde ifade edilecek olursa, o halde:

$$\dot{x} = v\cos\theta \quad (2.20)$$

$$\dot{y} = v\sin\theta \quad (2.21)$$

$$\dot{\theta} = \omega \quad (2.22)$$

$$\dot{v} = \frac{1}{m}F - F_c \quad (2.23)$$

$$\dot{\omega} = \frac{1}{J}(M_\theta - M_c) \quad (2.24)$$

Bu modelde, rotasyon sırasında robota etki eden Coriolis kuvvetini hesaba kattığımız için yukarıdaki denklemi aşağıdaki biçimde ifade ederiz:

$$\ddot{x} = \frac{1}{m}(F_x + m.V_y\omega) \quad (2.25)$$

$$\ddot{y} = \frac{1}{m}(F_y - m.V_x\omega) \quad (2.26)$$

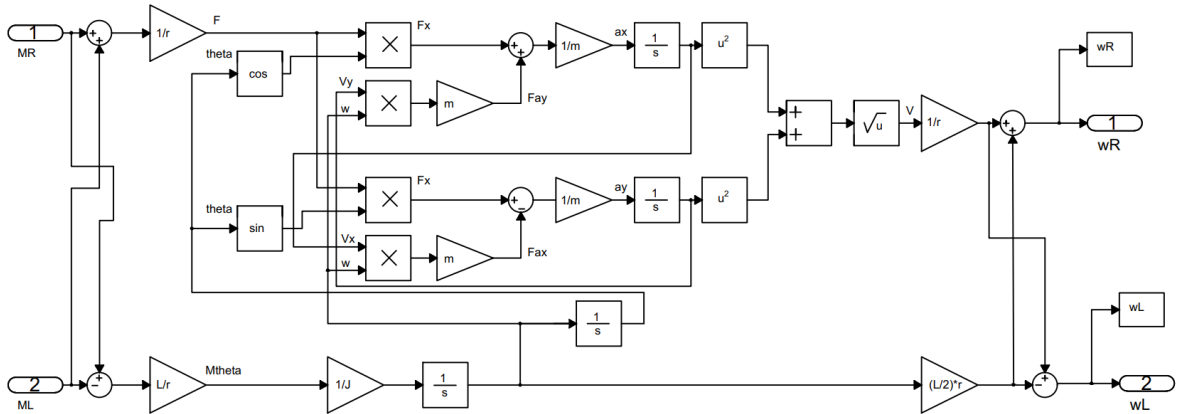
$$\ddot{\theta} = \frac{1}{J}(M_\theta - M_c) \quad (2.27)$$

Kinematik modelle benzerlik yaparak, her bir tekerleğin torkuyla ilgili olan açısıl tork ve hareket yörüngesi doğrultusunda etki eden kuvvet aşağıdaki denklemlerle ifade edilecektir:

$$F = \frac{1}{r}(M_R + M_L) \quad (2.28)$$

$$M_\theta = L(F_R - F_L) = \frac{L(M_R - M_L)}{r} \quad (2.29)$$

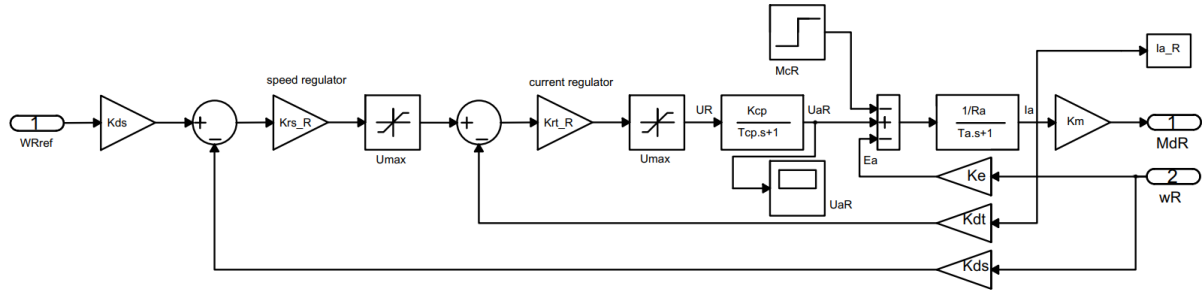
Burada dikkat edilmesi gereken husus, sürücülerin kendi iç dinamiklerine sahip olduğunu ve anında bir kontrol komutu uygulayamayacaklarını anlamak önemlidir. Elde edilen matematiksel modelin Matlab ortamındaki benzetim devresi Şekil 2.5’de verilmiştir.



Şekil 2.5. Robotun dinamik modeli

2.3. Hareketin Tekerleğe Aktarım Modeli

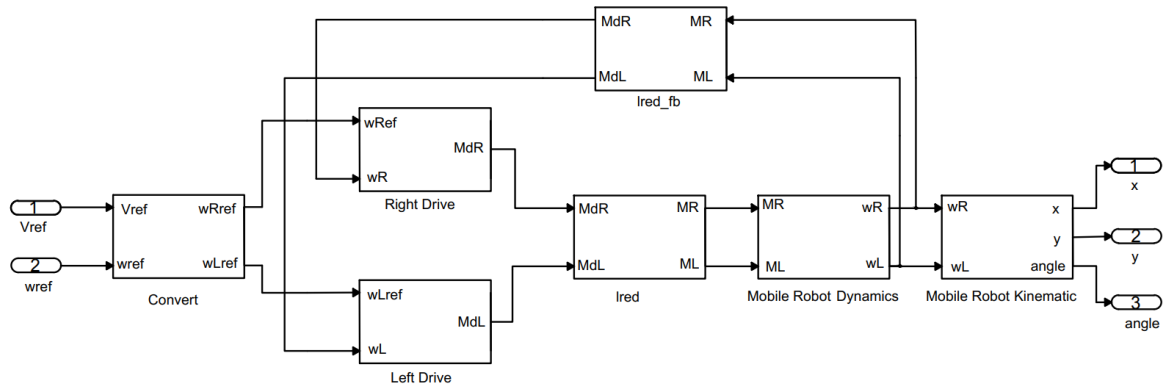
DC motorlar robotik cihazlarda yaygın olarak kullanılır ve mobil robotikte kullanılan ana aktarım elemanlarından biridir. Bu nedenle, sistemin daha ayrıntılı bir dinamik modelini oluşturmak için bir hareket aktarım modelini analiz etmek gerekir. Tekerleğe hareket aktarım modelinin blok şeması Şekil 2.6’da verilmiştir(Kypranov 2016).



Şekil 2.6. Hareketin robot tekerleğine aktarımın benzetim modeli

2.4. Dinamik Robot Sisteminin Bilgisayar Modeli

Bir mobil robot modelinin ana elemanlarının makro blokları oluşturulduktan sonra, mobil robotun genel benzetim modelini oluşturabiliriz(Kypranov 2016). Bu durum Şekil 2.7’de verilmiştir.



Şekil 2.7. Bir mobil robotun genel benzetim modelinin şematik gösterimi

3. KONTROL SİSTEMLERİ

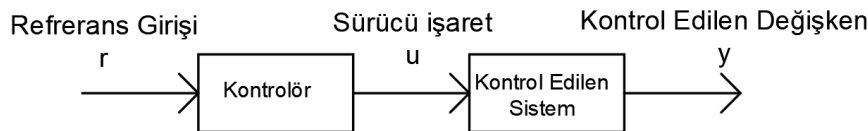
Temel anlamda kontrol sisteminin amacı giriş verilerini kullanarak çıkış verilerini planlanan biçimde ayarlamaktır (Kuo 1999). PID kontrol parametrelerini tanımlamadan önce kontrol sistemi türlerini açıklamada fayda vardır.

3.1. Kontrol Sistemi Türleri

Kontrol sistemleri açık ve kapalı çevrim olarak iki türlü incelenebilir:

a) Açık çevrim kontrol sistemi

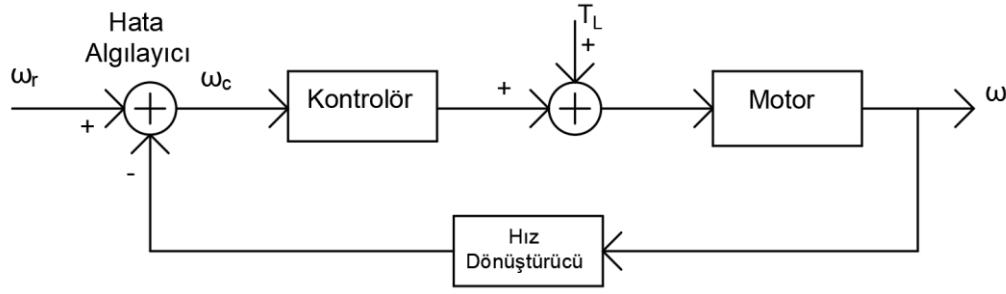
Açık çevrim sistemin temel bileşenleri Şekil 3.1’de gösterilmiştir. Kontrol edilen sistem önceden belirlenen kontrolör tarafından yönetilir. Kontrolör referans işarete(r) bağlı olarak kontrol işareti(u) üretir. Kontrolör açık çevrim sistemde yükseltici, filtre, mikrodenetleyici veya başka bir denetim elemanı olabilir. Açık çevrim sistemde çıkış(y) girişi etkilemez. Çıkış girişe bağlı olarak değişir. Örneğin basit bir elektrik devresinde anahtarla motor kontrol edilsin. Anahtar motoru açıp kapatır, ancak sistemde bir geri besleme olmadığı için yük altında motorun hızı azalacaktır. Açık çevrim sistem hassasiyet istenmeyen durumlarda kullanılabilir. Maliyeti düşük bir sistemdir (Kuo 1999).



Şekil 3.1. Açık çevrim kontrol sisteminin öğeleri (Kuo 1999)

b) Kapalı çevrim kontrol sistemi

Kapalı çevrimli sistemlerde sistemin çıkışından girişine geri besleme yapılır. Bu sistemde çıkış sadece girişe bağlı değildir. Çıkıştan elde edilen bilgiye göre giriş kontrol edilir. Şekil 3.2’de boştaki bir motorun kapalı çevrim kontrolü gösterilmiştir. Boşta motor referans olarak belirlenen hızda (ω_r) dönmelidir ancak T_L yük momenti sebebi ile referans hız ile çıkış hızı arasında fark oluşacaktır. Hız dönüştürücü geri besleme yaparak çıkış hızı ile referansın karşılaştırılmasını sağlar. Çıkış ile giriş hızı arasında oluşan fark kontrolör tarafından değerlendirilerek işaret üretilir. Motor çıkışı kontrolör tarafından üretilen sinyal ile referans değere getirilir. (Kuo 1999)

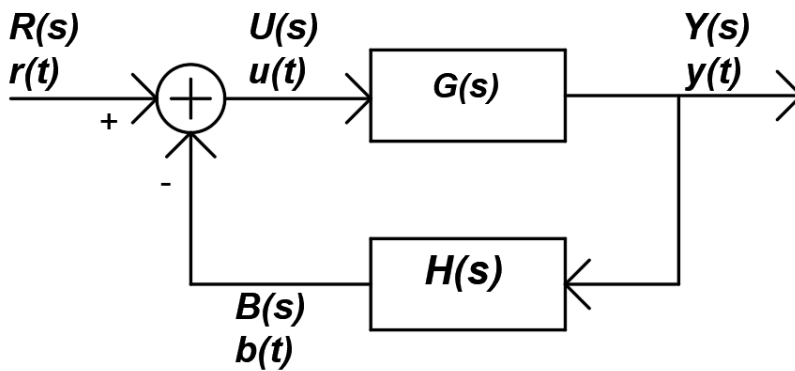


Şekil 3.2. Kapalı çevrim boşta hız kontrol sistemi (Kuo 1999)

3.2. Geri Besleme

Çıkış sinyalinin girişe pozitif veya negatif etkilemesine geri besleme denir. Geri beslemeli sistemlerde çıkış sinyali giriş sinyalini arttıracak şekilde uygulanırsa pozitif geri besleme yapılmış olur. Bu sistemlerde çıkış sürekli girişi arttırdığı için sistemi kararsızlığa götürecektir. Kontrol sistemlerinde negatif geri besleme kullanılarak referans bilgisi ile giriş bilgileri arasındaki fark azaltılır. Bu sistemlerde ayarlanan referans noktasının dışına çıkılırsa sistem buna tepki vererek çıkışı ayarlanan noktaya getirmeye çalışır (Kuo 1999).

Geri beslemeli bir sistemin temel blok diyagramı Şekil 3.3’de verilmiştir. Blok diyagramları ile sistemler basitçe gösterilerek modellenebilir. Transfer fonksiyonu ise doğrusal zamanla değişmeyen sistemlere uygulanır. Transfer fonksiyonu başlangıç durumları sıfır alınarak çıkış değişkeni ve giriş değişkenine laplace dönüşümü yapıldıktan sonra oranlanması ile elde edilir (Kuo 1999).



Şekil 3.3. Geri beslemeli bir kontrol sisteminin temel blok diyagramı (Kuo 1999)

Şekil 3.3’de verilen geri beslemeli bir sistemin transfer fonksiyonu aşağıdaki gibi elde edilebilir (Kuo 1999).

$$Y(s) = G(s).U(s) \quad (3.1)$$

$$B(s) = H(s).Y(s) \quad (3.2)$$

$$U(s) = R(s) - B(s) \quad (3.3)$$

$$Y(s) = G(s).U(s) = G(s)[R(s) - H(s).Y(s)] = G(s).R(s) - G(s).H(s).Y(s) \quad (3.4)$$

$$Y(s) + G(s).H(s).Y(s) = G(s).R(s) \quad (3.5)$$

$$Y(s)[1 + G(s).H(s)] = G(s).R(s) \quad (3.6)$$

$$M(s) = \frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s).H(s)} \quad (3.7)$$

Geri beslemenin sistemin toplam kazancına etkisi G kazancını yükseltme veya azaltma şeklinde olabilir. Geri besleme kararsız bir sistemi kararlı duruma getirebilir ancak yanlış yapılması durumunda kararsızlığa sebep olabilir. Geri besleme çeşitli parametre değişimlerine karşı sistemin etkilenmesini önler ancak giriş verilerine karşı duyarlıdır. Geri besleme ile ayrıca sisteme etki eden dış etkiler ve gürültülerin etkisi indirilebilir (Kuo 1999).

3.3. Kontrol Sistemlerinin Zaman Bölgesinde Analizi

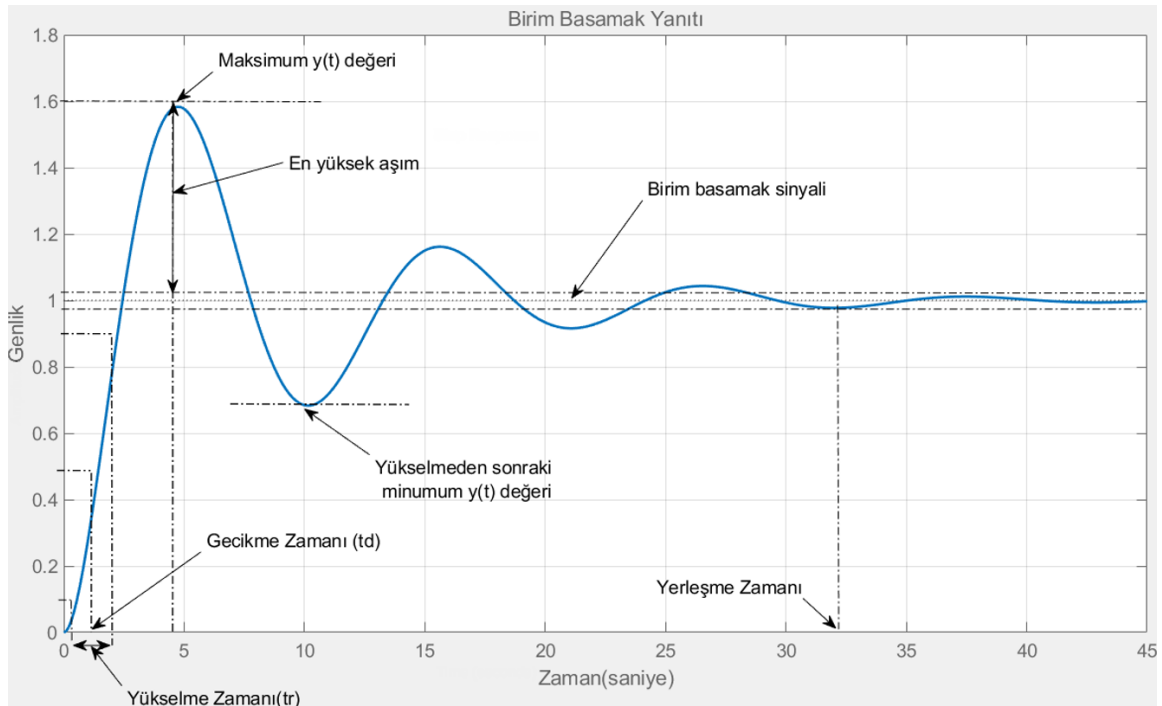
Kontrol sisteminin çıkışı $y(t)$ ile ifade edilirse sistem yanıtı $y(t) = y_t(t) + y_{ss}(t)$ olarak ifade edilir. Burada $y_t(t)$ geçici hal yanıtı ve $y_{ss}(t)$ ise kalıcı hal yanıtıdır. Geçici hal yanıtı zamanla sıfıra giden kısmı ($\lim_{t \rightarrow \infty} y_t(t) = 0$) ifade ederken, sürekli hal yanıtı ise geçici hal bittikten sonraki kısmı ifade eder. Sistemin geçici davranışı kontrol işlemin başlangıç koşullarında nasıl davranacağını gösterdiği için bilinçli olarak kontrol edilmesi çok önemlidir. Geçici hal bittikten sonra sistem çıkışının referansla aynı değerde olmaması durumunda kalıcı hal hatası oluşacaktır. Kalıcı hal yanıtı sistemin doğruluğu için ölçüttür. İyi bir sistem tasarımı için geçici ve kalıcı hal durumları dikkate alınması gerekir (Kuo 1999).

Şekil 3.3'deki sistemde $H(s) = 1$ yapılarak birim geri beslemeli bir sistem oluşturulduğunda hata, referans işaret $r(t)$ ile sistem çıkışı $y(t)$ arasındaki farktır. $e(t) = r(t) - y(t)$ olarak ifade edilir. Kontrol sistemlerinde genellikle $y(t)$ ile $r(t)$ işaretlerinin aynı olması yani hatanın sıfıra indirgenmesi istenir (Kuo 1999).

Kalıcı hal hatası çıkış sinyali ile referans sinyali arasında kalıcı bir hata olması durumunu ifade eder. Kalıcı hal hatası $e_{ss} = \lim_{s \rightarrow \infty} e(t)$ ile ifade edilir. Sürtünme, malzemelerin aşınması, sistemdeki diğer etkiler(gürültü vb.) dolayı hatayı tamamen sıfırlamak zor olabilir. Sistemin kararlı çalışması ve doğruluğu nihai amaç olduğu için kalıcı hal hatasının düşürülerek belli bir sınırdan tutulması gerekir (Kuo 1999).

Geçici hal hatası ise zaman yanıtının zamanla sifıra gitmesidir. Doğrusal bir sistemde geçici hal durumunun genliği ve süresi belli sınırdan tutulmalıdır. Sistemin geçici hal yanıtı için sisteme birim basamak sinyali uygulanarak birim basamak yanıtına bakılarak değerlendirilir. Örnek bir sistemin birim basamak yanıtı Şekil 3.4’de gösterilmiştir (Kuo 1999).

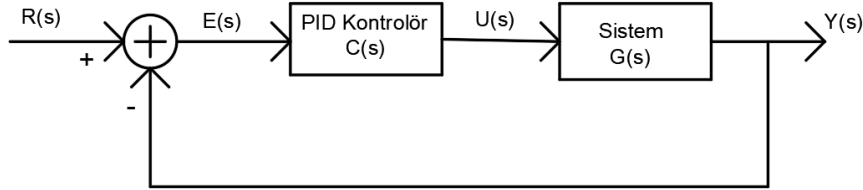
Birim basamak yanıtında en yüksek aşım birim basamak yanıtındaki en yüksek değer ile kalıcı hal değeri arasındaki farka bakılarak belirlenir. Genellikle aşımın yüzdesi olarak ifade edilir. Sistemin kararlılığının sorgulamasında ölçütür. Kararlı bir sistemde aşımın çok yüksek olmaması istenir. En yüksek aşım genellikle ilk aşımda olabileceği gibi negatif durumlarda da en yüksek aşımlar oluşabilir. Yükselme zamanı birim basamak yanıtının %10’dan %90’a ulaşana kadar geçen süreyi ifade eder. Gecikme zamanı birim basamak yanıtının %50’ye ulaşana kadar geçen süreyi ifade eder. Yerleşme zamanı ise hatanın belli bir band(%1,%2,%5) içinde kaldığı süreyi ifade eder (Kuo 1999).



Şekil 3.4. Birim basamak yanıtı

3.4. PID Kontrolör

PID kontrolör birçok kontrol sistemine kolay uygulanabildiği için çok yaygın kullanımı vardır. PID kontrolör oransal (P=Proportional), integral (I =Integral) ve türevsel(D=Derivative) olmak üzere 3 kısımdan oluşur. Basit geri beslemeli bir sistemin blok diyagramı Şekil 3.5’de gösterilmiştir. Çizelge 3.1’de PID kontrolör varyasyonları verilmiştir. (Copeland 2008)



Şekil 3.5. Geri beslemeli bir sistemde PID kontrolör blok diyagramı

Blok diyagrama göre;

$$Y(s) = U(s).G(s) \quad (3.8)$$

$$U(s) = E(s).C(s) \quad (3.9)$$

$$E(s) = R(s) - Y(s) \quad (3.10)$$

formülleri elde edilir.

Çizelge 3.1. PID kontrolör varyasyonları

Kontrolör Tipi	C(s)
P (Oransal)	K_p
I (İntegral)	$\frac{K_i}{s}$
D (Türev)	$K_d s$
PI (Oransal + İntegral)	$K_p + \frac{K_i}{s}$
PD (Oransal + Türev)	$K_p + K_d s$
PID (Oransal + İntegral + Türev)	$K_p + \frac{K_i}{s} + K_d s = K_p \left(1 + \frac{1}{T_i s} + T_d s\right)$

Burada;

K_p :
Oransal kazanç

K_i :
İntegral kazanç

K_d : Türevsel kazanç

T_i : İntegral zaman Sabiti

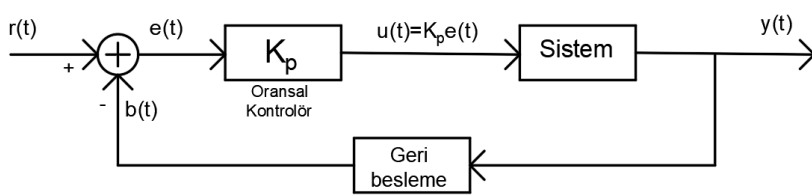
T_d : Türevsel zaman sabiti

olarak adlandırılır.

Oransal(P) kontrolörün temel işlevi sistemin giriş ve çıkışını karşılaştırarak üretilen hataya bağlı kazanç sağlamaktır (Şekil 3.6). Oransal kontrol kazancı K_p artırıldığında sistem daha hızlı tepki verir. Oransal kontrolde hata miktarı azaldıkça sistemin tepkisi azalır. Çok küçük hata değerlerinde oransal kontrol sistemi kontrol edemez ve kalıcı hal hatası meydana gelir. Oransal kontrolör yükselme zamanını ve kalıcı hal hatasını düşürürken aşımı arttırmaktadır. Oransal kontrolör çıkışı;

$$u(t) = K_p e(t) \quad (3.11)$$

olarak elde edilir.



Şekil 3.6. Oransal(P) kontrolör

PI kontrolde integral işlemi ile hata sinyalinin toplamı alınarak kararlı hal hatası iyileştirilir (Şekil 3.7). İntegral işlemi ile aşım ve yerleşme zamanı yükselirken, yükselme zamanını düşer aynı zamanda kalıcı hal hatası da giderilir. PI kontrolör çıkışı;

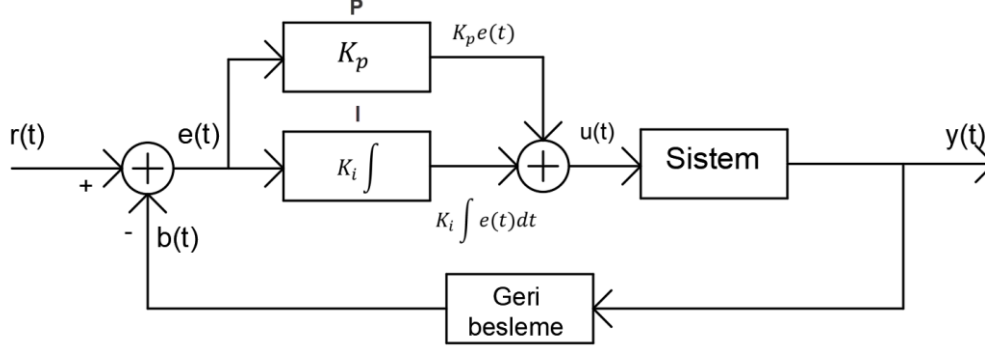
$$u(t) = K_p e(t) + K_i \int e(t) dt \quad (3.12)$$

olarak elde edilir.

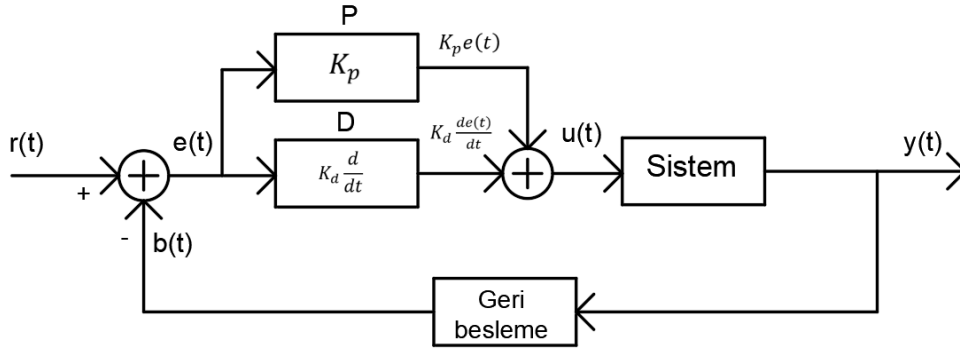
PD kontrolörde P kontrolöre türev eklenir (Şekil 3.8). Türev kontrolör kısmında hata sinyalinin türevi alınarak kapalı çevrim sistemin geçici yanıtı iyileştirilir. Türevsel kontrol ile sisteme öngörü kazandırılarak hatanın hangi yöne gittiği belirlenir ve aşım engellenir. PD kontrol ile sistemin sönümü ve yükselme zamanı düzeltilebilir ancak kalıcı hal hatasını düzeltmede etkisi çok azdır. PD kontrolör çıkışı;

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (3.13)$$

olarak elde edilir.



Şekil 3.7. PI kontrolör blok diyagramı

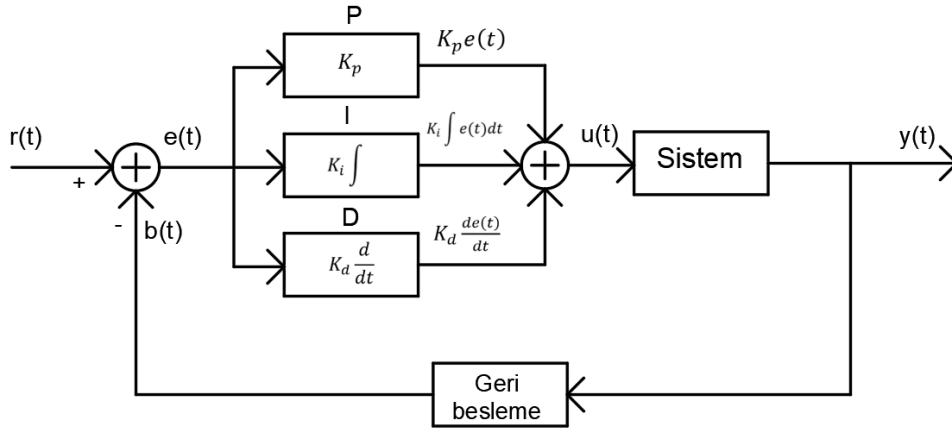


Şekil 3.8. PD Kontrolör blok diyagramı

Oransal kontrolöre integral ve türev eklenerek bu sistemlerin iyi yanları alınarak PID kontrolör gerçekleştirilir. PI kontrolör ile kararlılık iyileştirilerek diğer koşullar PD kontrol ile ayarlanabilir (Kuo 1999). PID kontrolör çıkışı;

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (3.14)$$

olarak elde edilir.



Şekil 3.9. PID kontrolör blok diyagramı

PID parametrelerinin sistem dinamikleri üzerine etkileri Çizelge 3.2’de gösterilmiştir (Jenkins 2016).

Çizelge 3.2. PID parametrelerinin sistem dinamikleri üzerine etkileri

Kontrol Parametreleri	Yükselme Zamanı	Aşım	Yerleşme Zamanı	Kalıcı Hal Hatası
K_p	Düşer	Yükselir	Az bir değişim	Düşer
K_i	Düşer	Yükselir	Yükselir	Giderir
K_d	Az bir değişim	Düşer	Düşer	Az bir değişim

PID kontrol tasarımında dikkat edilmesi gereken unsurlar:

- Yükselme zamanını iyileştirmek için oransal kontrol eklenir.
- Kalıcı durum hatasını gidermek için integral kontrol eklenir.
- Aşmayı azaltmak için türev kontrol eklenir.
- Kontrol parametreleri sistem yanıtı düzeltilene kadar ayarlanır.
- Eğer sadece PI kontrolör yeterliyse türev eklemeye gerek yoktur. Kontrolörü basit tutmakta fayda vardır.

3.5. PID Parametrelerini Ayarlama Metotları

PID kontrolör transfer fonksiyonu;

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (3.15)$$

olarak ifade edilir.

Kontrolör ayarı, sistemde istenen performans koşullarını yerine getirmek için kontrol parametrelerini (K_p , T_i ve T_d) seçme işlemine denir. PID parametreleri belirlemek için farklı yöntemler geliştirilmiştir (Kansagara 2018). Bunlardan bazıları aşağıda verilmiştir:

- a) Zeigler-Nichols metodu
- b) Cohen-Coon ayarlama metodu
- c) Wang-Juang-Chan ayarlama metodu
- d) Chien-Hrones-Reswick ayarlama metodu
- e) Deneme yanılma metodu
- f) Yazılım paketlerinin kullanılması

Bu metotlar K_p , T_i ve T_d parametreleri için uygun yaklaşımlar sunar. Sistemin kararlı bir şekilde çalışması için geliştirilmiştir. Bu metotlar ile bulunan parametreler her zaman sistem için uygun olmayabilir. Kabul edilemez bir aşım veya sonuçlar oluşabilir. Örneğin Ziegler-Nichols metodu ile tahmini kontrolör parametreleri belirlenerek ince ayarlar sonradan yapılabilir (Mohamed 2015).

Kontrol parametreleri deneme yanılma yöntemiyle bulmak için K_i ve K_d sıfırlanarak K_p kazancı sistem dengeli bir şekilde salınım yapana kadar artırılır. Daha sonra K_i kazancı salınım duracak ve kararlılık sağlanacak şekilde ayarlanır. Son olarak K_d kazancı aşımı önleyecek şekilde belirlenir (Kansagara 2018).

Ziegler-Nichols yöntemi ise transfer fonksiyonunun hesaplanmasının zor olduğu durumlarda PID kontrolör parametreleri belirlemek için kullanılabilir. Bu yöntemde basamak yanıtı ve frekans yöntemi olarak iki şekilde parametreler belirlenebilir. Ziegler-Nichols frekans yöntemi PID kontrolörünün parametrelerini ayarlamak için pratik bir yaklaşım sağlar. Bu yöntemde göre PID kontrol sistemi öncelikle P kontrolöre alınır ($T_i = \infty$ ve $T_d = 0$). Sistem sürekli salınım yapmaya kadar kazanç artırılır. Sürekli salınım olduğu durumdaki kazanç K_u , salınımın bir periyodu P_u ile ifade edilir. Böylece Çizelge 3.3'te verilen Ziegler-Nichols ayar tablosu kullanılarak PID parametreleri belirlenir. Kontrol kazancı G_m olarak

isimlendirilir. Kontrol kazancı artarsa sistem ω_{cg} frekansında salınım yapar. K_u ve P_u değerleri G_m ve ω_{cg} 'den aşağıdaki gibi elde edilebilir (Cao 2019).

$$K_u = G_m \text{ veya } 20\log_{10}(K_u) = G_m(dB) \text{ ise } K_u = 10^{G_m/20} \quad (3.16)$$

$$P_u = 2\pi/\omega_{cg} \quad (3.17)$$

Çizelge 3.3. Ziegler-Nichols ayar tablosu (Cao 2019)

Kontrol	Kp	Ti	Td
P	$K_u / 2$		
PI	$K_u / 2,2$	$P_u / 1,2$	
PID	$K_u / 1,7$	$P_u / 2$	$P_u / 8$

4. MOBİL ROBOT BİLEŞENLERİ ve ROBOT TASARIMI

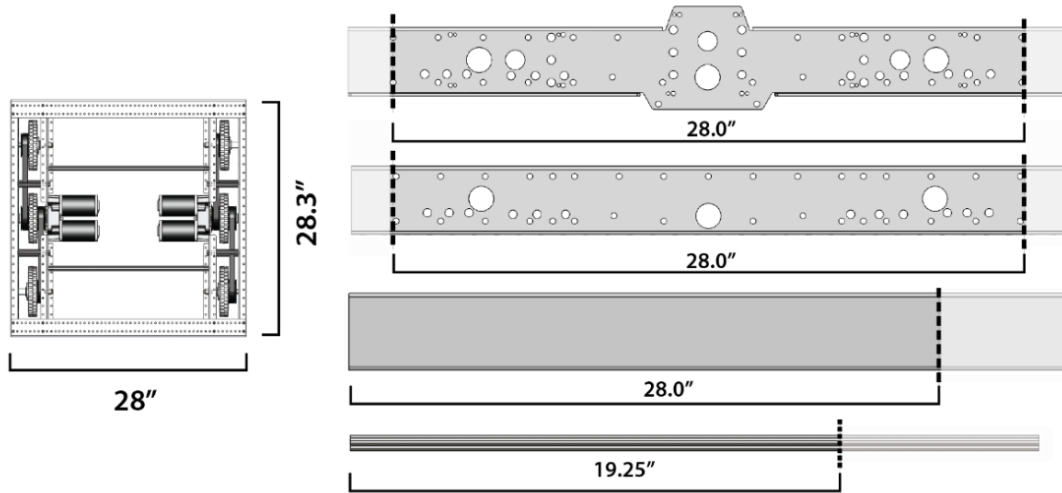
Bu bölümde mobil robotta kullanılan bileşenler, robot sisteminin elektronik ve yazılım tasarımları hakkında bilgi verilmiştir. Bu tezde amaçlanan konunun mobil robotların yörünge ve hareket kontrolü olmasından dolayı robotik sistemde pratik olması açısından First Robotics Competition(FRC) robotları için özel olarak tasarlanmış robot şasesi, motorlar ve elektronik bileşenler kullanılmıştır.

4.1. Mekanik Bileşenler

Mekanik bileşenler robot şasesi, lift sistemi, kol sistemi ve dişli kutularından oluşmaktadır.

4.1.1. Robot şasesi

Robot şasesi olarak AndyMark firmasının tasarlamış olduğu Şekil 4.1’de görülen AM14U4 model şase kullanılmıştır. Şasede 6 tekerlek bulunmaktadır. Merkezdeki tekerleklere dişli kutusu bağlanarak tork arttırılmıştır. Diğer tekerlekler merkezdeki tekerleğe bağlı kayışlar ile döndürülmektedir (AndyMark 2019).



Şekil 4.1. Robot şasesi (AndyMark 2019)

4.1.2. Motor dişli kutusu

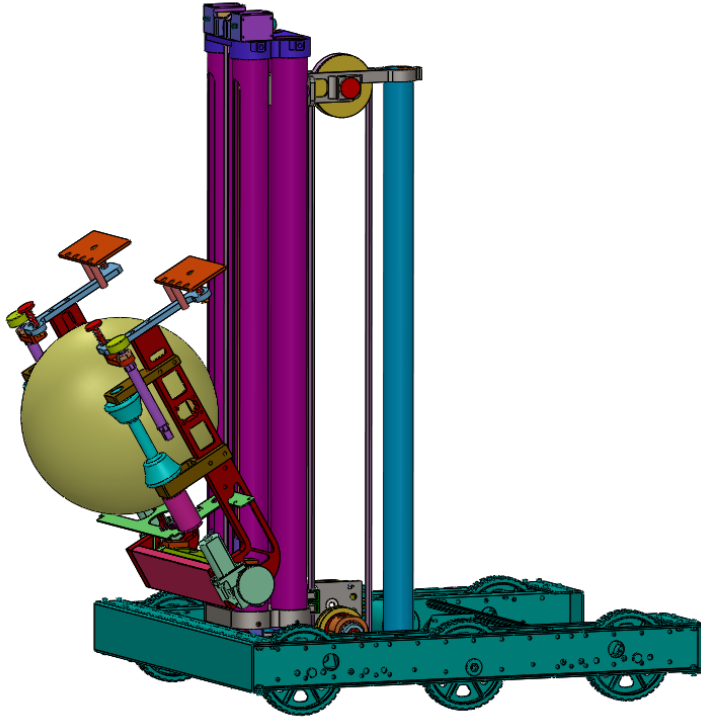
Robot şasesinde ve lift sisteminde Şekil 4.2’de gösterilen AndyMark firmasının CIM motorlar için tasarlamış olduğu Toughbox Mini dişli kutusu kullanılmıştır. Dişli oranı 10.71:1’dir. Dişli kutuları ile motorlardan alınan tork arttırılmıştır (AndyMark 2019a) .



Şekil 4.2. Motor dişli kutusu (AndyMark 2019a)

4.1.3. Robot lift ve robot kol sistemi

Robot lift ve kol sistemlerinde CNC İleri Teknoloji firması tarafından tasarımı yapılan mekanik bileşenler kullanılmıştır. Şekil 4.3'te mobil robot sistemi gözükmemektedir.

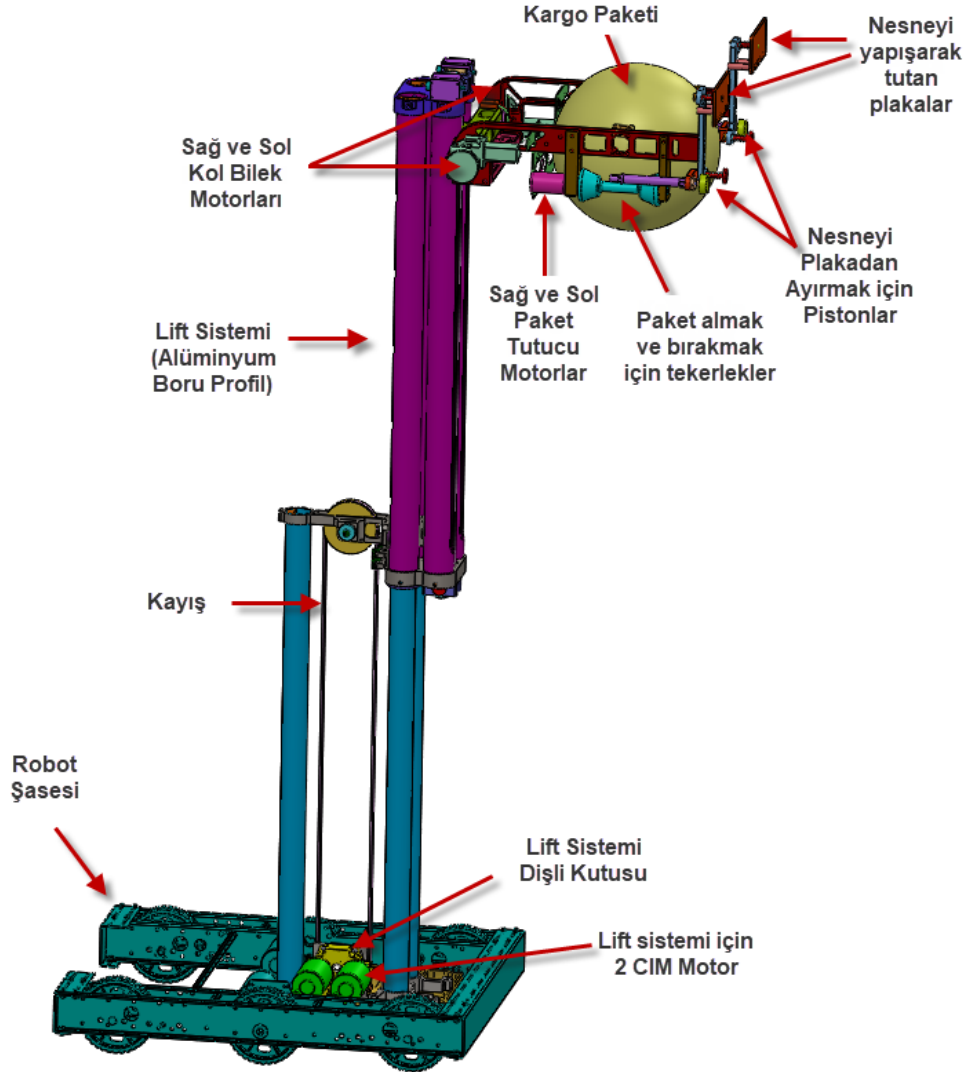


Şekil 4.3. Mobil robot sistemi

Şekil 4.4'te mobil robot mekanik bileşenleri gösterilmiştir. Robot lift sistemi için alüminyum boru profiller kullanılmıştır. Lift sistemine bağlı CIM motorların torku, dişli kutusu ile arttırılmıştır. Lift sistemi kasnak ve kayış sistemi kullanılarak motorlar tarafından yukarı aşağı hareketi sağlanır.

Robot sağ ve sol kol bileklerinde Snow Blower motor kullanılarak robot kolunun aşağı yukarı hareketi sağlanmıştır. Kargo paketlerini tutup bırakmak için iki kolda BAG motorlar kullanılmıştır. Bu motorlara bağlı tekerlekler içeri doğru döndürüldüğünde paket kola yerleşmekte ve dışarı doğru döndüğünde ise paket bırakılmaktadır.

Koldaki iki plaka bir nesneyi yapışarak almak için kullanılmıştır. İki pnömatik piston ise yapışan nesneyi iterek ayırmak için kullanılmıştır.



Şekil 4.4. Mobil robot mekanik bileşenleri

4.2. Robotun Elektronik Bileşenleri

Robot sistemindeki elektronik bileşenlerinin özellikleri aşağıdaki başlıklarda verilmiştir.

4.2.1. RoboRIO

Mobil robot sisteminde mikrodenetleyici olarak National Instruments firması tarafından üretimi yapılan roboRIO kullanılmıştır (Şekil 4.5). RoboRIO I2C, SPI, USB, RS232, Ethernet, PWM, analog giriş ve dijital giriş-çıkış pinleri içeren programlanabilir robot kontrol cihazıdır. Bu denetleyici ayrıca dahili olarak ivme ölçer portu ve elektronik bağlantı portuna sahiptir (National Instruments 2019a).



Şekil 4.5. Roborio denetleyicisi (National Instruments 2019a)

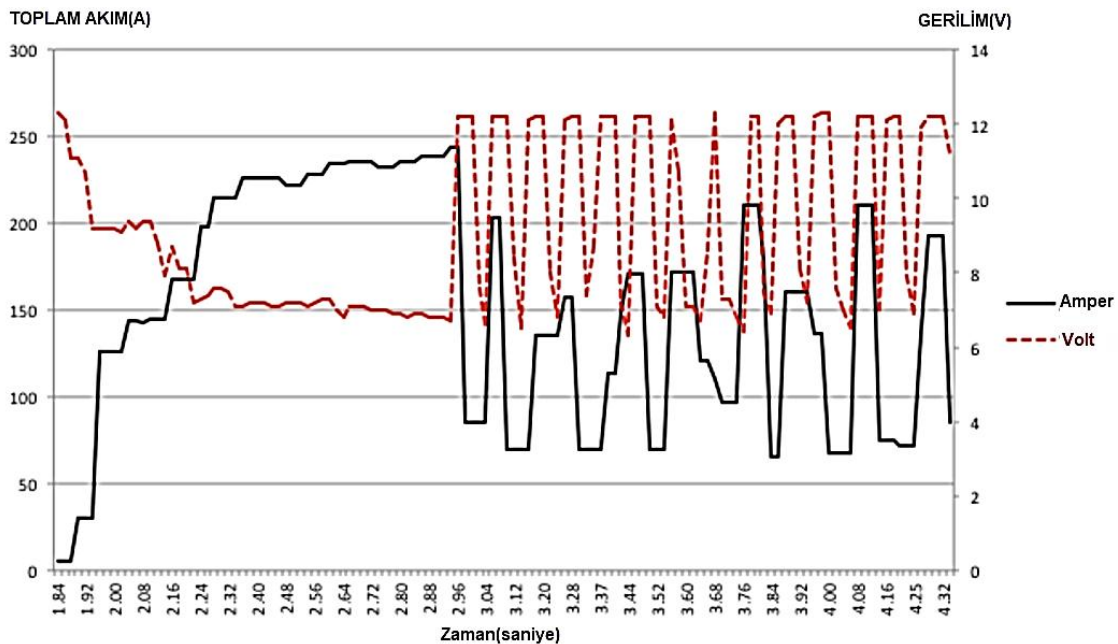
RoborRIO özellikleri:

- RoboRIO da bulunan Xilinx Z-7020 çift çekirdekli ARM Cortex-A9 işlemcisinin hızı 667 MHz'dir.
- Kalıcı belleği 512MB ve RAM belleği ise 256MB 533MHz hızında ve 16 bitliktir.
- Ağ bağlantısı 10BaseT ve100BaseTX Ethernet ile sağlanır. IEEE 802.3 standardına sahiptir.
- USB host olarak 2 adet port bulunur. Bu portlar USB 2.0 ve port başına maksimum 900mA verebilir. Bir adet bulunan USB aygıtı ise yine USB 2.0'dır.
- Analog giriş kanalı sayısı 8'dir. ADC birimi 12 bit ile örnekleme yapar. Nominal 0 ile 5V arasında gerilimler örneklenebilir. Örnekleme hızı 500kS/s ve bant genişliği minimum 20kHz'dir.

- Dijital giriş ve çıkış olarak 10 adet DIO pini, I2C(1 SDA ve 1 CLK) ve SPI pinleri bulunur.
- UART bağlantısı maksimum baud rate 230400 bps'dir.
- RS-232 seri portu maksimum baud rate 115200 bps'dir.
- 10 adet PWM bağlantısı ve 4 adet röle bağlantısı bulunur.
- RSL bağlantı noktasına 7-16V çalışan sinyal lambası takılabilir. Bu port maksimum 120mA verilebilir.
- RoboRIO çalışma gerilimi 7-16V_{DC}'dir. Güç tüketimi tipik olarak 5W maksimum 45W'tır.

RoboRIO blok diyagramı EK 1 ve giriş çıkış portları EK 2'de verilmiştir.

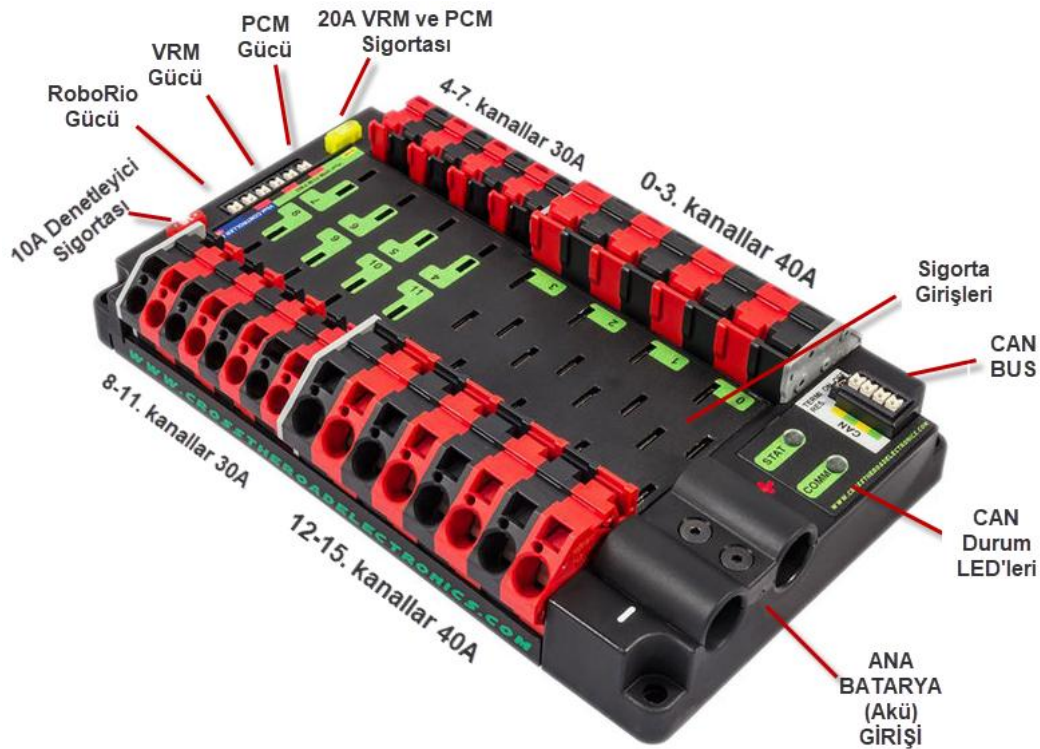
RoboRIO, ayrıca brownout korumasına sahiptir. Bu koruma sistemi yüksek akım çekilmesi durumunda meydana gelen akü geriliminin düşmesi ile roboRIO'nun sıfırlanmasını önlemek ve diğer kontrol sistemi bileşenlerini korumak için aşamalı bir koruma sağlar. Ani yüksek akımlara karşı robotun etkilenmemesi için aktarma organlarındaki motordan çekilen akımlar tasarımda dikkate alınmalıdır. Şekil 4.6'daki grafikten de görüldüğü gibi 240A çekilmesi durumunda gerilimin 7V'un altına düşmektedir. Toplam akımın maksimum 180A civarında tutulması ile sistemin etkilenmesinin önüne geçilebilir. Örneğin 4 motor aynı anda 40A akım çekerse yaklaşık sınıra ulaşılır. Bu sebepten dolayı robotik sistemde hangi fonksiyonların aynı anda çalışacağı dikkate alınmalıdır (WPI 2019).



Şekil 4.6. Toplam akım-gerilim eğrisi (WPI 2019)

4.2.2. Güç dağıtım paneli (PDP)

PDP Cross the Road Electronics firması tarafından üretilmiştir. PDP ile aküden çekilen akım sigortalardan geçtikten sonra tüm sistem bileşenlerine aktarılır. PDP roboRIO'ya doğrudan CAN ile bağlanır ve her kanaldaki akımlar izlenebilir. PDP giriş gerilimi 5,5V-16V arasında olabilir. Şekil 4.7'de güç dağıtım panelinin giriş-çıkış bağlantıları görülmektedir. PDP'nin 0-3, 12-15 arası kanalları için 40A sigorta, 4-11 arası kanalları için 20A veya 30A'lık sigortalar kullanılabilir. VRM ve PCM çıkışları için 20A sigorta ve roboRIO çıkışı için 10A sigorta kullanılır (CTR Electronics 2019).

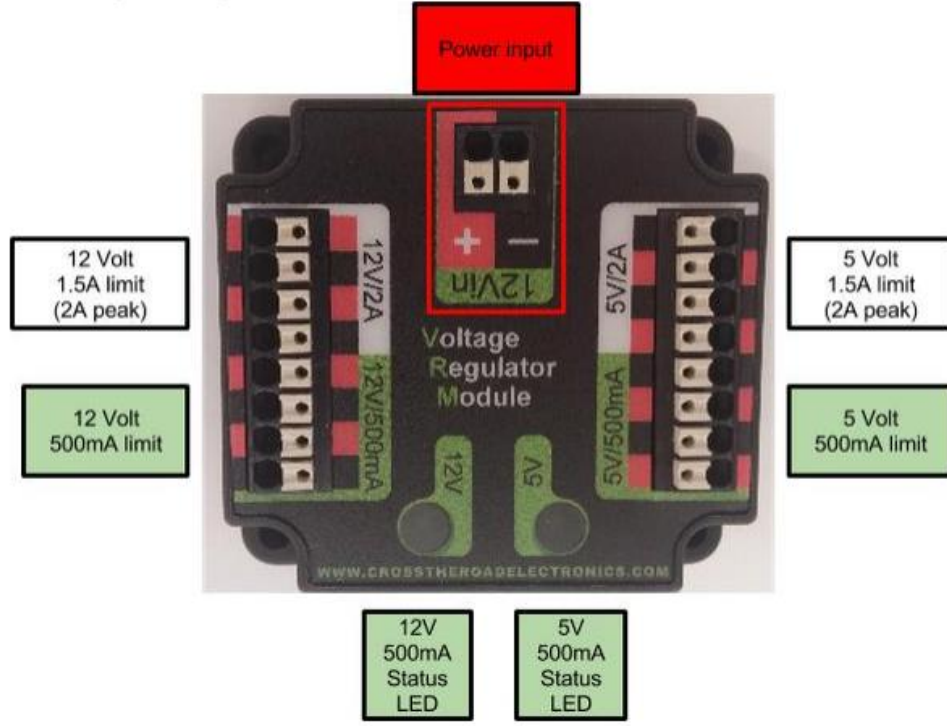


Şekil 4.7. Güç dağıtım paneli (CTR Electronics 2019)

4.2.3. Voltaj regülatör modülü(VRM)

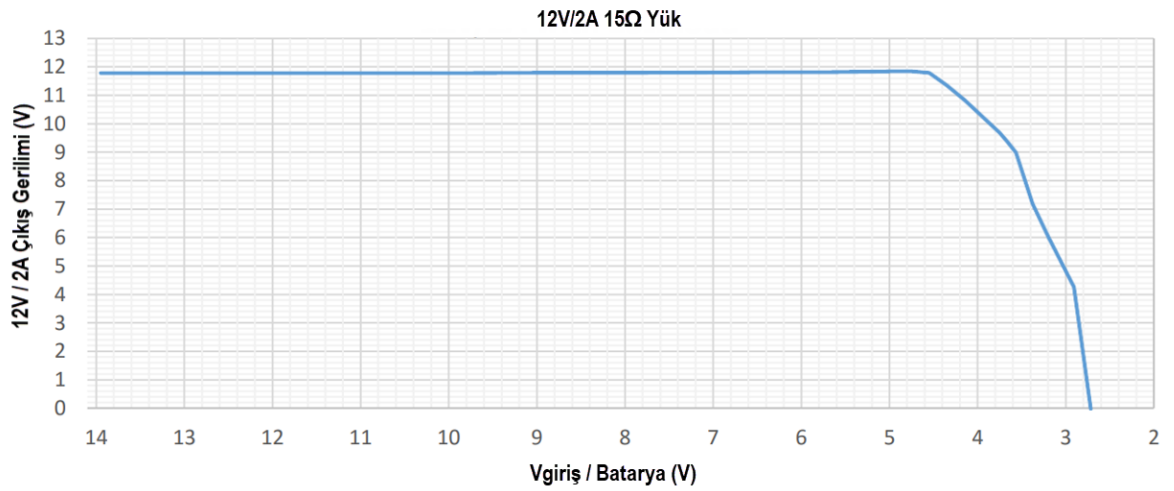
Voltaj regülatör modülü sistemde kullanılan cihazlara sabit gerilim uygulamak için kullanılan DC-DC dönüştürücüdür. VRM, giriş geriliminin istenen çıkış geriliminden yüksek veya düşük olması durumunda çıkış gerilimini sabit tutmak için kullanılmıştır. Kutupların ters bağlanmasına karşı koruması vardır. Robotik sistemde kullanılan Access Point VRM'nin 12V/2A çıkışına bağlanmıştır. VRM ile sistemde kullanılan arduino, sensörler ve özel devreler için regülasyon yapılır. VRM giriş gerilimi 5-16V arasında olabilir ve tipik olarak

12V ile beslenir. Şekil 4.8’de görüldüğü gibi VRM 5V/500mA, 5V/2A, 12V/500mA ve 12V/2A çıkışlara sahiptir (CTR Electronics 2019a).



Şekil 4.8. Voltaj regülör modülü (CTR Electronics 2019a)

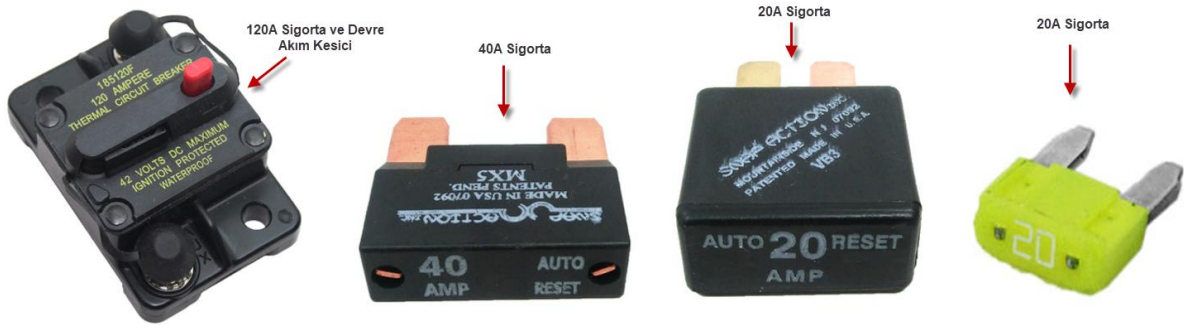
Şekil 4.9’da VRM’nin farklı giriş gerilimlerine karşılık 12V/2A çıkışının sabit kalma durumu gösterilmiştir. Modül girişindeki gerilim 5V’a düşene kadar çıkış gerilimi 12V’ta sabit tutulmuştur (CTR Electronics 2019a).



Şekil 4.9. VRM modülü giriş-çıkış gerilimi eğrisi (CTR Electronics 2019a)

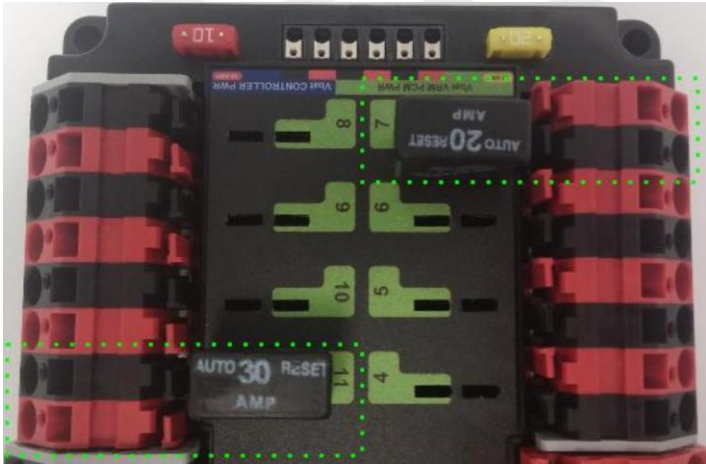
4.2.4. Sigortalar

Sigortalar herhangi bir kısa devre ve yüksek akıma karşı sistemi korumak amacıyla kullanılmıştır. PDP'ye bağlı sigortalar 10A, 20A, 30A ve 40A'dır. Ayrıca 120A'lık devre akım kesici sigorta ile sisteme akım geçişi sağlanır ve aynı zamanda sistem yüksek akıma karşı korunur. Robotik sistemde kullanılan sigortalar Şekil 4.10'da gösterilmiştir (AndyMark 2019b).



Şekil 4.10. Sigorta çeşitleri (AndyMark 2019b)

Güç dağıtım paneline sigorta bağlantıları Şekil 4.11'de görülmektedir.



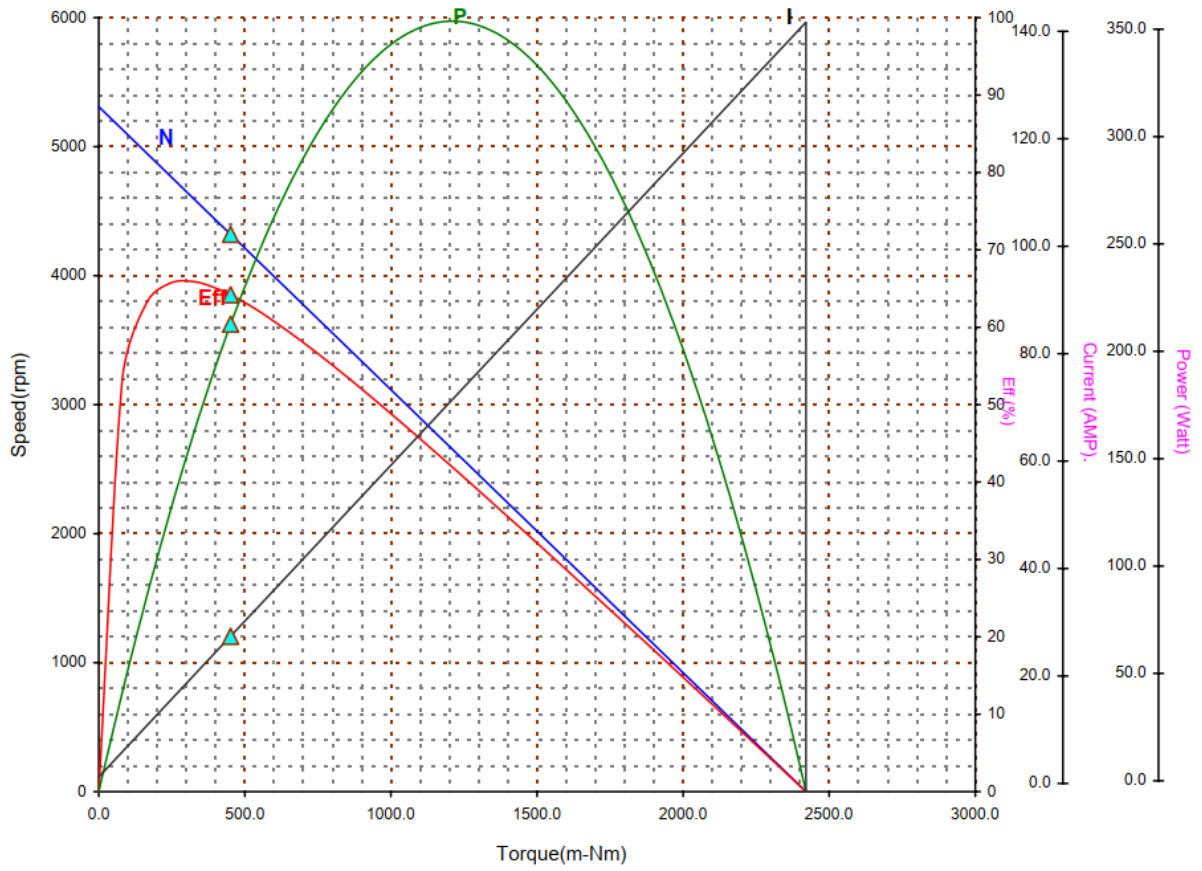
Şekil 4.11. PDP sigorta bağlantısı (CTR Electronics 2019)

4.2.5. CIM motor

Robotun hareket ve lift sisteminde kullanılan DC motordur(Şekil 4.12). Çalışma gerilimi 12V, yüksüz akımı 2.7A, yüksüz hızı 5310RPM, maksimum gücü 337W, durak akımı 133A ve durak torku 343,4in-oz dur. Motor şaftının yarıçapı 0,398cm'dir. CIM motor çalışma eğrileri Şekil 4.13'te verilmiştir (AndyMark 2019c).



Şekil 4.12. CIM motor (AndyMark 2019c)



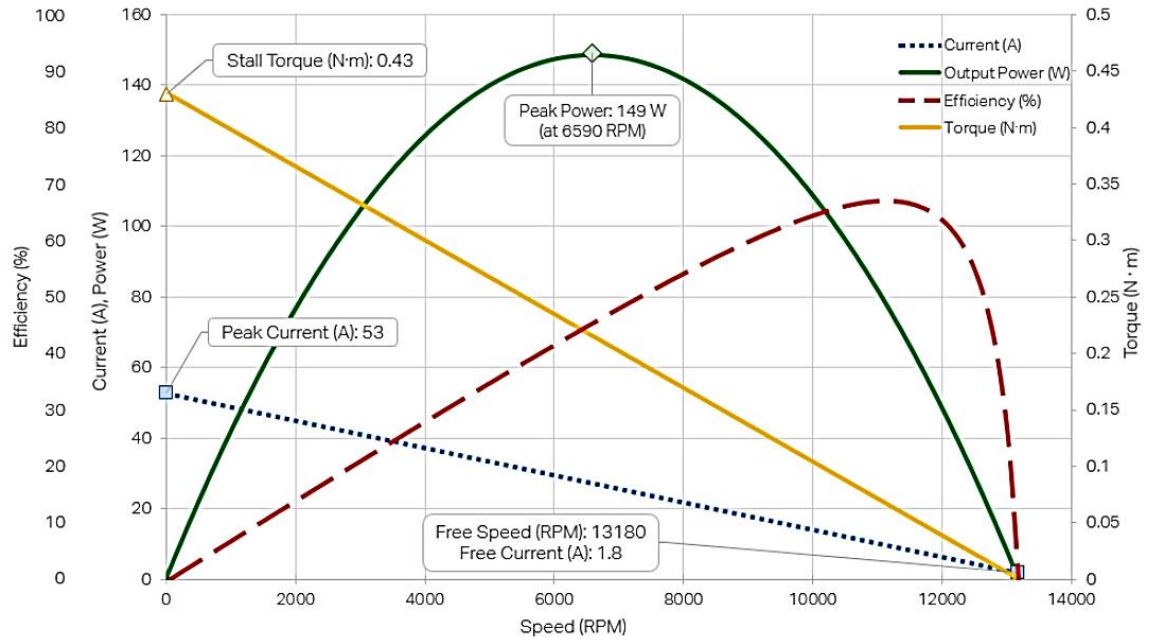
Şekil 4.13. CIM motor tork-hız grafiği (AndyMark 2019c)

4.2.6. BAG motor

Mobil robotun bir paketi alıp bırakması için robot kollarında BAG motor kullanılmıştır (Şekil 4.14). Çalışma gerilimi 12V, yüksüz akımı 1,8A yüksüz hızı 14000RPM, maksimum gücü 147W, durak akımı 41A ve durak torku 0,4Nm'dir. BAG motor çalışma eğrileri Şekil 4.15'te verilmiştir (VEX Robotics 2019).



Şekil 4.14. BAG motor (VEX Robotics 2019)



Şekil 4.15. BAG motor eğrileri (VEX Robotics 2019)

4.2.7. Snow blower motor

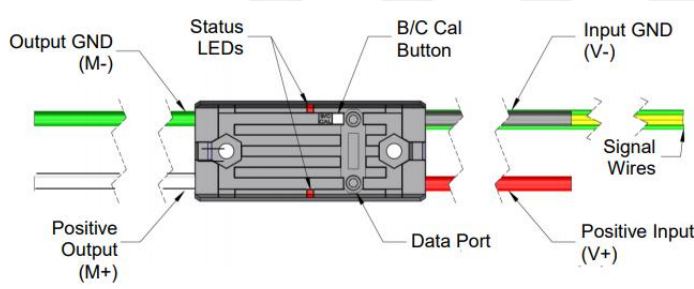
Mobil robotun kol bileğini hareket ettirmek için kullanılan bu motor dişli kutusu ile tek bir paket halindedir (Şekil 4.16). Güç tüketimi düşük olduğu için robotik sistemlerde kullanılabilir. Çalışma gerilimi 12V, yüksüz akımı 5A, yüksüz hızı 100RPM, maksimum gücü 20W, durak akımı 24A ve durak torku 70 in-lb'dir (AndyMark 2019d).



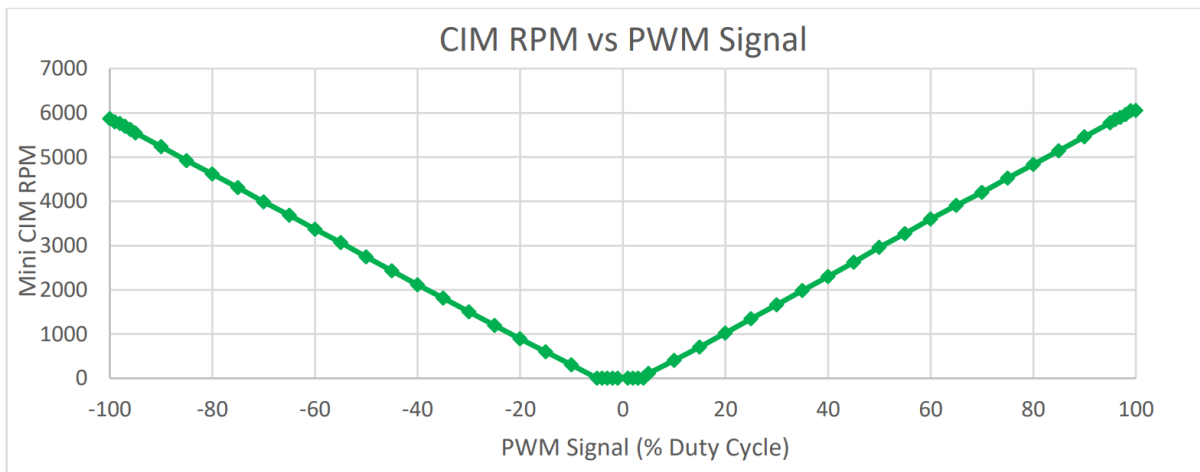
Şekil 4.16. Snow Blower motor (AndyMark 2019d)

4.2.8. TALON SRX motor sürücüsü

Cross the Road Electronics firması tarafından üretilen Talon SRX motor sürücüsü mobil robot sisteminde robota yön veren motorlara bağlanmıştır. Bu motor sürücü CAN ve PWM iletişim protokolleri ile kontrol edilebilir. Tümüleşik PID algoritmalarına hızlı cevap verebilir. Motor sürücü bileşenleri elektriksel olarak yalıtılmış olduğu için kısa devre yapmaz. Sağlam alüminyum bir kasa içerisinde soğutma işlemi yapılır. LED göstergeleri ile hata ayıklama ve çalışma durumları izlenebilir. Sürekli 60A akım verebilir. Çalışma gerilim aralığı 6-28V arasındadır ve nominal çalışma gerilimi 12V'tur. Sensör bağlantısı için girişlere sahiptir. Anahtarlama frekansı 15KHz, dalgalanma akımı 100A, PWM girişi yükselme zamanı 1-2ms ve PWM girişi periyodu 2,9-100ms'dir. Sahip olduğu CAN kabloları ile 63 adede kadar Talon SRX iletişim kurabilir. Sensörler doğrudan Talon SRX' e data portu üzerinden bağlanabilir. Talon motor sürücü bağlantıları Şekil 4.17'de, Talon SRX PWM sinyali ile kontrol edilen mini CIM motorun PWM değerlerine karşılık gelen hız değerleri Şekil 4.18'de verilmiştir (CTR Electronics 2019b).



Şekil 4.17. Talon motor sürücü bağlantıları (CTR Electronics 2019b)



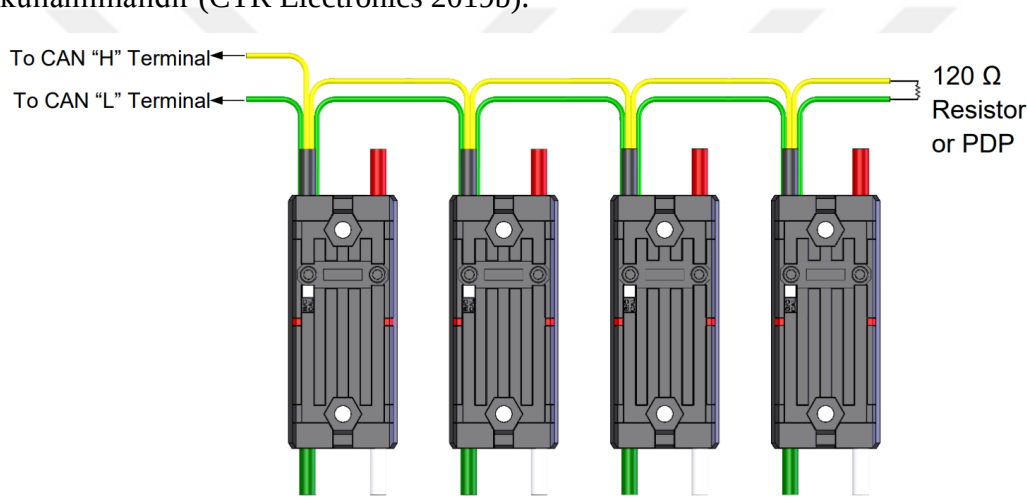
Şekil 4.18. Mini CIM motorda PWM - Hız grafiği(CTR Electronics 2019b)

Talon Motor Sürücü ile PWM kontrolü için sürücüdeki sarı kablo roboRIO'daki S pinine, yeşil kablo roboRIO'daki GND pinine bağlanmalıdır. Bunun için Şekil 4.19' daki gibi bir kablo kullanılarak bağlantı yapılabilir (CTR Electronics 2019b).



Şekil 4.19. Talon motor sürücü PWM kablo bağlantısı (CTR Electronics 2019b)

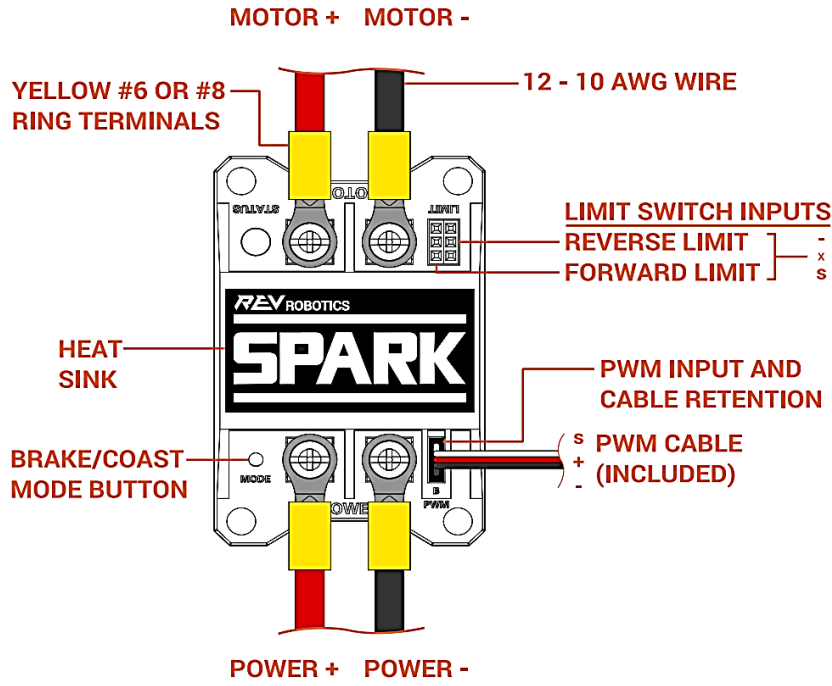
Talon motor sürücü CAN bağlantısı Şekil 4.20'de gösterilen şekilde aynı renkteki kabloların birleştirilmesi ile yapılır. Bağlantıyı sonlandırmak için 120 Ω 'luk bir direnç kullanılmalıdır (CTR Electronics 2019b).



Şekil 4.20. Talon motor sürücü CAN kablo bağlantıları (CTR Electronics 2019b)

4.2.9. SPARK motor sürücüsü

SPARK motor sürücüsü pasif soğutmalı PWM ile kontrol edilen fırçalı DC motor sürücüsüdür(Şekil 4.21). Mobil robotik sistemde lift ve kollardaki motorları kontrol etmek için kullanılmıştır. Üzerinde bulunan RGB LED ile motorun çalışma durumları izlenebilir. Nominal çalışma gerilimi 12V'tur. Sürekli 60A verebilir. Tepe akımı 2 saniye için 100A'dır. Giriş darbe genişliği aralığı 1ms-2ms, giriş çözünürlüğü 1 μ s, çıkış frekansı 15,625kHz, çıkış gerilim aralığı 0V - $\pm$ Vin, maksimum çıkış voltajı çözünürlüğü 0,001xVin'dir. Limit switch (sınır anahtarı) girişlerinden sinyal ve - giriş kısa devre edilerek motor hareketleri hem ileri hem de geri yönde programlama gerekmeden engellenebilir (REV Robotics 2019).



Şekil 4.21. SPARK motor sürücü bağlantıları (REV Robotics 2019)

4.2.10. Access point

WiFi ile robota kablosuz erişim sağlamak için Open-Mesh OM5P-AC Access point kullanılmıştır (Şekil 4.22). Kısa mesafelerde 1,17 Gb/sn gibi yüksek hız sağlar. İki adet dahili çift bantlı 2x2 (2,4 GHz) + 2x2 (5 GHz) antene sahiptir. DRAM belleği 128MB'dır. 2 adet Gigabit Ethernet (WAN ve LAN) portları vardır. Gücü 19dBm'dir. 802.11b/g/n 2.4 GHz + 802.11a/n/ac 5 GHz bandlarında çalışır. 25-50' iç mekan ve 100' dış mekan menzili vardır (AndyMark 2019e).



Şekil 4.22. Access Point (AndyMark 2019e)

4.2.11. Akü

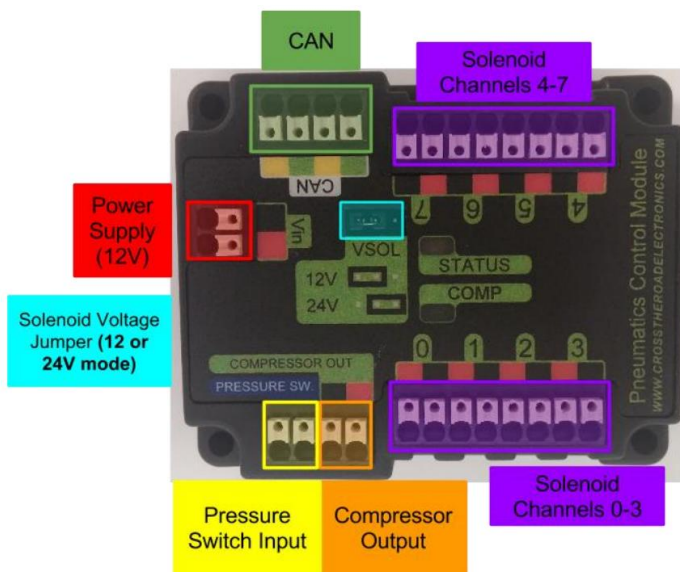
Robotun tüm gücü 12V 18Ah'lık bir akü ile sağlanmıştır (Şekil 4.23). Kurşun asitli bir aküdür (AndyMark 2019f).



Şekil 4.23. 12V/18Ah Akü (AndyMark 2019f)

4.2.12. Pnömatik kontrol modülü(PCM)

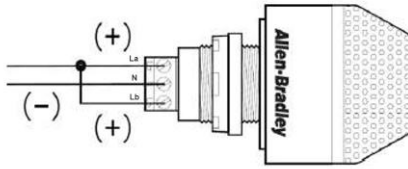
Pnömatik kontrol modülü mobil robotik sistemde kompresör, selenoid ve basınç anahtarı gibi elemanları kontrol etmek için kullanılmıştır. Şekil 4.24'te modülün giriş ve çıkışları gösterilmiştir. Bu modül ile kompresör ve 8 selenoid CAN haberleşmesiyle kontrol edilebilir. Üzerindeki jumper ile 12V veya 24V selenoid seçimi yapılabilir (CTR Electronics 2019c).



Şekil 4.24. Pnömatik kontrol modülü (CTR Electronics 2019c)

4.2.13. Robot sinyal lambası

Mobil robotik sistemde kullanıcıya robotun çalışma durumu hakkında ışıklı uyarı vermek için kullanılmıştır. Sinyal lambası 12V AC veya DC ile çalışabilmektedir. Sinyal lambası Şekil 4.25'te gösterildiği gibi bağlantısı yapıldıktan sonra RoboRIO üzerindeki RSL portuna bağlanır. Lambanın sürekli yanması robota enerji verildiğini ancak robotun bilgisayar sistemi üzerinden aktif edilmediğini belirtir. Lambanın yanıp sönmesi robota enerji verildiğini ve robotun aktif olduğunu belirtir. Lambanın sönük olması ise robota enerji verilmediğini, roboRIO'nun çalışmadığını veya lamba bağlantılarının RSL portuna doğru yapılmadığını gösterir (WPI 2019a).



Şekil 4.25. Robot sinyal lambası (Anonim 2019)

4.2.14. Enkoder

Enkoder olarak CIM motorlar için özel üretilmiş ve CIM motor miline doğrudan bağlanabilen manyetik bir enkoder olan CIMcoder kullanılmıştır (Şekil 4.26). Robotik sistemde lift sistemine bağlı motordan konum ve yön bilgisi alarak lift sistemini istenilen referans değere getirmek için kullanılmıştır. CIMcoder VDD, CH A, GND ve CH B pinlerine sahiptir. Çalışma gerilimi 3.3-5V_{DC} arasındadır. Devir başına 20 kare dalga çıkış üretir. 2 kanala sahiptir ve bu kanallardan alınan verilerle motorun yön ve konum bilgileri hesaplanır (AndyMark 2019g).



Şekil 4.26. CIMcoder (AndyMark 2019g)

4.2.15. Sınır anahtarı

Sınır anahtarları otomasyon sistemlerinde nesne ve konum algılamak için kullanılan bir algılama cihazıdır. Mekanik olarak nesne ile temas ettiğinde veya nesne ile temas kesildiğinde algılama yapılabilir. Sınır anahtarları ile nesnelerin renginden, şeklinden veya boyutundan ayrı olarak konum algılaması yapılır. Sınır anahtarları nesneye dokunduğu için hassas nesnelerde kullanılmaz. Hareketli sistemlerde kullanılırken aşınma meydana oluşabilir. Maliyetleri düşük ve kullanımı kolaydır (AutomationDirect 2019).

Robotik sistemde lift sistemine ait konum bilgisini sıfırlamak için kullanılmıştır. Lift en alt konuma geldiğinde sınır anahtarı kapanarak lift sistemine bağlı enkoder sıfırlanır. Bu sayede olası enkoder hatalarının önüne geçilmiştir.

Şekil 4.27’de sınır anahtarı bacak isimleri gösterilmiştir. Sınır anahtarı bağlantısı için ortak kontak(COM) GND’ye bağlanmıştır. Normalde açık kontak(NO) ise roboRIO DIO pinine bağlanarak durum bilgisi işlemciye iletilir. Normalde kapalı kontak kullanılmamıştır (Honeywell 2019).



Şekil 4.27. Sınır anahtarı (Honeywell 2019)

4.2.16. JİROSKOP

Robotun sabit bir yörüngede düz gitmesi için açısal hızı ölçmeye yarayan Şekil 4.28’de gösterilen ADXRS450 isimli Jiroskop (Gyroscope) kullanılmıştır. ADXRS450 ile z ekseninde ± 300 °/s 'ye kadar açısal hız algılanabilir. Açısal hıza ait veriler 32 bit SPI paketinin içinde 16 bit olarak gönderilir. RoboRIO üzerindeki SPI portuna doğrudan takılarak kullanılabilir. Jiroskop ile robotun hangi yöne döndüğü ve dönüş açısı bilgileri alınabilir. Jiroskop sıfırlanarak referans açı alındıktan sonra robot saat yönünde döndürüldüğünde robotun açısı saat yönünde artacaktır, saat yönünün tersi döndürülmesi durumunda ise azalacaktır (Analog Devices 2019).



Şekil 4.28. Jiroskop (AndyMark 2019h)

4.2.17. Kamera

Robot üzerinden bilgisayara görüntü almak için Şekil 4.30’da resmi verilen Microsoft Lifecam HD-3000 modeli kamera kullanılmıştır. Kamera ile roboRIO’ya USB 2.0 üzerinden bağlanarak bilgisayara görüntü aktarılır. CMOS görüntü sensörü olan kamerada kare hızı 30 fps’dir. Maksimum görüntü çözünürlüğü 1280x720’dır (Microsoft 2019).



Şekil 4.29. Microsoft Lifecam HD-3000 kamera (Microsoft 2019)

4.3. Yazılım Bileşenleri

Mobil robot sisteminde FRC kontrol yazılım bileşenleri kullanılmıştır. Bu bileşenler, robot kodunun oluşturulması, geliştirilmesi ve hata ayıklanmasında yardımcı olur (WPI 2019b).

4.3.1. LabVIEW FRC

LabVIEW, donanım yapılandırması, ölçüm verileri ve hata ayıklamada dahil olmak üzere uygulamayı tüm yönleriyle görselleştirmeye yardımcı olan bir grafiksel bir programlama dilidir. National Instruments tarafından geliştirilmektedir. LabVIEW grafiksel ve veri akış odaklı bir dildir. Programların hazırlanması metin tabanlı dillerden daha kolaydır. Algoritmalar görsel olarak izlenebilir (WPI 2019b).

LabVIEW FRC, robotun programlanması için desteklenen üç dilden biridir. Diğer desteklenen diller JAVA ve C++’dır. LabVIEW FRC 2019 programı, LabVIEW ile programlama yapmak için kurulması gerekir (WPI 2019c).

4.3.2. FRC Update Suite

FRC Update Suite programı ile robotun kontrolü için gerekli olan sürücü istasyonu programı FRC Driver Station ve diğer FRC yardımcı programlarının kurulumu yapılır (WPI 2019c).

4.3.3. WPI robotik kütüphanesi

WPI Robotik kütüphanesi(WPILib) , FRC robot kontrol sistemi için tasarlanmış yazılım sınıfı setidir. Bu kütüphanede sensörler, motor sürücü cihazları, sürücü istasyonu, zamanlama ve saha yönetimi gibi ek yardımcı fonksiyonlar bulunur. Kütüphane C++ ve Java dillerini destekler. WPILib, robotta kullanılan özel donanım ve aygıtlara destek sağlamak amacıyla ek komut setlerine sahiptir. FPGA donanımı ile mikroişlemci performansı artırılırken kesme gibi gerçek zamanlı işlemlerdeki karmaşıklığı azaltır (WPI 2019d).

4.3.4. Visual Studio Code IDE (C ++ \ Java)

Visual Studio Code, robotu programlamak için gerekli olan 3 dilden ikisi olan C++ ve JAVA dillerini destekleyen geliştirme ortamıdır. Bu tezdeki programlar JAVA dili kullanılarak geliştirilmiştir (WPI 2019c).

4.3.5. FRC Radio Configuration Utility

Mobil robot sisteminin kablosuz haberleşmesi için OM5P-AC Access Point kullanılmıştır. Access point konfigürasyon ayarları FRC Radio Configuration Utility programı ile EK 3'teki gibi yapılmıştır. Access Point konfigürasyonu tamamlandıktan sonra robota kablosuz olarak kod atılabilir, robot kablosuz olarak yönetilebilir ve robotun çalışma durumları kablosuz olarak izlenebilir (WPI 2019c).

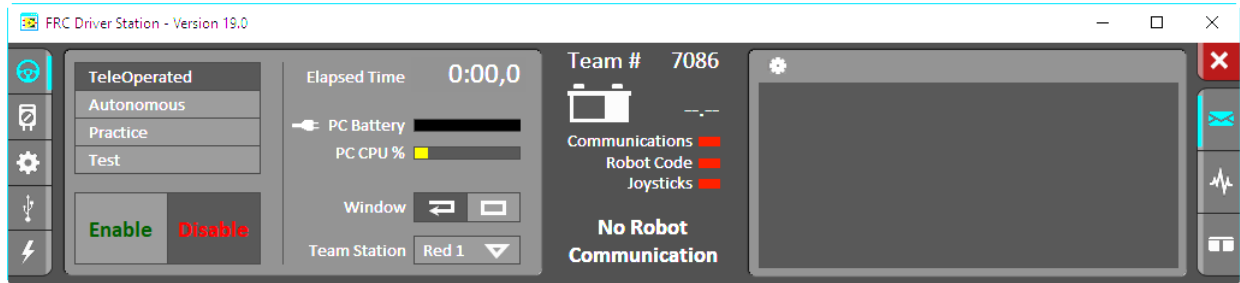
4.3.6. FRC roboRIO Imaging Tool

Robot kodları roboRIO'ya yüklenebilmesi için roboRIO'ya image işlemi yapılmalıdır. RoboRIO Imaging Tool programı ile roboRIO'daki mevcut image dosyaları görüntülenebilir aynı zamanda image dosyaları yüklenebilir. Image işlemi EK 4'te gösterilmiştir. RoboRIO image işleminden önce, FRC Update Suite programının kurulmuş olması ve roboRIO elektriksel gücü sağlanmış olmalıdır. RoboRIO USB Aygıt portundan PC'ye bir USB kablosu ile bağlanarak image işlemi yapılır (WPI 2019c).

4.3.7. FRC Driver Station

FRC Driver Station NI LabVIEW ile geliştirilmiştir. Robotun tüm kontrolü bu sürücü istasyonu tarafından yönetilir. Robot otonom veya sürücü moduna geçirilebilir, robot aktif veya pasif yapılabilir, bağlantının durumu, robot kodu, joysticks bağlantısı, akü gerilimi, bağlantı problemleri, elektriksel hatalar vb. birçok durum izlenebilir.

FRC Driver Station programı görünümü Şekil 4.30'da gösterilmiştir. Bağlantı için programdaki ayarlar sekmesinden takım numarası girilmelidir. TeleOperated modunda robot sürücü modunda çalışırken Autonomous modunda otonom olarak çalışır. Robotu aktif yapmak için Enable butonuna, pasif yapmak için ise Disable butonuna tıklanır (WPI 2019c).



Şekil 4.30. FRC Driver Station (WPI 2019c)

4.3.8. FRC LabVIEW Dashboard

FRC LabVIEW Dashboard, FRC Driver Station ile birlikte kurulur. FRC Driver Station açıldığında otomatik olarak başlatılır. Bu gösterge tablosunun amacı robotun çalışması hakkında kullanıcıya geri dönüt vermektir. Bu uygulama ile robota bağlı kameralardan görüntü alınabilir ayrıca robot IP adresi, akü durumu göstergesi ve programdaki değişken durumları vb. robot hakkında birçok bilgi bu gösterge tablosunda gösterilir (WPI 2019c).

4.3.9. Smart Dashboard

Smart Dashboard, robot ait verileri metin olarak veya grafiksel olarak gerçek zamanlı gösteren bir Java programıdır. Smart Dashboard uygulama linkine tıklayarak veya FRC Driver Statiton programındaki ayarlar sekmesinden açılabilir. Bu program ile robot programındaki değişken verileri robot çalışırken görüntülenerek programın akışı izlenebilir. Robota bağlı sensörlere ait veriler, aktüatörler, diğer giriş elemanlarının durum bilgileri ile robot kodundaki alt sistemlerin doğru çalışıp çalışmadığı hakkında bilgiler alınabilir (WPI 2019c).

4.3.10. Shuffleboard

Shuffleboard, robot ait verileri metin olarak veya grafiksel olarak gerçek zamanlı gösteren Smart Dashboard benzeri bir uygulamadır. Smart Dashboard'da yer almayan bazı özelliklere sahiptir. Arayüzü anlaşılabilirliği arttırmak için ızgara şeklindedir. Birden fazla sayfada veriler gösterilebilir ve çalışmalar saklanabilir (WPI 2019c).

4.4. Robot Elektronik Sistemi

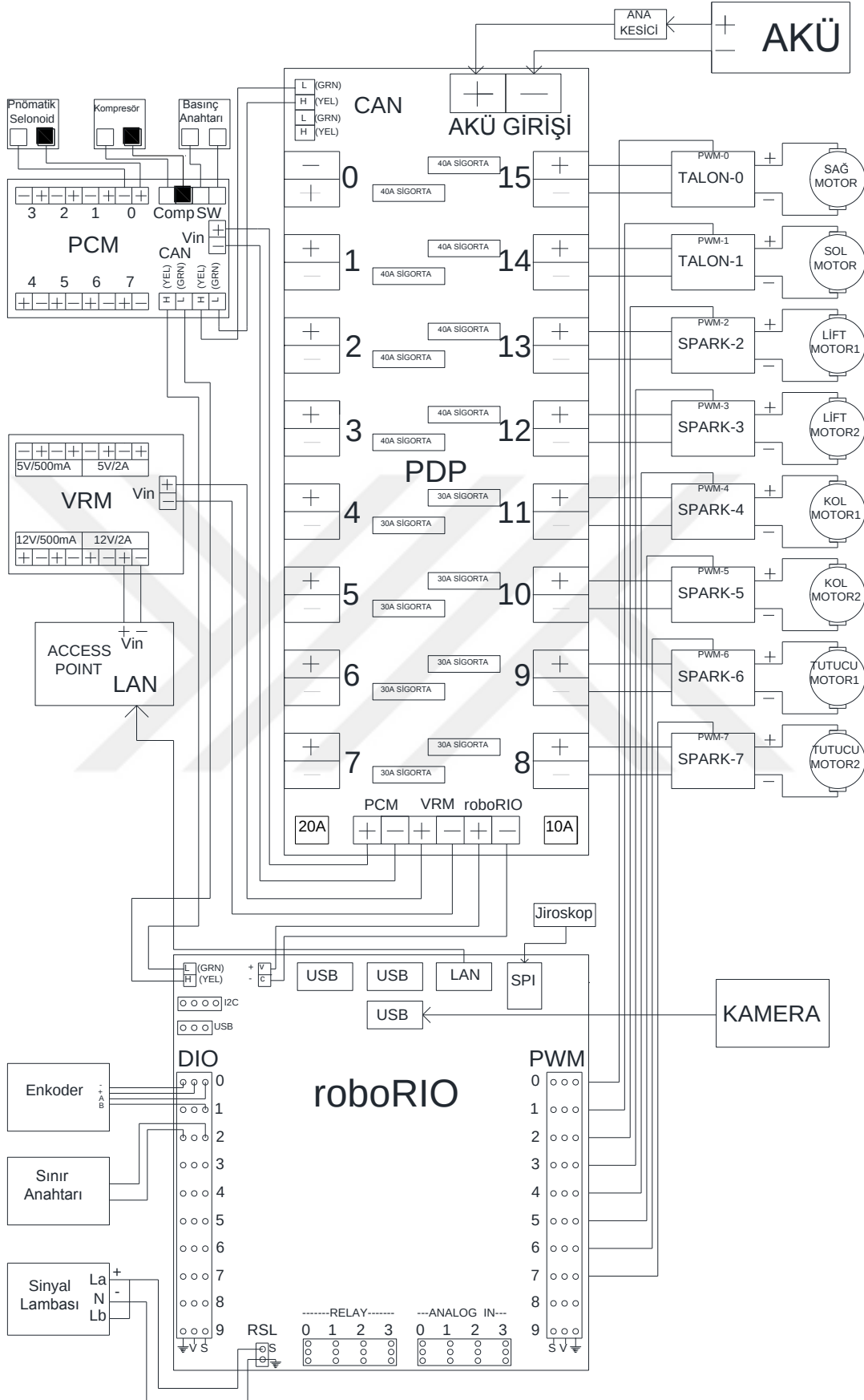
Robotun mekanik hareketleri elektronik sistem tarafından yönetilmektedir. Mobil robot sisteminin elektronik tasarımına ait blok diyagram Şekil 4.31'de görülmektedir.

- PDP, aküden aldığı gerilimi dağıtmak amacıyla kullanılmıştır. PDP üzerindeki sigortalar ayrı ayrı çıkışları yüksek akıma karşı korumaya yarar. Sistemde kullanılan 8 motor sürücü, roboRIO, VRM ve PCM modülleri 12V beslemesini PDP üzerinden almaktadır.
- 120A'lık sigorta barındıran ana kesici ile yüksek akımlara karşı koruma sağlanır aynı zamanda aküden PDP ye akım geçişi sağlanır. Acil durumlarda robota akım geçişini kesmek için robot üzerinde kolay ulaşılabilecek bir yere konulmuştur.
- RoboRIO sensörlerden(enkoder, kamera, jiroskop, sınır anahtarı, mesafe sensörü) ve kullanıcı tarafından aldığı verileri yüklenen koda göre işleyerek tüm robotun kontrolünü sağlar. Tuş takımından bilgiler roboRIO'ya kablosuz olarak Access Point ile iletilir. RoboRIO yüklenen yazılım ile robot otonom veya kullanıcı tarafından kumanda ile kablosuz olarak kontrol edilmiştir. RoboRIO 8 motor sürücüyü PWM ile kontrol eder (Çizelge 4.1). Ayrıca roboRIO pnömatik kontrol modülüne gerekli bilgileri göndererek pnömatik sistemin çalışmasını sağlar.
- Voltaj regülatör modülü (VRM) PDP'den aldığı 12V gerilimi farklı akım ve gerilim değerlerinde sabitlemek için kullanılmıştır. VRM çıkışları 12V/500mA, 12V/2A, 5V/500mA ve 5V/2A'dir. Access Point beslemesini VRM'deki 12V/2A çıkışından alır.
- Access Point LAN ile roboRIO'ya bağlanmıştır ve kablosuz olarak bilgisayar ile roboRIO arasındaki iletişimi kurmaktadır. Kullanıcı bilgisayar üzerinden Access Point ile bağlantı kurduktan sonra FRC Driver Station programı ile robotu kontrol edebilir.
- Jiroskop robota ait açısal verileri roboRIO'ya iletmek için kullanılmıştır. RoboRIO üzerinde bulunan SPI portuna bağlanmıştır.

- Kamera ile robot üzerindeki görüntüler kullanıcının bilgisayarına kablosuz olarak aktarılır. Smart Dashboard veya Shuffleboard programları ile görüntüler takip edilebilir.
- Enkoder, lift sistemine bağlı motora takılmıştır ve liftin konum bilgilerini roboRIO'ya gönderir. Enkoder roboRIO'daki DIO-0 ve DIO-1 pinlerine bağlanmıştır.
- Sınır anahtarı lift sistemi en alta konumlandığında lift enkoderi sıfırlayarak olası enkoder hataları engellemek için konulmuştur. RoboRIO'daki DIO-9 pinine bağlanmıştır.
- Sinyal lambası roboRIO RSL portuna bağlanarak robotun çalışma durumlarını ışıklı uyarı vererek gösterir.
- Pnömatik kontrol modülü(PCM) ile roboRIO CAN protokolünü kullanarak iletişim kurar. PCM'e 1 adet kompresör, 1 solenoid valf ve 1 adet pnömatik basınç anahtarı bağlanmıştır.

Çizelge 4.1. RoboRIO PWM çıkışlarına göre robotun hareketleri

RoboRIO PWM Çıkışları	RoboRIO PWM çıkışlarına göre robotun hareketleri
PWM-0 ve PWM-1	Robotun şasesine bağlı 2 adet motoru kontrol eden Talon motor sürücülere gerekli PWM sinyalleri gönderilerek robotun sağa-sola ve ileri-geri hareketleri sağlanır.
PWM-2 ve PWM-3	Robotun lift sistemine bağlı 2 adet motoru kontrol eden SPARK motor sürücülere gerekli PWM sinyalleri gönderilerek liftin hareketi sağlanır.
PWM-4 ve PWM-5	Robotun kolundaki bilek sistemine bağlı 2 adet motoru kontrol eden SPARK motor sürücülere gerekli PWM sinyalleri gönderilerek kolun hareketi sağlanır.
PWM-6 ve PWM-7	Robotun nesne tutma sistemine bağlı 2 adet motoru kontrol eden SPARK motor sürücülere gerekli PWM sinyalleri gönderilerek tutma sisteminin hareketi sağlanır.



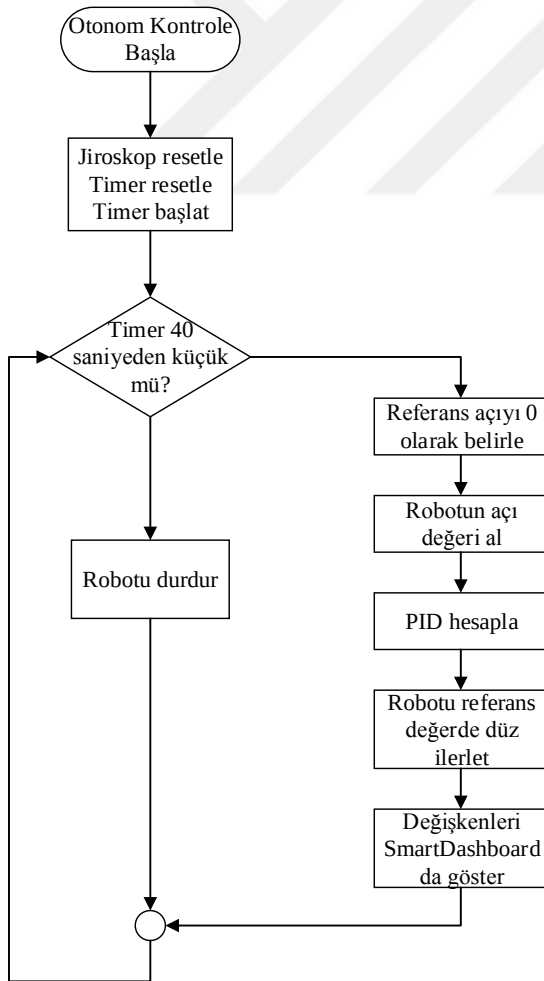
Şekil 4.31. Mobil robot elektronik sistemi

4.5. Mobil Robot Sistemi Yazılımı

Mobil robot sisteminde FRC yazılım bileşenleri kullanılarak otonom ve kullanıcı fonksiyonunda robot kodları yazılmış ve robotun çalışma durumları gerçek zamanlı izlenmiştir. Kodlar için Visual Studio Code programına eklenen WPILib kütüphaneleri ve FRC Java programlama rehberi kullanılmıştır (WPI 2019e).

Visual Studio Code programında Java dilinde yazılan kodlar yine aynı program ile derlenerek robotun işlemcisi roboRIO'ya yüklenmiştir. Robot kodlarında temel sınıf olarak TimedRobot kullanılarak periyodik yöntemlerin öngörülebilir bir zamanda(20ms) çağırılması sağlanmıştır. Robotik sistem bir bilgisayara kurulu FRC Driver Station programı ile otonom veya kumanda ile sürücü ile kontrol edilebilmektedir. Mobil robot sistemine ait kodlar EK 5'te verilmiştir.

4.5.1. Robotun otonom kontrolü

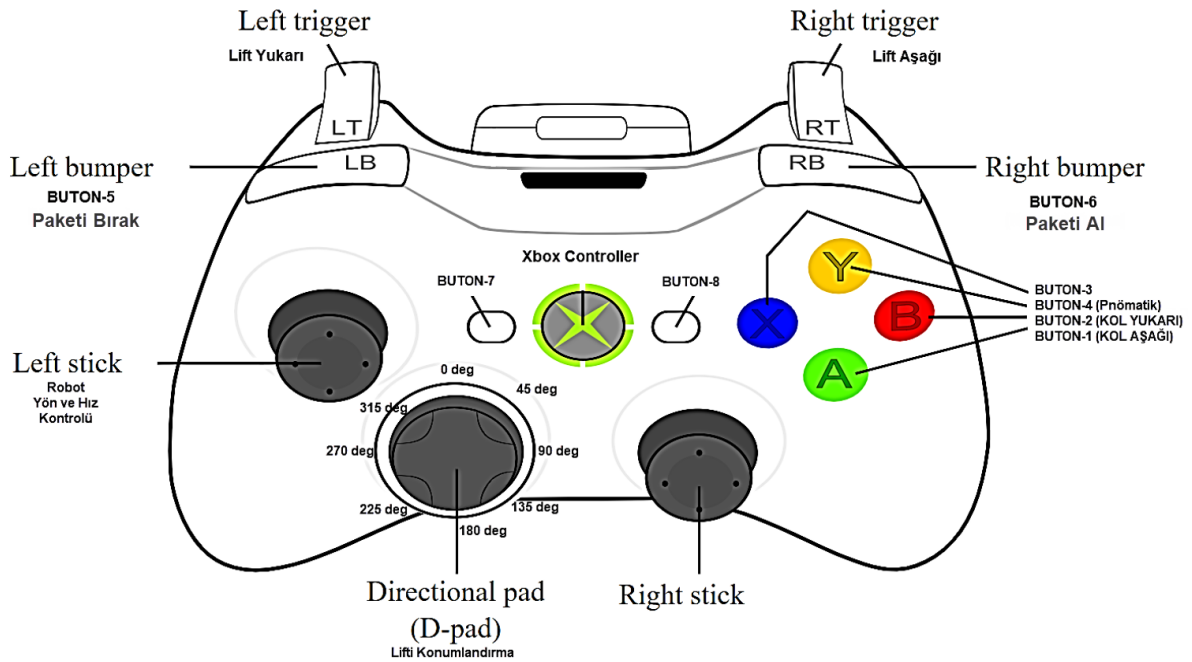


Şekil 4.32. Robot otonom kontrol akış diyagramı

Sistemin otonom çalışmadaki akış diyagramı Şekil 4.32’de gösterilmiştir. Otonom çalışmada robotun 40 saniye sabit bir yörüngede düz gitmesi sağlanmıştır. İlk olarak jiroskop ve timer resetlenmiştir. Timer başlatıldıktan sonra 40 saniye işlemlerin gerçekleşmesi için sorgulama yapılmıştır. Referans açı değeri 0° olarak belirlenmiştir. Robotun açısal konumu jiroskop ile alındıktan sonra PID kullanılarak robotun yönlendirilmesi yapılmıştır. Değişken durumları Smart Dashboard programında gösterilmiştir.

4.5.2. Robotun sürücü kontrolü

Robotik sistemin sürücü fonksiyonunda kontrolü için Şekil 4.33’te görülen Microsoft Xbox kablolu oyun kumandası kullanılmıştır. Tuşların ürettiği değerler FRC Smart Dashboard programı ile tespit edilmiştir. Mobil robot sisteminin sürücü kontrolüne ait akış diyagramı EK 6’da gösterilmiştir.



Şekil 4.33. Microsoft Xbox kablolu oyun kumandası (Anonim 2019a)

Tuş takımındaki Left Stick hareketli butonu ile robotun yön ve hız kontrolü yapılmıştır. Lift sistemi Left Trigger butonu ile yukarı çıkarken Right Trigger butonu ile aşağı inmektedir.

Bu iki analog tuş, basılı olmadığında “0” bilgisi üretirken basma durumuna göre en fazla “1” bilgisi üretir. Lift sistemi yukarı çıkarken maksimum hız %100’de aşağı inerken ise %30’da tutulmuştur. Bunun sebebi lift sistemi yukarı çıkarken sürtünme ve ağırlık gibi

faktörler yukarı çıkmaya engel olurken aşağı inerken ise ağırlık sistemi aşağı indirmektedir. Lift sistemi istenen konuma ulaştıktan sonra aşağıya kaymaması için kumandadan tuşa basılmadığı sürece yukarı yönde %20 bir güç uygulanmıştır.

Right bumper(RB) butonu ile tutucu sisteme ait motorlar içeri doğru çevrilerek paket alınmakta Left bumper(LB) butonu ile tutucu sisteme ait motorlar dışarı doğru çevrilerek paket bırakılmıştır.

A butonu robot kol bileğini aşağı indirirken B tuşu ise yukarı kaldırmaktadır.

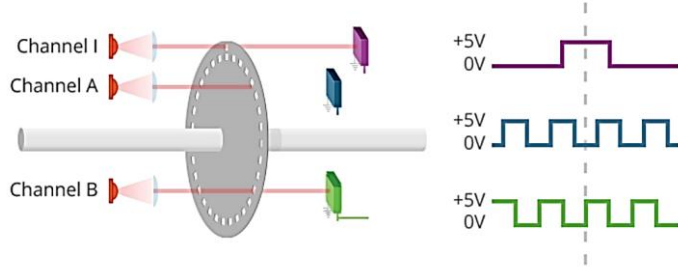
Y tuşu pnömatik sistemdeki solenoidi aktif yaparak pistonun ileri hareketini sağlayarak yapışan nesne paketini ayırır.

Directional Pad(D-Pad) tuşu ile lift sistemine ait referans değerler belirlenmektedir. Belirlenen referans değere göre PID kontrolör çalıştırılarak liftin istenen konuma ulaşması sağlanmıştır. Directional Pad tuşu ile 0°, 45°, 90°, 135°, 180°, 225°, 270° ve 315° gibi farklı açı değerleri gönderilebilir. Tasarlanan lift sisteminde 270° kademe 0, 180° kademe 1, 90° kademe 2 ve 0° kademe 3 için kullanılmıştır.

4.6. Mobil Robot Lift Sistemin Konum Bilgisini Elde Etme

Mobil robot lift sisteminin konum bilgisi lifte ait motor miline bağlanan enkoder tarafından ölçülmektedir. Enkoder, mekanik hareketi algılayarak hız, konum, mesafe ve yön bilgileri içeren elektrik sinyallerine çeviren bir sensördür.

Şekil 4.34'te döner bir enkoder diski mile sabitlenerek serbestçe milin etrafında dönmesi gösterilmiştir. Diskin bir tarafında verici sinyal kaynağı diğer tarafında bir alıcı vardır. Disk döndükçe vericiden çıkan sinyal alıcıya geçer veya engellenir. Sinyal alıcıya geçtiğinde çıkış darbesi üretilir. A ve B kanallarındaki üretilen sinyaller arasında 90° faz farkı vardır (Cui 2019).

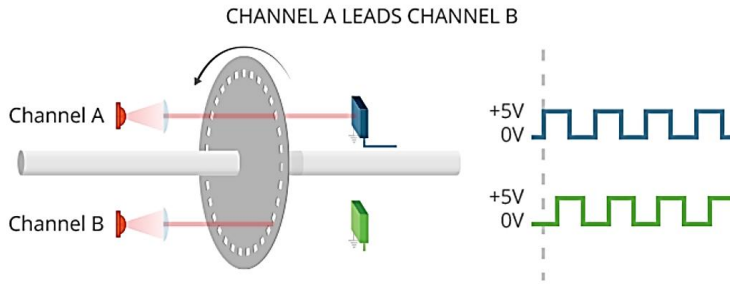


Şekil 4.34. Enkoder ile çıkış darbelerinin üretilmesi (Cui 2019)

4.6.1. Enkoder ile yön bilgisi elde etme

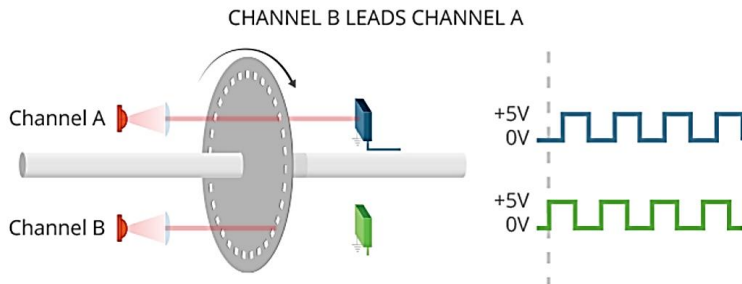
Motorun şaft yönünün tespiti bazı uygulamalar için gerekli olabilir. Örneğin mobil robotik sistemlerde motor şaftının dönme yönü algılanarak önceden belirlen referans değer için gerekli darbe elde edildiğinde gerekli işlemler başlatılabilir.

Şekil 4.35'te A kanalına sinyal B kanalından önce gelerek şaftın saat yönünün tersinde döndüğü belirlenir.



Şekil 4.35. Şaftın saat yönünün tersinde dönmesi (Cui 2019)

Şekil 4.36'da B kanalına sinyal A kanalından önce gelerek şaftın saat yönünde döndüğü belirlenir.



Şekil 4.36. Şaftın saat yönünde dönmesi (Cui 2019)

4.6.2. Enkoder ile Hız Bilgisini Elde Etme

Enkoder ile çıkış darbeleri belirli bir zaman aralığında sayılarak hız tespit edilebilir. Enkoder tarafından üretilen darbe sayısı ve devir başına gerekli darbe sayısı bilinirse dakikadaki devir hızı;

$$S = (C/PPR)/(T/60) \quad (4.1)$$

formülü ile hesaplanabilir. Formülde;

S : Dakikadaki devir sayısı (RPM)

C: Enkoder tarafından sayılan darbe sayısı

PPR: Devir başına gerekli darbe sayısı

T: Darbelerin sayıldığı zaman aralığı (saniye)

olarak ifade edilir.

Enkoder ile üretilen darbeler için sayma, zamanlama ve hesaplamalar gerçek zamanlı bir şekilde elektronik sistemler tarafından yapılarak hız bilgisi elde edilir (Cui 2019).

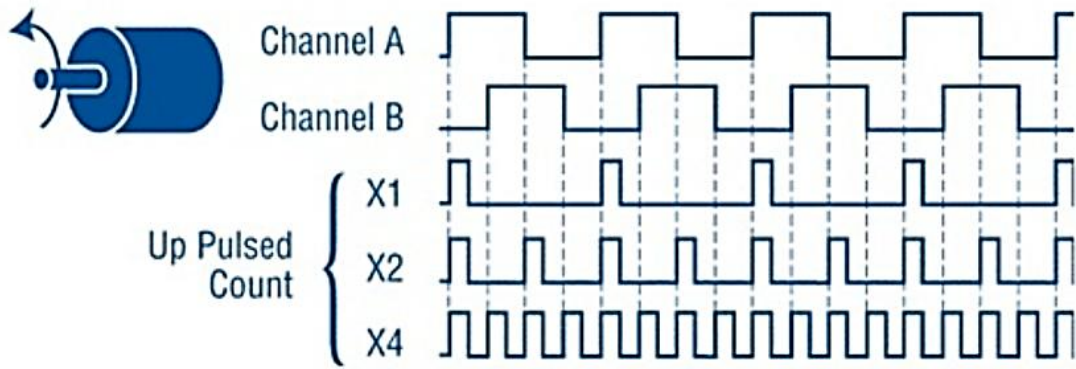
4.6.3. Enkoderler ile mesafe bilgisi elde etme

Enkoder ile üretilen darbe işaretleri sayılarak motor shaftının konum bilgisi veya shafta bağlı diğer mekanik bileşenlerin kat ettiği mesafeler tespit edilebilir. Motor mili saat yönünün tersine döndürüldüğünde enkoder darbe sayıcı bir arttırılır ve saat yönünde döndürüldüğünde ise bir çıkartılır. Darbe sayımı ile enkoder arasındaki mekaniksel ilişkiye dayanarak mesafe elde edilebilir. Birçok uygulamada kullanılan döner hareket kasnakları ve dişli çarklarının hareketi mekaniksel bileşenler tarafından doğrusal harekete dönüştürülür. Enkoderin PPR'si bilinirse doğrusal hareket hesaplanabilir (Cui 2019).

4.6.4. Enkoder Çözünürlüğü

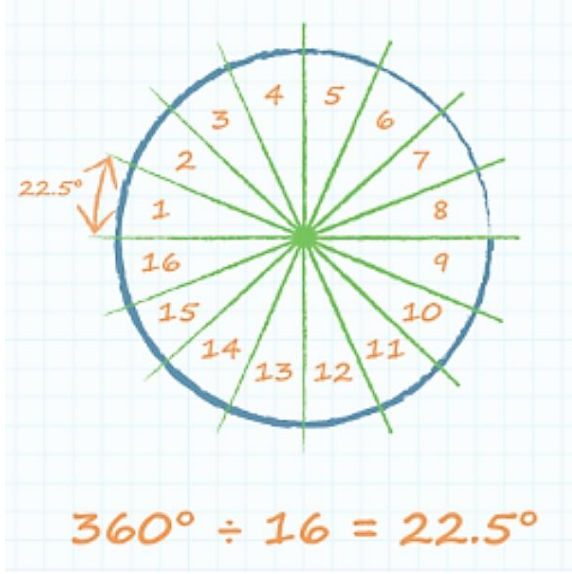
Tek kanallı kodlayıcılarda motorlardan kaynaklanan titreşimlerden kaynaklanan sayım hataları oluşabilir. Enkoder algılama yaptığı anda mekanik hareket durursa sayım hatası meydana gelebilir. Enkoder kod çözme işleminde bir döngüde durum değişiklikleri sayılır. Şekil 4.37'de enkoder ölçek faktörleri ile üretilen darbelerin sayımı gösterilmiştir. X1 ile ifade

edilen çıkış sinyalinde sadece A kanalının yükselen kenarlarının sayısı sayılmıştır ve çözünürlük düşüktür. X2 ile ifade edilen çıkışta A kanalındaki darbe sinyalinin hem yükselen hem alçalan kenarlarının sayısı sayılmıştır ve çözünürlük iki katına(x2) çıkmıştır. X3 ile ifade edilen çıkışta hem A kanalındaki hem de B kanalındaki darbe sinyalinin hem yükselen hem alçalan kenarlarının sayısı sayılmıştır ve çözünürlük dört katına(x4) çıkmıştır. Yani 1 kare dalga döngüsü için enkoderden 4 darbe elde edebilir. Her döngüde 4 sayma işlemiyle hassasiyet 4 katına çıkarılmıştır. Böylece çift yönlü konum algılama ve uzunluk ölçme uygulamalarında doğruluk artırılır (Dynapar 2019).



Şekil 4.37. Farklı ölçek faktörleri ile darbe sayımı

Bir kodlayıcının kat ettiği mesafeyi veya hangi hızda döndüğünü belirlemek için enkoder çözünürlüğünün bilinmesi gerekir. Çözünürlük enkoderin kaç eşit parçaya bölündüğünü gösterir. Enkoder üreticileri tarafından devir başına darbe sayısı(PPR) belirlenerek çözünürlük ayarlanır. Çözünürlüğün yüksek olması daha hassas ölçüm yapılmasına olanak verir. Devir başına sayım ise CPR olarak ifade edilir ve 4X kod çözme işlemi için PPR'nin 4 katıdır. Çözünürlük kullanılarak her bir darbe periyodunun kaç mekanik dereceye eşit olduğu hesaplanabilir. Şekil 4.38'de PPR değeri 16 olan bir enkoderin her darbede kaç derece değiştiği görülebilir (Cui 2019a).



Şekil 4.38. Devir başına gerekli darbe sayısı ve darbe derecesi (Cui 2019a)

4.6.5. Enkoder Çeşitleri

a) Optik enkoder: En hassas uygulamalarda hareketi kontrol etmek için elektronik cihazlarda kullanılır. Çok yaygın bir kullanımı vardır. Kızılötesi LED ve fototransistör kullanılarak çıkış kodu üretir. Mevcut enkoder çeşitlerinden en yaygın olan türdür.

b) Manyetik enkoder: Manyetik akı değişiklerini algılayarak çıkış kodu üretir. Genellikle olumsuz koşullarda kullanılır. Aşırı sıcaklık, yüksek nem ve partikül veya sıvılara maruz kalan uygulamalarda kullanılır.

c) Fiber optik: Lazer ve fototransistör kullanılarak çıktı kodu üretir. Yanıcı gazların(metan, propan vb.) bulunduğu patlama oluşturabilecek uygulamalarda kullanılır.

d) Kapasitif enkoder: Yüksek frekanslı bir referans sinyali kullanarak kapasitedeki değişikleri tespit ederek çıkış kodu üretir. Patlama oluşturabilecek uygulamalarda kullanılmaz. Manyetik enkoder ile benzer çevresel faktörlere dayanabilir ve optik enkoderden daha iyi performans verebilir (Cui 2019).

4.6.6. Mobil robotik sistemde enkoder ayarları

Mobil robotik sistemde kullanılan CIMcoder için devir başına gerekli darbe sayısı (PPR) 20 çözünürlüğüne sahiptir. Ölçek faktörü enkoderin kod çözme işini ne kadar hassaslıkla yaptığını gösterir. Tasarlanan sistemde ölçek faktörü X4 olarak kullanılmıştır ve devir başına üretilen ölçüm sayısı(CPR);

$$CPR = 4. PPR = 4.20 = 80 \quad (4.2)$$

olarak elde edilmiştir.

Darbe başına alınan yol (DPP) her darbede şaftın kat ettiği mesafeyi ifade eder. Eğer şaft bir dişli sisteme takılırsa bu değeri ayarlamak için dişlinin çapını, devir başına darbe sayısını ve sistemdeki tüm aktarma organlarını bilmek gerekir. Aktarma organlarını hesaba katmadan motor milinin çevresi($2\pi r$) ve devir başına gerekli darbe sayısı kullanılarak darbe başına alınan yol(DPP);

$$DPP = \text{Motor milinin çevresi} / PPR \quad (4.3)$$

Üretilen darbe sayısı(C);

$$C = L. SF / DPP \quad (4.4)$$

Enkoder tarafından üretilen darbe sayısına göre motor şaftının aldığı yol(L);

$$L = C. DPP / SF \quad (4.5)$$

Motor milinin tur sayısı;

$$\text{Motor milinin tur sayısı} = C / (PPR. SF) \quad (4.6)$$

Dişli kutusunun tur sayısı;

$$\text{Dişli kutusunun tur sayısı} = \text{Motor milinin tur sayısı} / \text{Dişli kutusu oranı} \quad (4.7)$$

Liftin Aldığı yol;

$$\text{Liftin aldığı yol} = \text{Dişlinin tur sayısı} \times \text{Kasnağın çevresi} \quad (4.8)$$

formülleri ile hesaplanabilir.

Üsteki formüllerde;

PPR: Devir başına gerekli darbe sayısı (pulses per revolution)

CPR: Devir başına üretilen ölçüm sayısı (Counts per Revolution)

SF: Ölçek faktörü (ScaleFactor)

DPP: Darbe başına alınan mesafe (Distance Per Pulse) (cm)

C: Üretilen darbe sayısı (Count)

L: Motor şaftının aldığı yol (cm)

$2\pi r$: Motor milinin çevresi (cm)

olarak ifade edilmiştir(Cui 2019).

Tasarımı yapılan mobil robot lift sistemi kasnak sistemi ile iki kademeli olarak tasarlanmıştır. Lift 1cm yol aldığı anda liftin üst kademesine bağlı robot kolu 2cm yol almaktadır. Liftin en alt mesafesinin yerden yüksekliği 20cm'dir. Örneğin lift 40cm yol aldığı anda robot kolu 80cm yol almakta böylece robot kolu yerden 100cm yüksekliğe çıkmış olmaktadır.

Tasarımda kullanılan CIM motor şaftının yarıçapı 0,398cm ve çevresi 2,496cm'dir. Dişli kutusu oranı 1:10,71'dir. Dişli kutusu miline bağlı kasnağın çapı 27cm'dir. Bu sebeple dişli kutusu 1,49 tur döndüğünde lift ~40cm yukarı kalmaktadır.

Yukarıdaki formüller kullanılarak aşağıdaki Çizelge 4.2'de oluşturulmuştur.

Çizelge 4.2. Mobil robot lift sistemi konum bilgisi hesaplamaları

Ölçek faktörü (SF)	Devir başına gerekli darbe sayısı (PPR)	Üretilen darbe sayısı (C)	Darbe başına alınan yol (DPP)	Motor şaftının aldığı yol	Motorun Tur Sayısı	Dişli kutusunun tur Sayısı	Liftin aldığı yol
4	20	80	0,125	2,50	1	0,09	2,52
4	20	857	0,125	26,78	10,7125	1,00	27,01
4	20	640	0,125	20,00	8	0,75	20,17
4	20	1280	0,125	40,00	16	1,49	40,34
4	20	1920	0,125	60,00	24	2,24	60,50

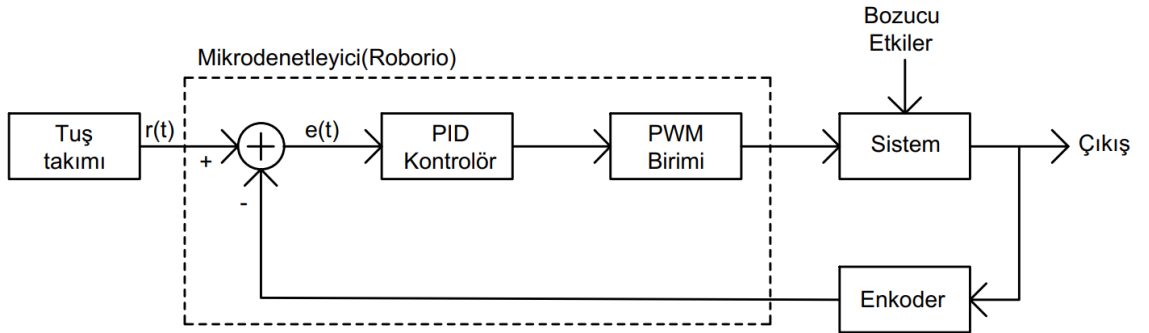
5. ROBOTİK SİSTEM YÖRÜNGE VE HAREKET KONTROLÜ DENEYLERİ

Mobil robotun lift sistemini istenen konuma getirme ve robotun düz bir yörüngede hareketi için PID kontrolör işlemleri gerçekleştirilmiştir. P, PD, PI ve PID kontrolör etkilerini görmek için gerçek zamanlı grafikler Smart Dashboard programı ile çizdirilmiş ve test sonuçları değerlendirilmiştir.

5.1. Mobil Robotun Lift Sistemini Konumlandırma

Mobil robot için tasarlanan lift sistemi ile nesneler yerden alınıp istenen referans değere tek bir tuş ile bırakılması sağlanmıştır. Lift sisteminin belirli bir konuma hatasız konumlanması için kontrolör tasarımı gereklidir. Sistemin kararlı olması, sürekli hal hatasının sifıra yakın olması, aşımın düşük olması, yükselme ve yerleşme zamanının düşük olması ayrıca diğer sistemi bozucu etkenlerden (gürültü vb.) etkilenmemesi amaçlanmıştır.

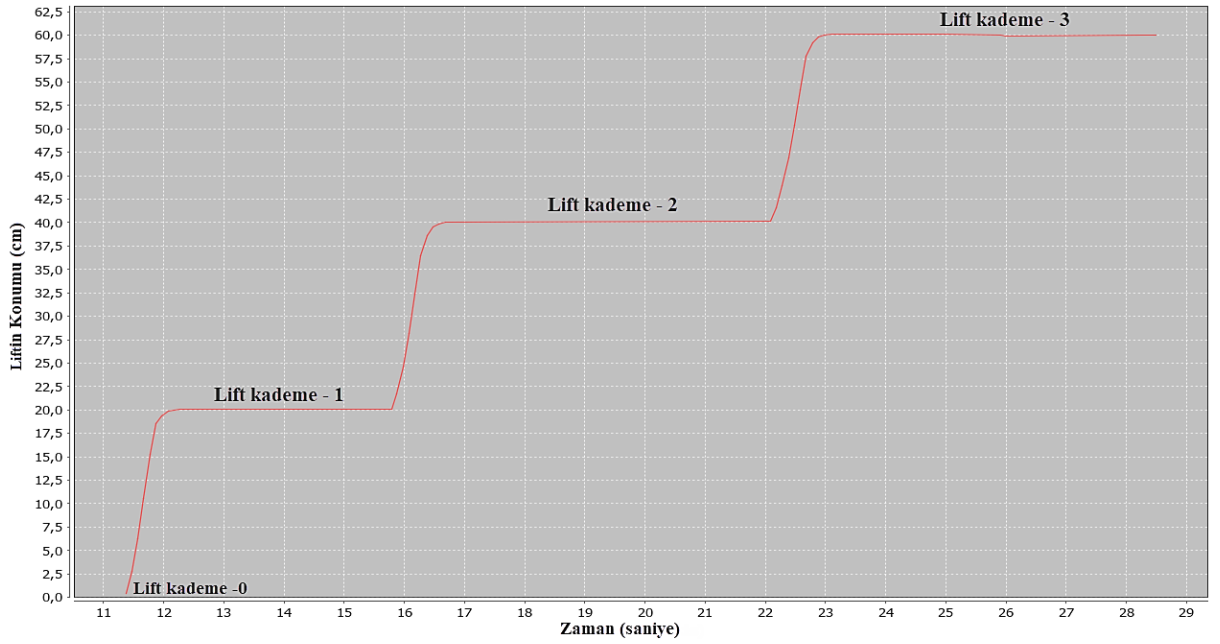
Tasarımı yapılan kontrol sisteminin blok diyagramı Şekil 5.1’de görülmektedir. Tuş takımı ile referans değerler girilir. Sistemde motor miline bağlı bir enkoder ile çıkış konumu ölçülür. Çıkış konumu ile referans konum karşılaştırılarak kontrolör tarafından motor referans konuma getirilir. Mikrodenetleyici ile konum verisi hız verisine dönüştürülerek motor kontrol edilir. Sürtünme, lift sisteminin ağırlığı, lifte bağlı kolun ağırlığı gibi diğer bozucu etken ve gürültüler blok diyagramda bozucu etkiler olarak ifade edilmiştir.



Şekil 5.1. Lift sistemi için kontrol sistemi blok diyagramı

Robot lift sistemi için 2 adet CIM motor dişli kutusuna bağlanmıştır. Motorlar 2 adet SPARK Motor sürücüsü ile kontrol edilmektedir. Motor sürücülere uygulanan PWM sinyali ile liftin aşağı yukarı hareketleri sağlanmıştır. Motora bağlı enkoder ile motorun konum bilgisi mikrodenetleyiciye (roboRIO) aktarılmıştır.

Kumanda ile gönderilen referans değerlere göre lift sisteminin konumları Şekil 5.2’de gösterilmiştir. Lift sistemi tasarımı 4 kademeli çalışacak şekilde düzenlenmiştir. En alt kademede referans değer 0 olarak belirlenmiştir. 1, 2 ve 3. kademeler için referans değerler sırasıyla 20cm, 40cm ve 60cm olarak belirlenmiştir. Bu referans değerler kullanıcı tarafından kumanda ile mikrodenetleyiciye gönderilir. Lift sistemi için tasarımı yapılan kontrolöre liftin konum bilgileri enkoder tarafından geri besleme yapılır. Kontrolör kumanda ile seçilen referans değere lift sistemini konumlandırır.



Şekil 5.2. Kumanda ile lift sisteminin kontrolü

Lift sistemi PID kontrol işlemi için JAVA programlama dilinde yazılan kod Çizelge 5.1’de verilmiştir. Program kodlarında pid_lift() fonksiyonu çağrılarak lift sistemi kontrolörü çalıştırılır. Programda ilk olarak PID parametrelerini ifade eden oransal kontrol kazancı K_p , integral kontrol kazancı K_i ve türev kontrol kazancı K_d ayarlanmıştır. Lifte bağlı enkoder ile okunan değerler `olcun_konum` değişkenine aktarılmıştır. Tuş takımı ile belirlenen referans değer ile liftin konumu arasındaki fark alınarak hata miktarı hesaplanmıştır. Zamana bağlı olarak hata miktarı ile son hata arasındaki fark türev değişkenine aktarılmıştır. Zamana bağlı olarak hatalar toplanarak toplam_hata değişkenine aktarılmıştır. Hata değeri son hata değişkenine `son_hata=hata;` komutu ile aktarılmıştır. Büyük hata değerleri integral toplamını çok artıracığı için integral işlemi referans değere 7cm kala devreye alınmıştır.

Oransal kontrol çıkışı için `p_out=hata*Kp;` komutu, integral kontrol çıkışı `i_out=toplam_hata*Ki;` komutu, türev kontrol çıkışı `d_out=tuv*tuv*Kd;` komutu ile

hesaplanmıştır. PID kontrol çıkışı sistem çıkışı olarak lift sistemine bağlı motorların hızını belirlemektedir ve $pid_out = p_out + i_out + d_out$; komutu ile hesaplanmıştır.

PID çıkışlarına limit konularak motorlara uygulanacak maksimum PWM sinyallerine sınırlama getirilmiştir. Lift yukarı çıkarken maksimum PWM sinyali “1” olarak belirlenmiştir. Lifti aşağı indirmek için PWM sinyali “-0.1” ile sınırlandırılmıştır. Aşağı indirme değişkeninin düşük olmasının nedeni liftin ve lifte bağlı kolun ağırlığıdır. Hesaplanan PID

Çizelge 5.1. Lift sistemi PID kontrolör kodu

```
public void pid_lift()
{
    Kp = 0.5; Ki = 0.5; Kd = 0.05; // kontrolör parametreleri belirlendi
    olculen_konum= getDistanceTraveled();//lifte bağlı enkoderden veri okundu
    hata = referans_konum - olculen_konum; //hata hesaplandı
    turev=(hata-son_hata)/0.02; //türev işlemi yapıldı
    toplam_hata= toplam_hata + hata*0.02; //integral toplamı hesaplandı
    son_hata= hata; //hata içeriği son_hata olarak belirlendi
    if(hata>7) toplam_hata=0; //integral toplamı sıfırlandı

    p_out=hata*Kp; //oransal çıkış hesaplandı
    i_out=toplam_hata*Ki; //integral çıkış hesaplandı
    d_out=turev*Kd; //türev çıkış hesaplandı
    pid_out = p_out + i_out + d_out; //PID çıkış hesaplandı

    if(pid_out >= 1) pid_out= 1; //PID çıkış sınırlandı
    else if(pid_out < - 0.1) pid_out= - 0.1; //PID çıkış sınırlandı

    lift_motor1.set(pid_out); //PID değeri lift motoruna gönderildi
    lift_motor2.set(pid_out); //PID değeri lift motoruna gönderildi
}
```

çıkış verisi lifte bağlı motor sürücülere gönderilerek liftin hareketi sağlanır.

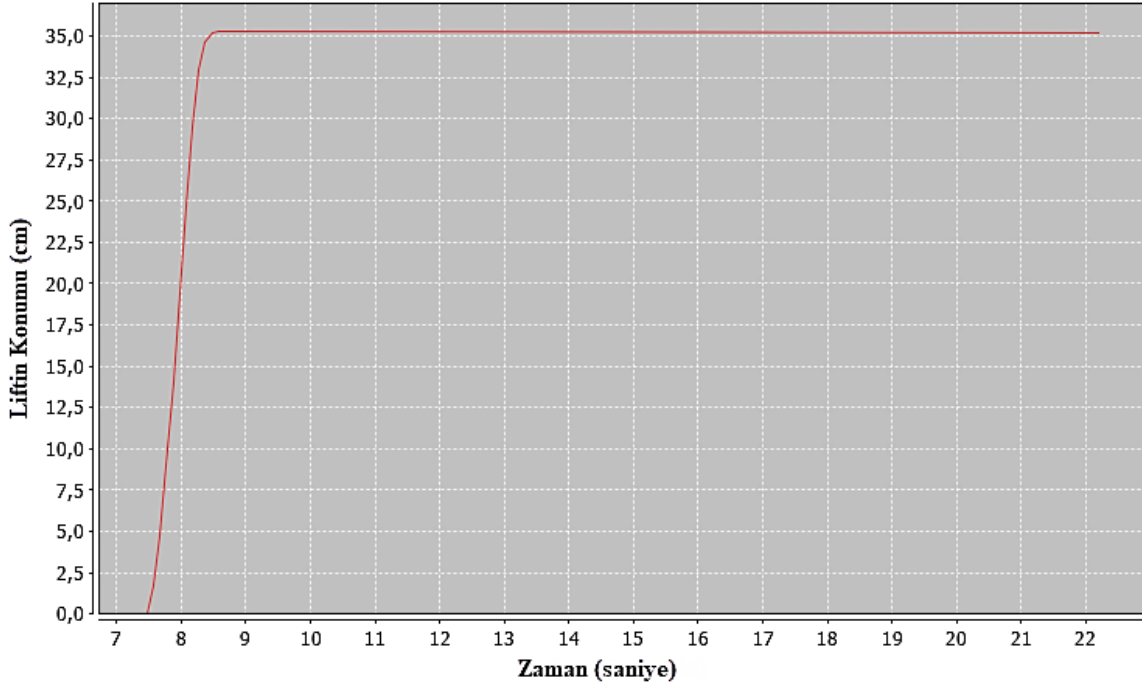
Lift sistemi için tasarlanan PID kontrolör parametreleri değiştirilerek P, PD, PI ve PID kontrolör etkileri aşağıdaki başlıklarda incelenmiştir. Test amacıyla yapılan deneylerde referans lift sistemi konumu 40cm olarak ayarlanmıştır.

5.1.1. Lift sistemi P kontrolör tasarımı

Oransal (P) kontrolde çıkış, hata ile orantılı olarak değişmektedir. Hata miktarı arttıkça çıkış artmakta ve hata miktarı azaldıkça çıkış azalmaktadır. Oransal kontrol işlemi için

programda integral ve türev kazançları sıfırlanmıştır. Hata miktarı ile K_p kazancı çarpılarak oransal çıkış hesaplanmıştır. Referans değer kumanda ile 40 olarak seçilmiştir. Bu referans değere göre lift sisteminin kademe 2'ye ulaşması istenmiştir.

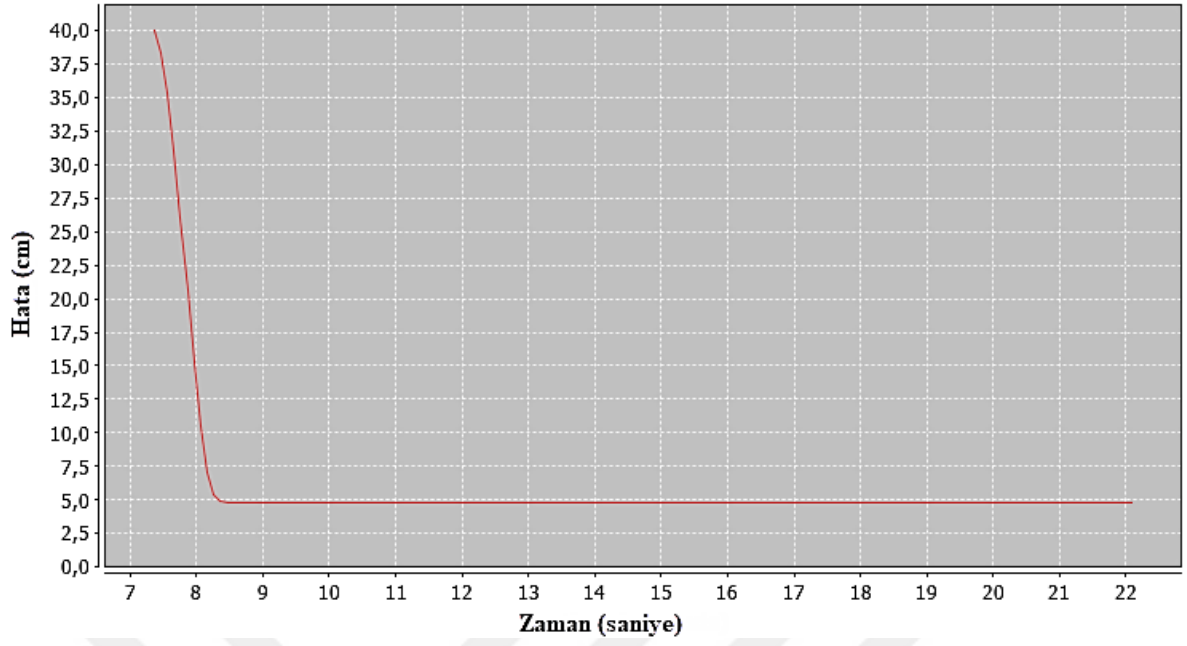
Oransal kontrol kazancı $K_p=0,05$ için konum-zaman grafiği Şekil 5.3'te ve hata-zaman grafiği Şekil 5.4'te gösterilmiştir. Konum-zaman grafiğinden liftin konumunun referans değer 40'a ulaşmadığı görülebilir.



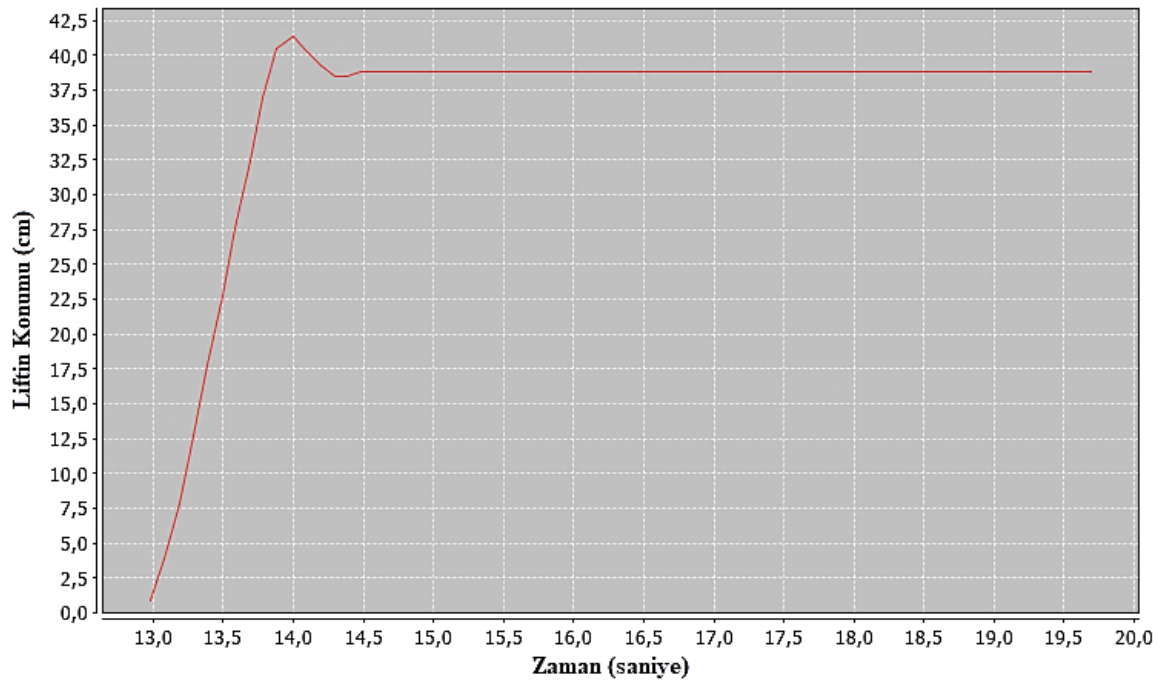
Şekil 5.3. P Kontrol konum-zaman eğrisi ($K_p=0,05$)

Şekil 5.4'te verilen hata-zaman grafiğinden kalıcı hal hatası ~ 5 cm olarak görülebilir. Kontrolör çıkışı hata miktarına bağlıdır. Liftin konumu referans değere yaklaştıkça hata azalır ve oransal kontrolör lift sistemini referans değere ulaştıramaz ve bu durum kalıcı hal hatasına sebep olur. Ayrıca oransal kontrolde dış etkiler kalıcı hataları daha da arttırabilir. Örneğin akü geriliminin düşmesi veya sürtünmenin artması gibi etkenler liftin konumunu referans değerden daha da uzaklaştırabilmektedir.

Kalıcı hal hatasını düşürmek için $K_p=0,25$ yapılarak Şekil 5.5'te verilen konum-zaman grafiği elde edilmiştir. Grafik analiz edildiğinde K_p kazancının artmasının aşımaya sebep olduğu ve kalıcı hal hatası bir miktar azalttığı görülebilir.



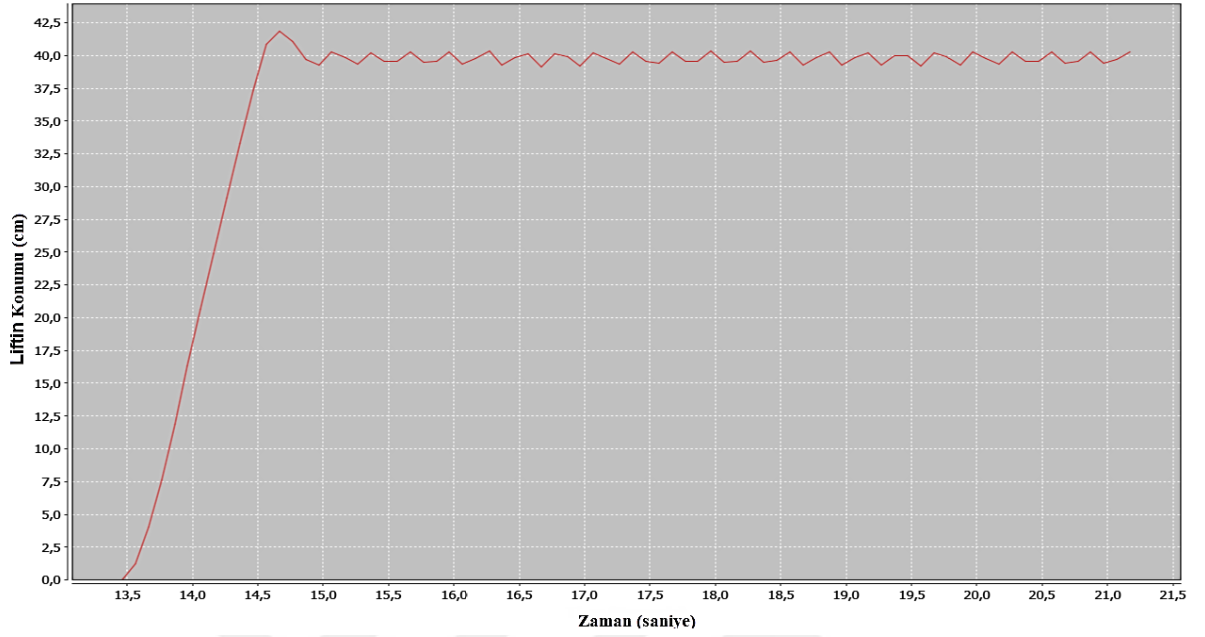
Şekil 5.4. P kontrol Hata-zaman eğrisi ($K_p=0,05$)



Şekil 5.5. P kontrol konum-zaman eğrisi ($K_p=0,25$)

$K_p=1$ yapıldığında ise liftin konum-zaman grafiği Şekil 5.6'daki gibi elde edilmiştir. Grafik incelendiğinde lift konumunun referans değeri etrafında sürekli olarak salınım yaptığı görülmüştür. Liftin sürekli aşağı yukarı hareket etmesine sebep olduğu için K_p 'nin belli

değerin üzerinde arttırılması mobil robotik sistemde mekaniksel ve elektriksel birçok problemlere sebep olabileceği belirlenmiştir.

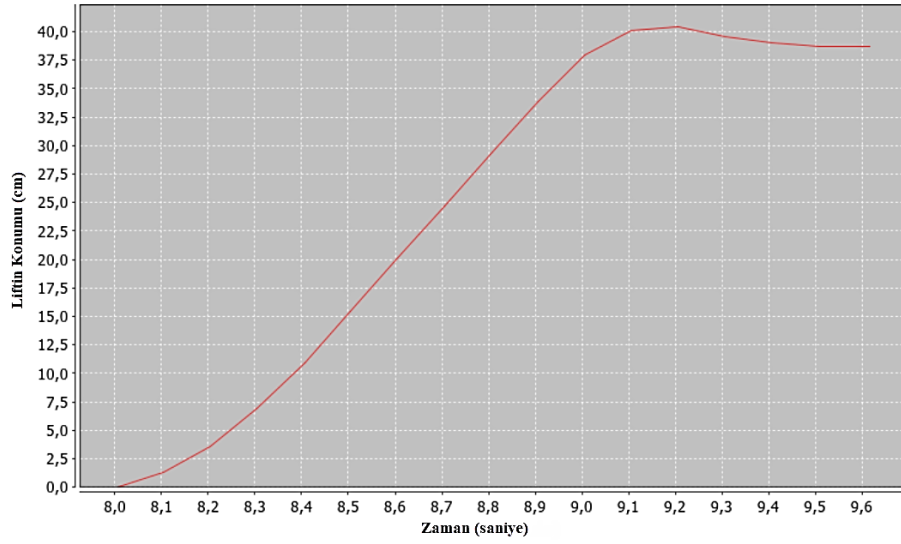


Şekil 5.6. P kontrol konum-zaman eğrisi ($K_p=1$)

Oransal kontrol ile yapılan testlerde robot lift sisteminde K_p 'nin düşük olması durumunda yüksek miktarda kalıcı hal hatasına sebep olduğu görülmüştür. K_p arttırıldığında ise aşım oluşmuş ve kalıcı hal hatası bir miktar düzelmiştir. K_p 'nin çok büyük belirlenmesi yerleşme zamanını uzatmakta hatta lift sisteminde salınımlara sebep olarak robotik sistemde özellikle mekaniksel problemler oluşturabileceği görülmüştür.

5.1.2. Lift sistemi PD kontrolör tasarımı

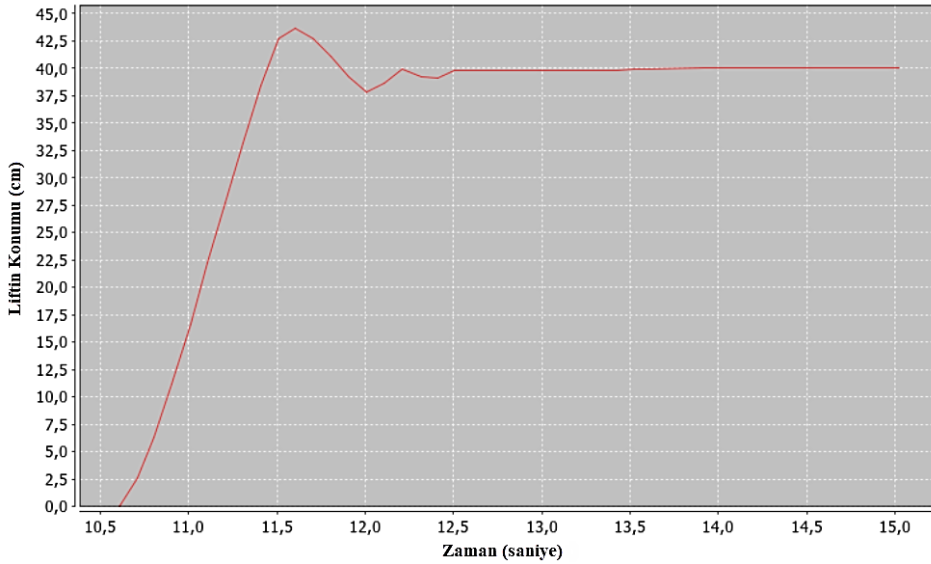
Oransal kontrolöre türev(D) eklenerek PD kontrolör gerçekleştirilmiştir. $K_p=0,25$; $K_i=0$; $K_d=0,015$; olarak belirlendiğinde Şekil 5.7'de verilen konum-zaman grafiği elde edilmiştir. Burada türevin etkisi ile sisteme öngörü kazandırılarak aşım engellenmiştir. PD kontrolörün kalıcı hal hatasını etkilemediği görülmüştür. Lift sistemindeki amacımız istenen referans değere ulaşmak olduğu için PD kontrolör mobil robot lift sistemi uygulamaları için tek başına yetersizdir.



Şekil 5.7. PD kontrol konum-zaman eğrisi

5.1.3. Lift sistemi PI kontrolör tasarımı

Oransal kontroldeki kalıcı hal hatasını sıfırlamak için PI kontrol işlemi yapılmıştır. Bu kontrolde oransal kontrol ve hatanın zamanla integralleri toplanıp çıkış üretilmiştir. PI kontrol parametreleri $K_p=0,45$; $K_i=1,1$; $K_d=0$; olarak belirlendiğinde Şekil 5.8'deki gibi konum zaman grafiği elde edilmiştir.



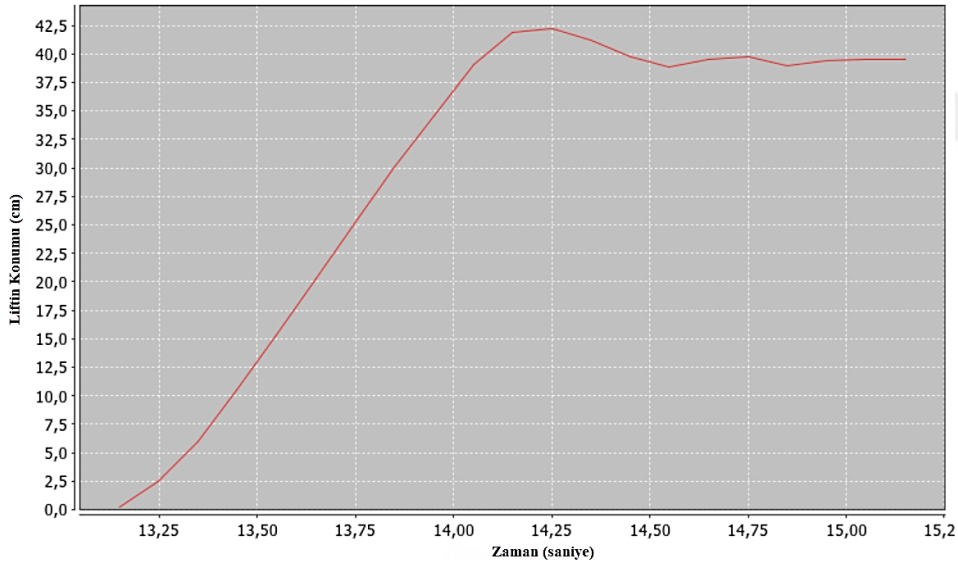
Şekil 5.8. PI kontrol konum-zaman eğrisi

Tasarlanan sistemde integral kontrol, referans değere 7cm kala devreye alınmıştır. Bunun sebebi integral yığılmasını önlemek içindir. Çünkü integral toplamının çok artması kontrolörü başarısız kılacaktır. Tasarımı yapılan PI kontrolde hata miktarı 7'ye düşüne kadar

oransal kontrol ile lift sistemi hareket ettirilmiştir. Hata miktarının 7'nin altına düştüğünde integral işlemi devreye alınmıştır. Konum-zaman grafiğinden görüldüğü gibi lift sisteminde bir miktar aşım oluşmakta ve salınım yaparak referans değere gelmektedir. Oransal kontrole eklenen integral etkisinin kalıcı hal hatasını giderdiği ve robotik lift sistemlerinde hatayı sıfırlamak için kullanılabileceği görülmüştür.

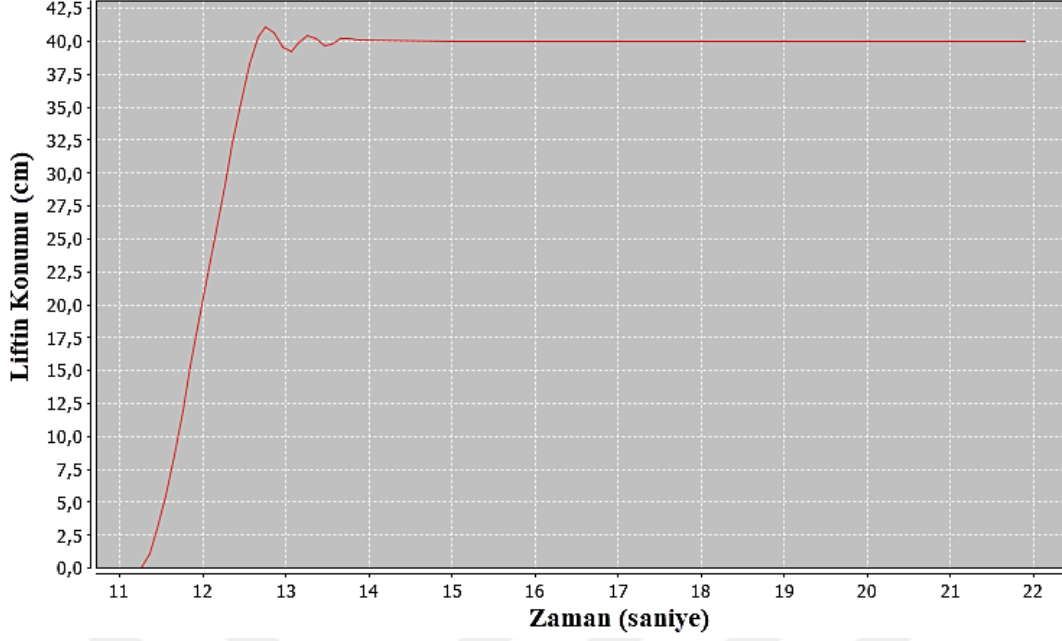
5.1.4. Lift sistemi PID kontrolör tasarımı

PID kontrol tasarımına öncelikle P kontrolör tasarımı yapılarak başlanmıştır. P kontrolde hatayı azaltmak için K_p kazancı 0,5 seçilerek konum-zaman Şekil 5.9'daki gibi elde edilmiştir. Referans değer 40 olduğu için mobil robot lift sistemi konumu 0'dan 40 ya doğru ilerlerken hata değeri de 40'den 0'a doğru azalmaktadır. Hata miktarı düştükçe P kontrolör çıkışı (Pout) zayıflamakta ve lift sistemini hareket ettirmede yetersiz kalmaktadır. Konum-zaman grafiğinden görüldüğü gibi geçici hal durumu geçtikten sonraki lift konumu referans değerde kalamamış ve sistemde sürekli bir hata oluşmuştur.



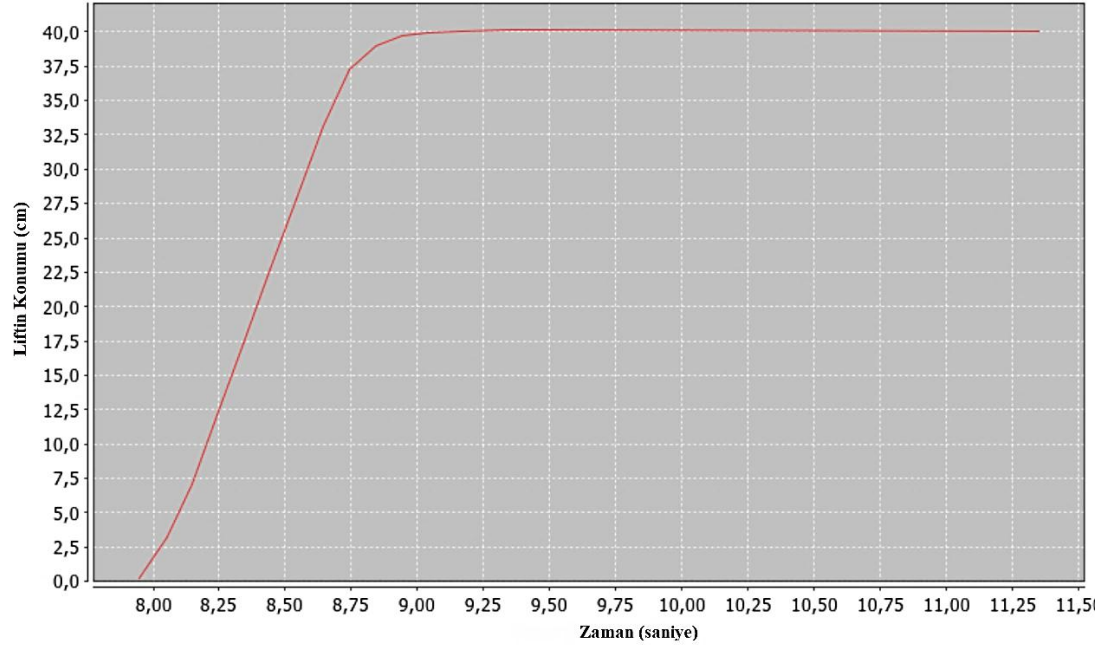
Şekil 5.9. P kontrol konum-zaman eğrisi ($K_p=0,5$)

Oransal kontroldeki kalıcı hal hatasını engellemek için sisteme integral ilave edilerek PI kontrolör gerçekleştirilmiştir (Şekil 5.10). İntegral yığılmasını önlemek için referansa değere 7cm kala integral devreye alınarak kalıcı hal hatası giderilmiştir. Sistemin kalıcı durumunu sağlamada integral çıkışı aktif olmuştur.



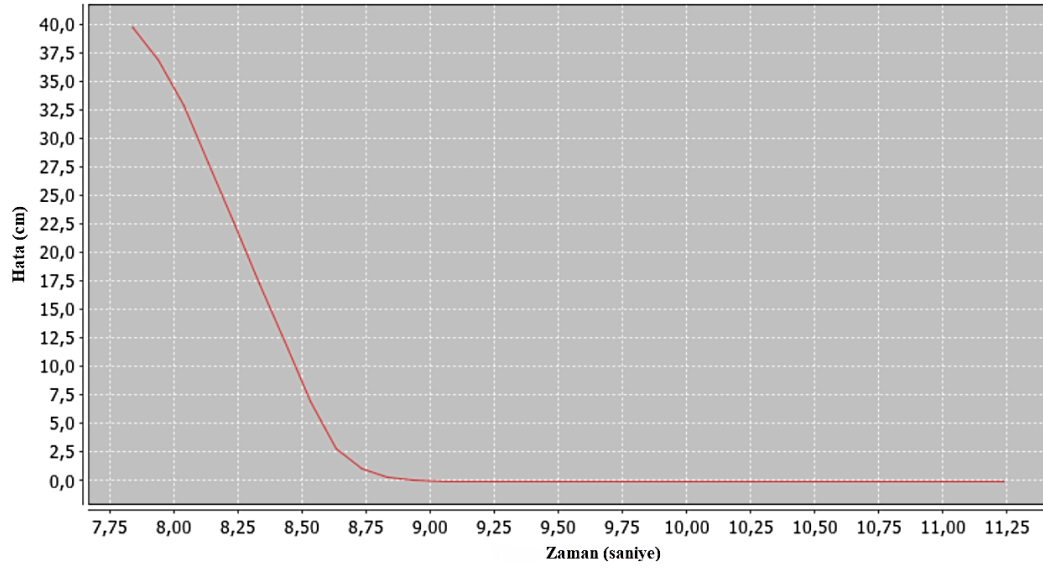
Şekil 5.10. PI kontrol konum-zaman eğrisi ($K_p=0,5$; $K_i=0,5$ ve $K_d=0$)

PI kontrol işlemi ile kalıcı hal hatası düzelmiş ancak sistemin geçici yanıtında salınımlar söz konusudur. Sistemin geçici yanıtını düzeltmek için PI kontrolöre türev eklenerek PID kontrol gerçekleştirilmiştir. PID kazançları $K_p=0,5$; $K_i=0,5$ ve $K_d=0,05$ olarak belirlendiğinde lift sistemin konum zaman grafiği Şekil 5.11’de gösterilmiştir.

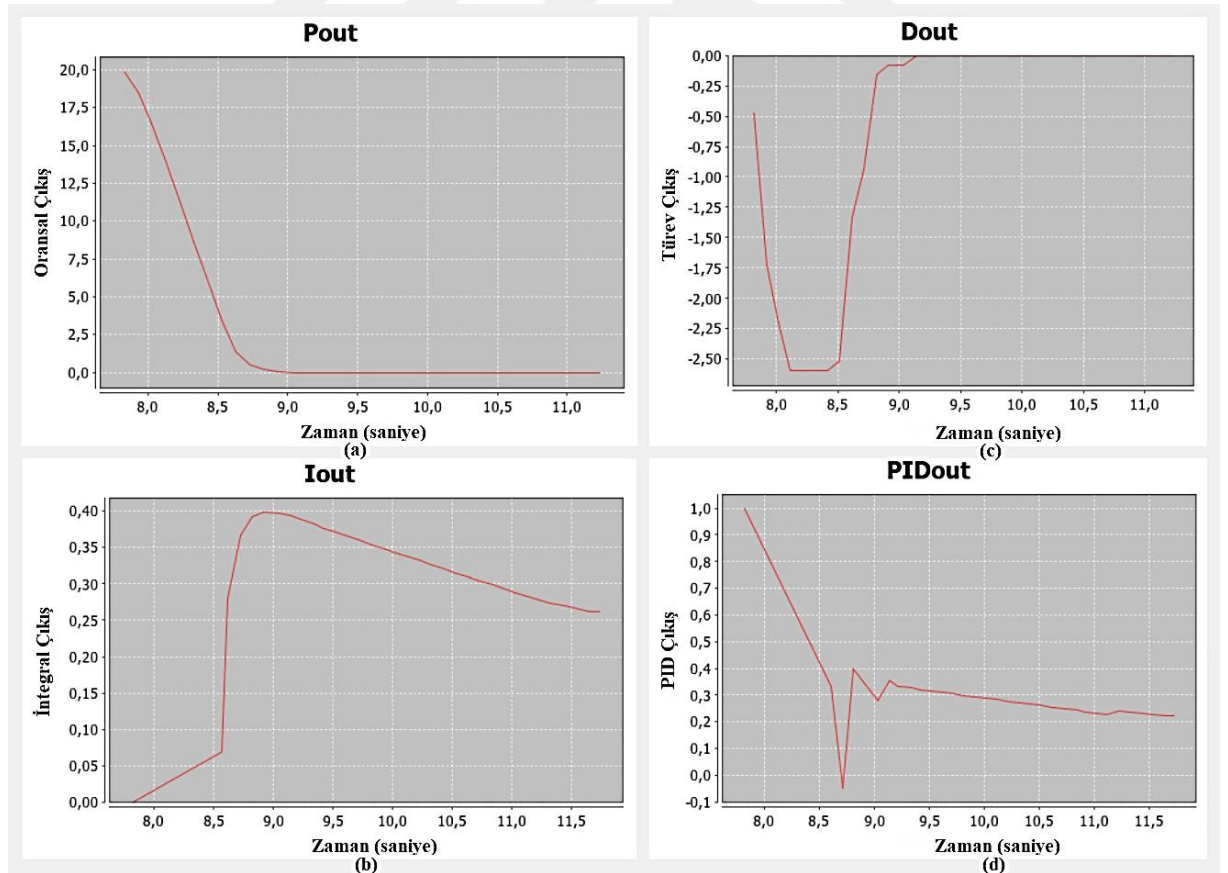


Şekil 5.11. PID kontrol konum-zaman eğrisi ($K_p=0,5$; $K_i=0,5$ ve $K_d=0,05$)

Şekil 5.12’de verilen hata zaman grafiğinde görüldüğü gibi ~1 saniyede hata 0’a düşürülerek lift sistemi konumu referans değere sabitlenmiştir.



Şekil 5.12. PID kontrol hata-zaman eğrisi($K_p=0.5$; $K_i=0.5$ ve $K_d=0.05$)

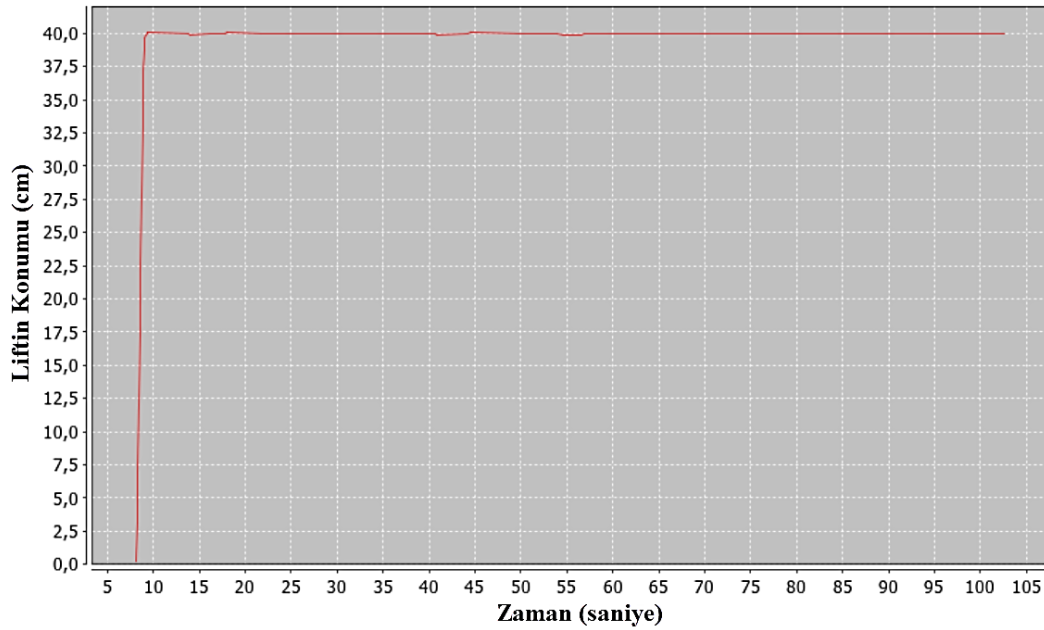


Şekil 5.13. PID kontrolör çıkış eğrileri ($K_p=0.5$; $K_i=0.5$ ve $K_d=0.05$) (a) oransal kontrol çıkışı (b) integral kontrol çıkışı (c) türev kontrol çıkışı (d) PID kontrolör çıkışı

Şekil 5.13'teki grafiklerde oransal çıkış(Pout), integral çıkış(Iout) ve türev çıkışın(Dout) tasarlanan PID kontrolör çıkışına(PIDout) etkileri gösterilmiştir. Grafiklerden oransal çıkışın hata miktarını azalttığı, integral çıkışın liftin referans değere ulaşip bu değerde kalmasını sağladığı ve türev çıkışın sisteme öngörü kazandırarak kontrolör çıkışına negatif etki vererek aşımı engellediği görülebilir. Sistemin kalıcı durumunda sadece integral çıkış aktif olmuş geçici durumunda ise oransal ve integral kontrolör hataları azaltmış türev kontrol ise aşimleri önlemiştir.

Lift sistemin ~100 saniye boyunca referans konum 40'da kalması için PID kontrolör etkileri aşağıda incelenmiştir.

Şekil 5.14'teki konum-zaman grafiğinde lift konumunun referans konum olan 40cm'de sabit kaldığı gösterilmiştir.

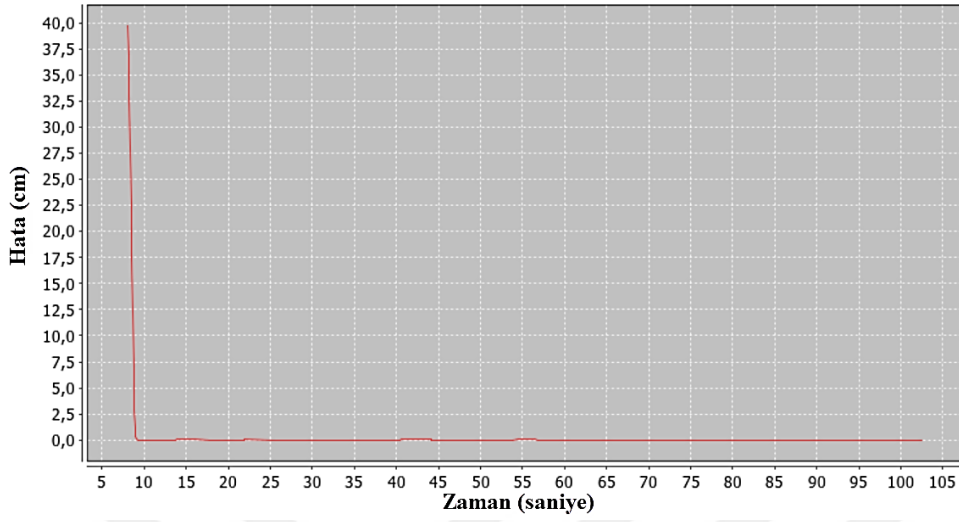


Şekil 5.14. PID kontrol konum-zaman eğrisi (~100s analiz)

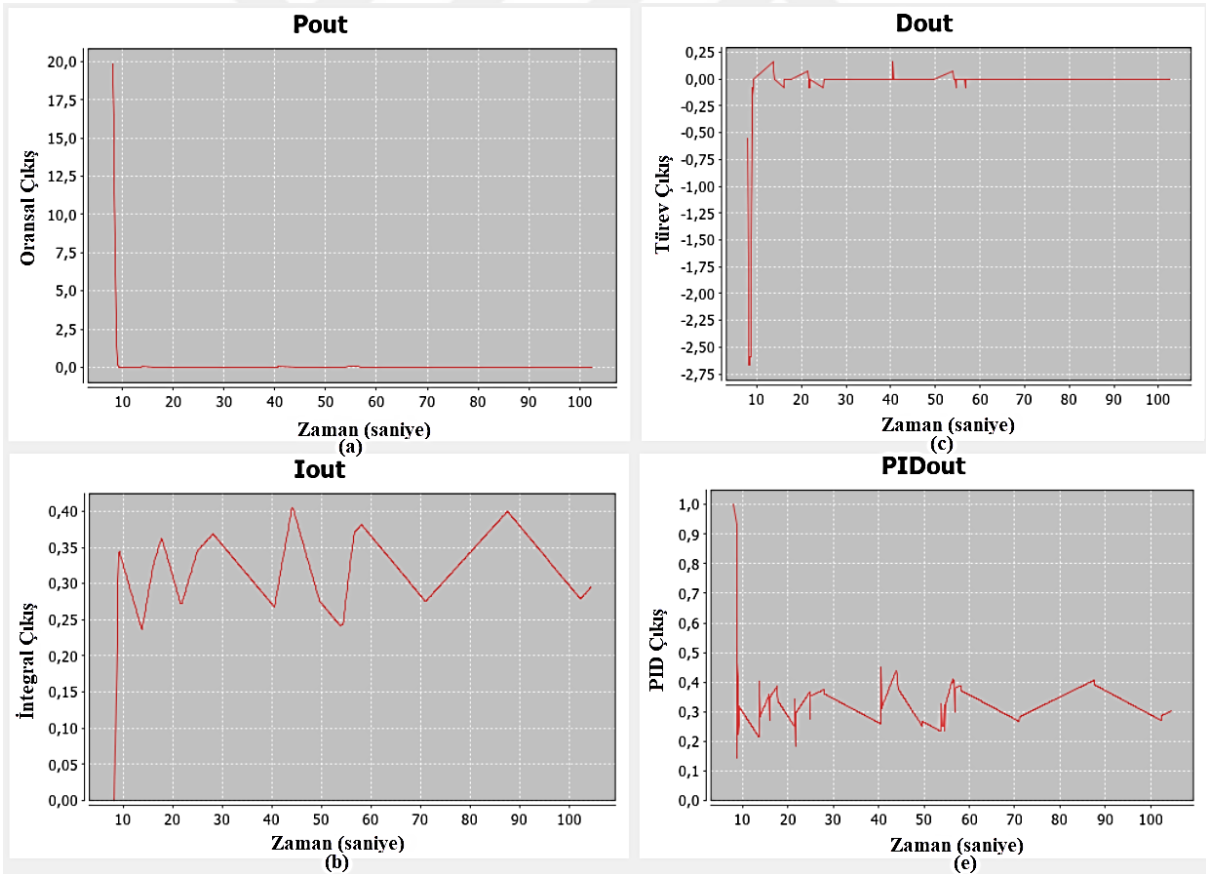
Şekil 5.15'teki hata-zaman grafiğinden hatanın 0'da tutularak sistemin kalıcı durumunun sağlandığı görülebilir.

Şekil 5.16'da oransal çıkış, integral çıkış ve türev çıkışın PID kontrolör çıkışına etkileri gösterilmiştir. Kalıcı duruma ulaştıktan sonra oransal çıkışın etkisi olmadığı, türevsel çıkışın aşimleri azaltmak için çok az değiştiği görülebilir. Kalıcı durumda integral kontrolün etkili olduğu görülebilir. İntegral çıkış kalıcı durumda 0,25 ile 0,4 arasında değişerek lift

sistemini referans konumda tutmaktadır. PIDout grafiğinde P, I ve D kontrolörlerin sisteme toplam etkisi gösterilmiştir.



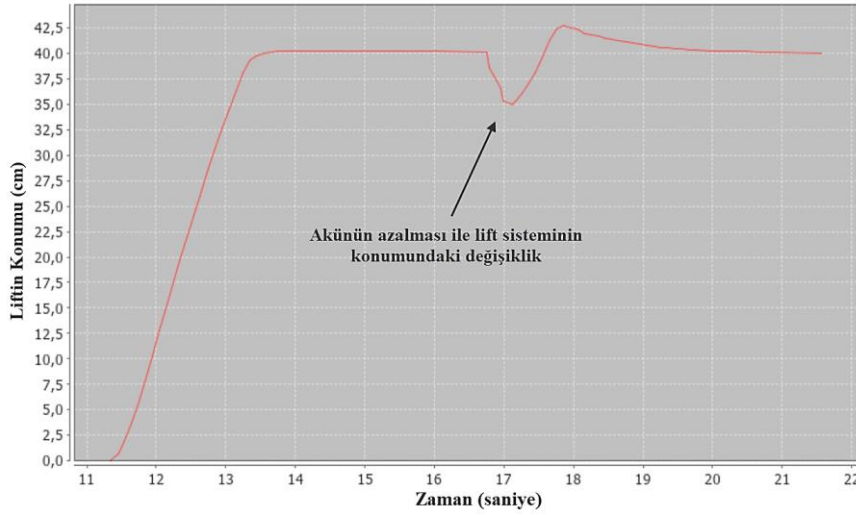
Şekil 5.15. PID kontrol hata-zaman eğrisi(~100s analiz)



Şekil 5.16. PID kontrolör çıkış eğrileri(~100s analiz) (a) oransal kontrol çıkışı (b) integral kontrol çıkışı (c) türev kontrol çıkışı (d) PID kontrolör çıkışı

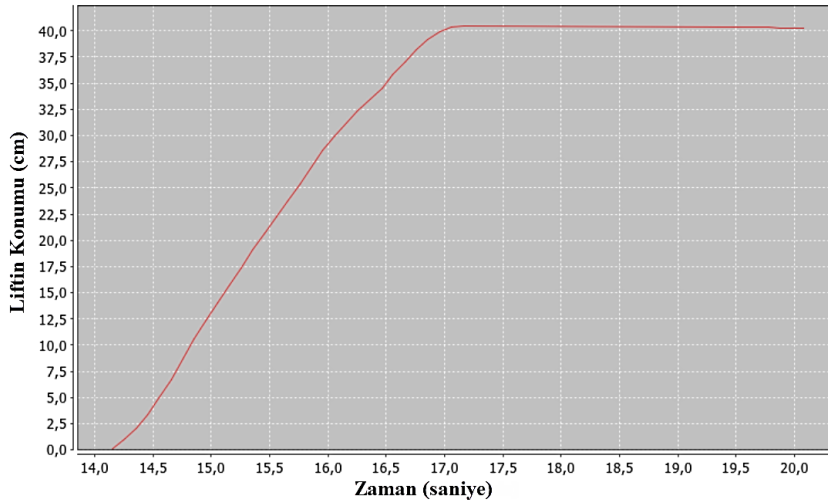
5.1.5. Mobil robot lift sisteminde akü geriliminin PID kontrolör üzerine etkileri

Mobil robot sistemi için gerekli güç 12V/18Ah akü tarafından sağlanmıştır. Şekil 5.17’de akü geriliminin azalması ile lift sisteminin konumundaki değişimler gösterilmiştir. Lift konumların azalmalarına PID kontrolör ani bir cevap vererek lift konumunu tekrar referans değere getirmiştir.



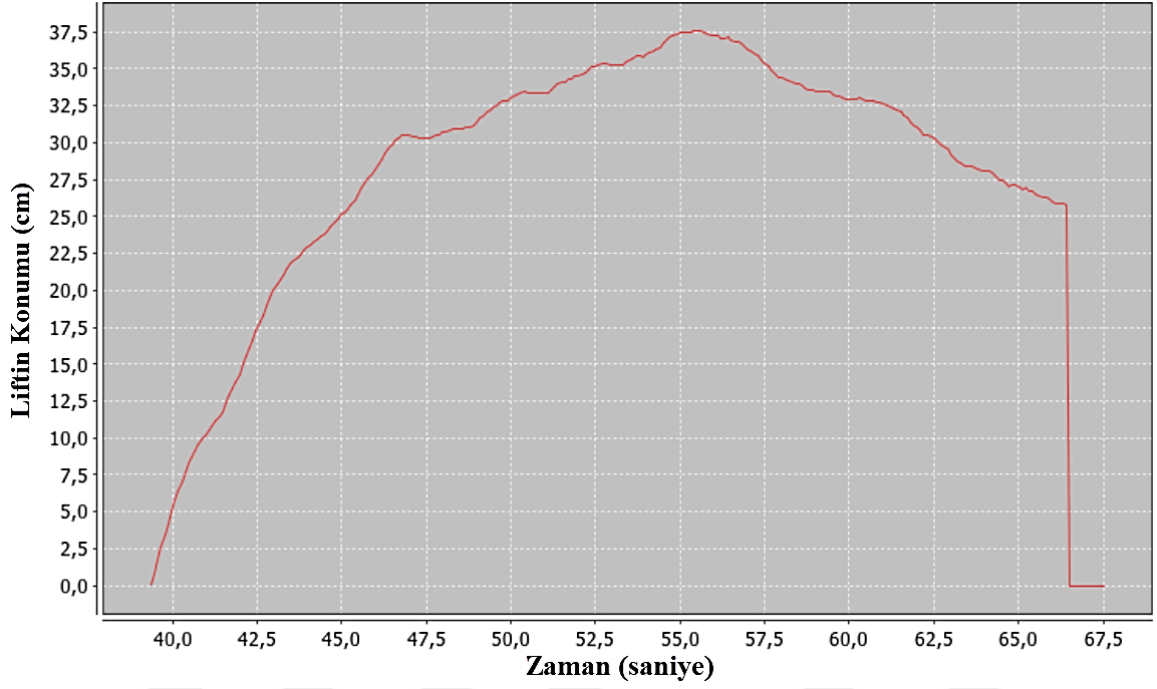
Şekil 5.17. Akü geriliminin lift sistemi üzerine etkileri

Akü geriliminin 11,5V'un altına düşmesi durumunda lift sisteminin konumundaki değişim Şekil 5.18’de gösterilmiştir. Motorlar gerekli gücü alamamasından dolayı kesikli olarak çalışmakta, yerleşme zamanı ~3s olarak PID kontrolörün mobil robot lift sistemini referans değere ulaştırma süresi uzamıştır.



Şekil 5.18. Akü geriliminin 11,5V seviyesine düşmesi durumunda lift sisteminin konumu

Akü gerilimi 11V seviyesine düşmesi durumunda lift sisteminin referans konuma ulaşamadığı ve tekrar 0 konumuna düştüğü Şekil 5.19’da gösterilmiştir. Akü geriliminin çok düşmesi sistem performansını etkileyecekse aküden yapılan geri besleme sonuçlarına bağlı olarak mobil robot sistemi bazı özellikleri kapatılarak gereksiz güç harcanması önlenabilir.



Şekil 5.19. Akü geriliminin 11V seviyesine düşmesi durumunda lift sisteminin konumu

Kısaca özetlemek gerekirse P kontrolör ile hata azalmış, PI kontrolör ile sistemin kalıcı yanıtı düzeltilmiş ve PID kontrolör ile sistemin hem kalıcı hem de geçici yanıtı düzeltilerek lift sistemi referans değer olarak belirlenen 40’a ulaşırken hata ~ 0 ’da sabitlenmiştir. Mobil robot lift sistemlerinde besleme geriliminin azalması ile kontrolör etkilerinin değiştiği, yerleşme zamanının uzadığı, kesikli bir çalışma olduğu gerilimin çok azalması durumunda ise lift konumunun referans değere ulaşamadığı görülmüştür.

5.1.6. Ziegler - Nichols metodunun uygulanması

Kontrolör parametrelerinin Ziegler-Nichols frekans yöntemine göre belirlenebilmesi için sistem oransal kontrol ile kontrol edilerek lift sisteminin sürekli salınım yaptığı durumdaki kazanç(K_u) ve salınım periyodunun(P_u) bulunması gerekir. Sistemin sürekli salınım yaptığı durumda $K_u=1$ ve $P_u=0,28$ olarak belirlenmiştir. Bu değerler kullanılarak Ziegler-Nichols ayar tablosu Çizelge 5.2’deki gibi oluşturulmuştur.

Çizelge 5.2. Ziegler-Nichols metodu ile PID parametrelerinin belirlenmesi

Kontrol	Kp	Ti	Td
P	$K_u / 2 = 0,5$		
PI	$K_u / 2,2 = 0,454$	$P_u / 1,2 = 0,233$	
PID	$K_u / 1,7 = 0,588$	$P_u / 2 = 0,14$	$P_u / 8 = 0,035$

Ziegler-Nichols metoduna göre PID parametreleri;

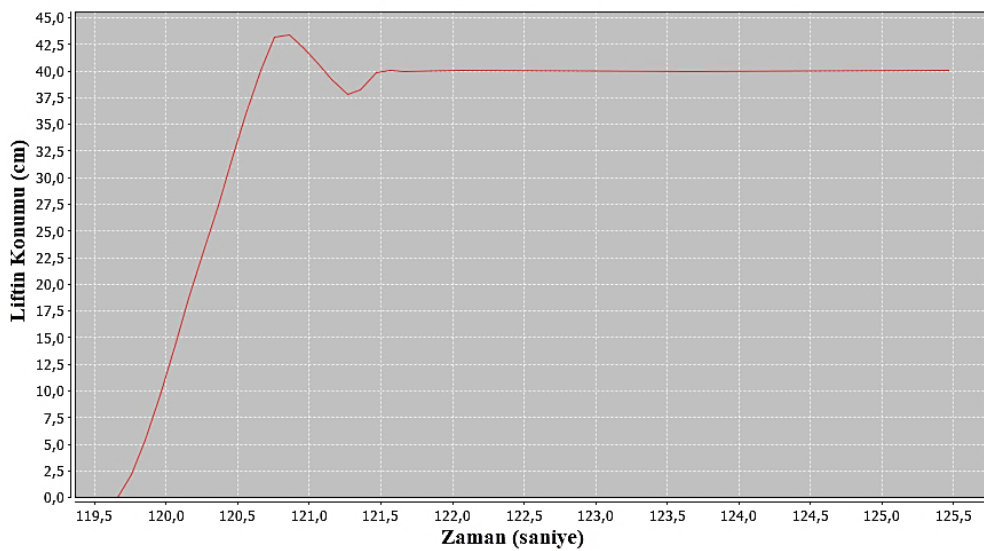
$$K_p = 0,588 \quad (5.1)$$

$$K_i = \frac{K_p}{T_i} = \frac{0,588}{0,14} = 4,2 \quad (5.2)$$

$$K_d = K_p T_d = 0,588 \cdot 0,035 = 0,02 \quad (5.3)$$

olarak elde edilir.

Ziegler-Nichols metodu ile bulunan PID parametreleri kullanılarak lift sistemi yanıtı Şekil 5.20'deki gibi elde edilmiştir. Sistem yanıtına bakıldığında ~2 saniyede referans değere ulaşılmıştır. Sistemde bir miktar aşım meydana gelmiştir. Ziegler-Nichols metodu tek başına mobil robotlarda kullanılan lift sistemleri için cevapları yetersiz kalabilir. Bu sebeple Ziegler-Nichols metodu ile tespit edilen PID parametrelerin üzerinde ayarlama yapmakta fayda vardır.

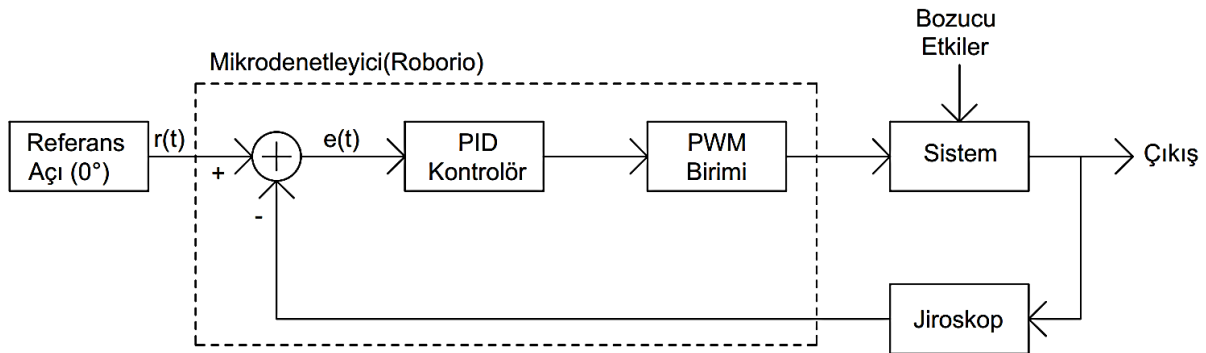


Şekil 5.20. Ziegler-Nichols Metodu ile lift sisteminin konum-zaman eğrisi

5.2. Robotu Düz Bir Yörüngede Hareket Ettirmek için PID Kontrolör Tasarımı

Robotun otonom olarak düz bir yörüngede hareketi için jiroskop sensörü ile ölçülen robota ait açı değerleri kullanılarak PID kontrolör gerçekleştirilmiştir. Tasarlanan sistemde sürtünme, ağırlık merkezinin dengesizliği vb. dış etkenler en aza indirilerek robotun düz bir yörüngede(referans değerinde) hareket etmesi sağlanmıştır. Sistemi zamanla bozacak etkenlere PID kontrolör cevap vermektedir. Sistemin sürekli hal hatası ~ 0 , bozucu etmenlere karşı düşük aşım göstermesi ve hızlı cevap vermesi, yerleşme zamanının düşük olması amaçlanmıştır.

Yörünge kontrol sisteminin blok diyagramı Şekil 5.21’de görülmektedir. Sistemdeki jiroskop ile robotun açısı ölçülerek mikrodnetleyiciye geri besleme yapılır. Çıkış açısı ile referans açısı(0°) karşılaştırılarak robot kontrolör tarafından referans açığa getirilir. Açı değeri mikrodnetleyici tarafından hız verisine dönüştürülür. PID kontrolör çıkışında elde edilen hız verisi, robotun sağ ve sol motorlarına bağlı Talon motor sürücülere iletilmiştir. Motor sürücüler PWM ile kontrol edilerek motorlara gerekli güç uygulanıp robotun hareketi sağlanır.



Şekil 5.21. Yörünge kontrol sisteminin blok diyagramı

5.2.1. Robotun düz yörüngede hareketi için PID kontrolör kodları

Robotun düz yörüngede gitmesi için PID kontrolör kodu Çizelge 5.3’te verilmiştir. Programda ilk olarak referans açı değeri 0 olarak belirlenmiştir. PID parametreleri $K_p2 = 0,08$; $K_i2 = 1$; $K_d2 = 0,05$; olarak ayarlanmıştır. Jiroskoptan ölçülen açı değerleri `olculen_aci` değişkenine aktarılmıştır. `if(hata2<-10||hata2>10) {toplam_hata2=0; }` komutu ile hata miktarının -10’un altına ve +10’un üstüne çıkması durumunda integral toplamı sıfırlanmıştır. PID işlemleri yapıldıktan sonra robotun maksimum dönüş hızları ± 0.8 olarak

belirlenmiştir. PID ile hesaplanan yön hızı myRobot.arcadeDrive(0.50, pid_out2); komutu ile robota gönderilir. Bu komuttaki ilk parametre robotun ileri ve geri gitmesini sağlarken ikinci parametre robotun sağa sola dönmesini sağlar. İlk parametre 0.50 ayarlanarak robotun %50

Çizelge 5.3: Robotun düz yörüngede gitmesi için PID kontrolör kodu

```
public void pid_robot()
{
    referans_aci=0; //referans açısı 0 olarak belirlendi
    Kp2 = 0.08; Ki2 =1; Kd2 = 0.05; //pid Kontrolör parametreleri belirlendi
    olculen_aci= gyro.getAngle(); //açı değeri alındı

    hata2 = referans_aci - olculen_aci; // hata2 hesaplandı
    turev2=(hata2-son_hata2)/0.02; // türev işlemi yapıldı
    toplam_hata2 = toplam_hata2 + hata2*0.02; // toplam hata hesaplandı
    son_hata2 = hata2; // hata2 değeri son_hata2 değişkenine aktarıldı
    if(hata2<-10||hata2>10) {toplam_hata2=0; } // integral toplamı sıfırlandı

    p_out2=hata2*Kp; // oransal çıkış hesaplandı
    i_out2=toplam_hata2*Ki; // integral çıkış hesaplandı
    d_out2=turev2*Kd; // türev çıkış hesaplandı
    pid_out2 = p_out2 + i_out2 + d_out2; // PID çıkış hesaplandı

    if(pid_out2 >= 0.8) pid_out2= 0.8; // çıkış sınırlandı
    else if(pid_out2 < -0.8) pid_out2=-0.8; // çıkış sınırlandı

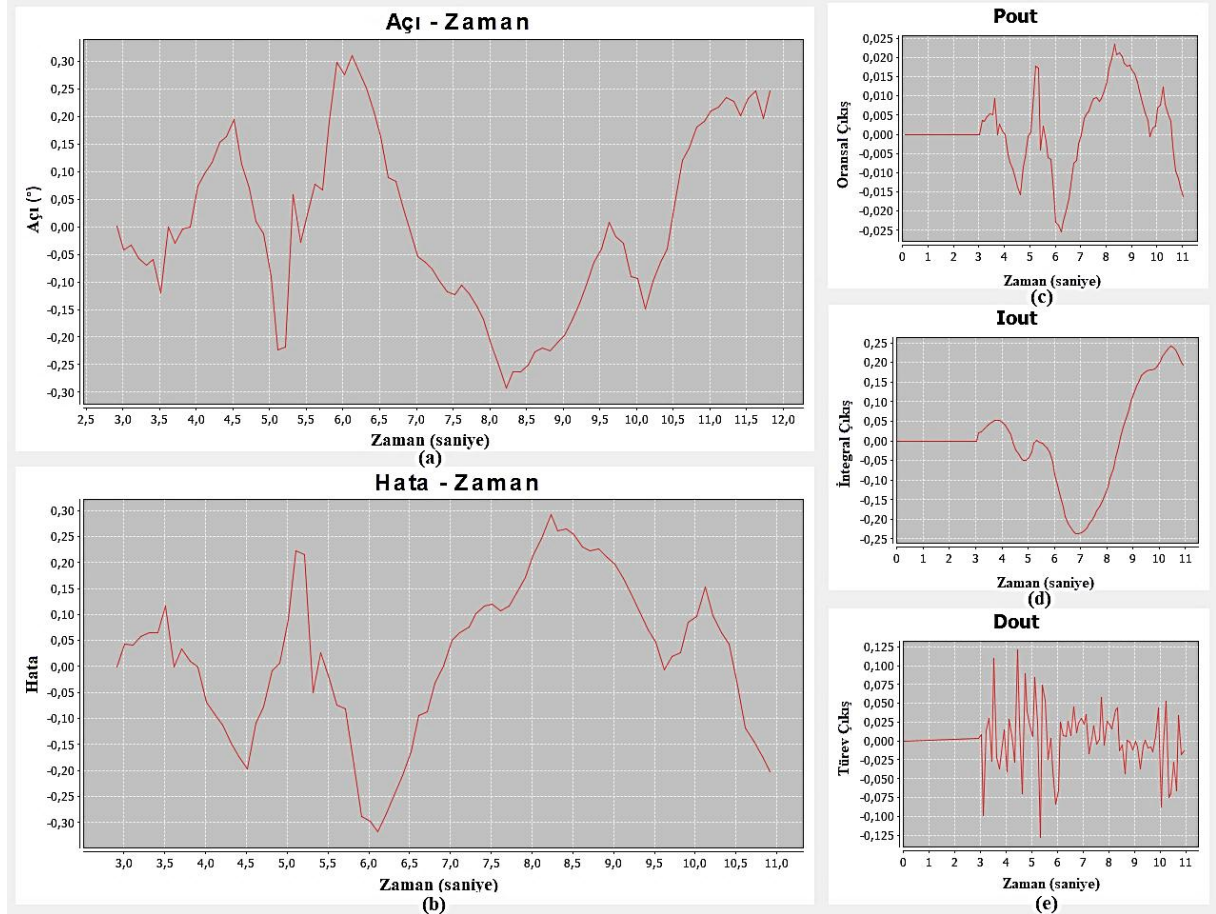
    myRobot.arcadeDrive(0.50, pid_out2); // robotun hareketi sağlandı
    myRobot.arcadeDrive(0.50, pid_out2); // robotun hareketi sağlandı
    Timer.delay(0.004); // gecikme verildi
}
```

PWM sinyali ile ileri hareketi sağlanmıştır. Robotu sağa ve sola döndüren ikinci parametre hesaplanan pid_out2 değişkeni ile belirlenmiştir.

5.2.2. Robotun düz yörüngede hareketine ait PID kontrolör etkileri

Robotun düz yörüngede hareketine ait PID kontrolör etkileri Şekil 5.22'deki grafiklerde gösterilmiştir. Jiroskop ile okunan robotun açısal konumları açı-zaman grafiğinde gösterilmiştir. Açı-zaman grafiğinden robotun "0" referans değerine yakın çok küçük salınımlar yaptığı görülmektedir. PID kontrolör ile robot, referans değere çok yakın açılarda tutulmuştur. Hata-zaman grafiğinde hatalar zamana göre çizdirilmiştir. Hata miktarı $\pm 1^\circ$ 'nin altında görülmüştür. Pout grafiğinde sisteme oransal kontrolörün etkisi, Iout grafiğinde sisteme integral kontrolörün etkisi, Dout grafiğinde sisteme türevsel kontrolörün etkisi gösterilmiştir.

Tasarımdaki oransal kontrol büyük hatalara karşı robotu düzeltmeye çalışır. Oransal kontrol küçük hatalarda robotu hareket ettirecek gücü sağlayamaz. Sisteme eklenen integral kontrol, kalıcı hal hatalarını sıfıra yakın değerlerde tutmaktadır. Sistemde oransal ve integral kontrolden kaynaklanan aşırımları azaltmak için D kontrol eklenerek çıkış hareketleri yumuşatılmıştır. PID kontrolör robotun referans değerde gitmesini sağlarken dış etmenlerden kaynaklanan olumsuz durumlara iyi bir cevap vermiştir.



Şekil 5.22. Robotun düz yörüngede hareketine ait PID kontrolör etkileri (a) açı-zaman grafiği (b) hata-zaman grafiği (c) oransal kontrol çıkışı (d) integral kontrol çıkışı (e) türev kontrol çıkışı

5.2.3. Robotun düz yörüngeden çıkarılmasına PID kontrolör cevabı

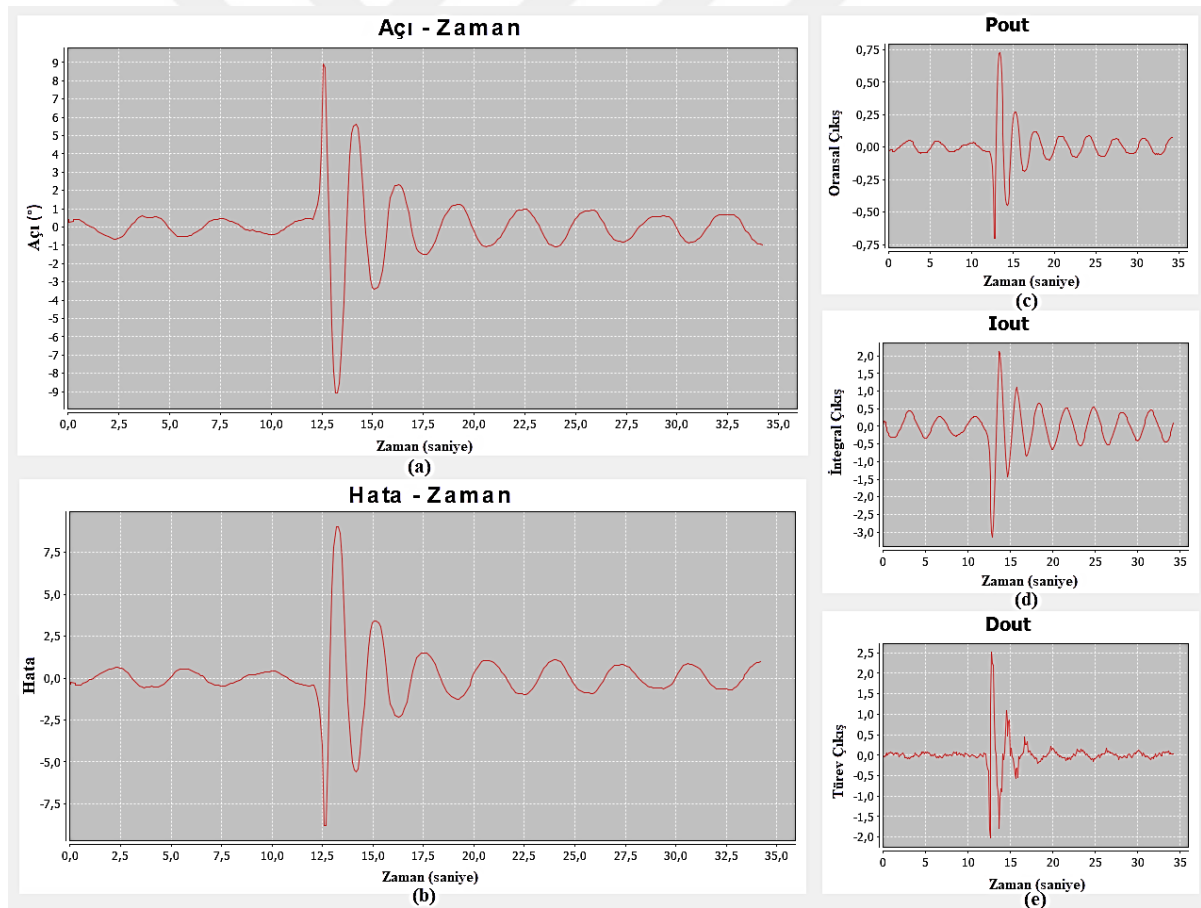
Düz bir yörüngede giden robota dışarıdan bir etki yapılarak robotun açısı değiştirildiğinde PID kontrolör etkileri Şekil 5.23'de verilen grafiklerde gösterilmiştir. Aç-zaman grafiğinden görüldüğü gibi dışarıdan yapılan bir etkiye kontrolör çok hızlı bir şekilde cevap vererek hatayı $\pm 1^\circ$ ye getirmiştir. Robot önce hızlı sonra yavaşlayarak sağ sol yapıp

referans açıya yakın bir değerde tutulmuştur. Hata-zaman grafiği sistemdeki hataları zamana göre göstermektedir.

Pout grafiğinde sisteme oransal kontrolörün etkisi görülmektedir. Hata miktarı arttıkça oransal kontrolörün etkisi artmaktadır. Oransal kontrol kararlı hal hatasını düşürmesine rağmen hata miktarı azaldıkça etkisi olmamaktadır.

Iout grafiğinde sisteme integral kontrolörün etkisi görülmektedir. Hata miktarının artması ile integral toplamı birikerek hatayı düzelterek kalıcı hal hatasını önemli oranda önlemiştir.

Dout grafiğinde sisteme türevsel kontrolörün etkisi gösterilmiştir. Hatanın değişimi ile sisteme öngörü kazandırılıp kontrolör tepkisi belli bir miktar yavaşlatılmış, salınım azaltılmış ve sistemin geçici yanıtı düzeltilmiştir.



Şekil 5.23. Robotun düz yörüngeden çıkarılmasıyla PID kontrolör cevabı(a) açı-zaman grafiği (b) hata-zaman grafiği (c) oransal kontrol çıkışı (d) integral kontrol çıkışı (e) türev kontrol çıkışı



6. SONUÇ ve ÖNERİLER

Bu yüksek lisans tezinde, mobil robot sisteminin bileşenleri incelenerek elektronik ve yazılım tasarımı açıklanmış robotun yörünge ve hareket kontrolü incelenmiştir. Robot, otonom ve uzaktan kumanda ile kontrol edilmiş, robotun lift sisteminin istenen konuma ulaşması ve robotun düz yörüngede gitmesi için PID kontrolör tasarımı gerçekleştirilmiştir. Tasarımı yapılan kontrolörün etkileri gerçek zamanlı konum-zaman ve açı-zaman grafikleri çizdirilerek incelenmiştir. Kontrolör etkilerine bakıldığında sadece P kontrolörün hataları azalttığı ancak yüksek aşımara ve kalıcı hal hatasına sebep olduğu için robotik sistemlerde tek başına kullanımının yetersiz olduğu görülmüştür. PD kontrol ise aşımları önlediği ancak kalıcı hal hatasına çözüm olmadığı görülmüştür. PI kontrol ile kalıcı hal hatasının düzeldiği ve kontrol parametreleri iyi seçilirse robot kontrolör tasarımında yeterli olabileceği görülmüştür. PI kontrol yapılırken ilk durumda hatanın yüksek olması integral toplamını arttırdığı için integral yığılması problemini ortaya çıkardığı görülmüştür. Bu durum hatanın belli değere düştükten sonra integral eklenerek çözülebileceği tespit edilmiştir. PI kontrole türev eklenerek PID kontrolör gerçekleştirilerek aşımalar engellenmiştir. PID kontrolör, kontrol parametreleri doğru seçildiğinde çok iyi sonuçlar verebilmektedir.

Robotik sistemlerde hata toleransının düşük istendiği durumlarda PI veya PID kontrolör çözüm olabilir. Ancak dış etkiler kontrolör sonuçlarını zamanla değiştirebildiği için kontrolör tasarımında dikkate alınmalıdır. Özellikle sürtünme ve aküdeki değişiklikler mobil robot sistemini etkileyebilir. Bu etkiler sensörler ile kontrolöre alınan geri besleme değerlerine göre kontrol parametreleri değiştirilerek giderilebilir. Örneğin kontrolör, akünün çok azalması durumunda lift sistemini hareket ettiremeyeceğini tespit ederek gereksiz yere lift sistemi motorları zorlanmayabilir. Ayrıca aküden alınan geri beslemeler ile sisteme öngörü kazandırılarak öncelikli sistemler için gerekli güç sağlanabilir, bazı sistemlerin çalışması engellenebilir.

Bu tezdeki robot kontrol tasarımlarının farklı mikroişlemcili platformlara uyarlanabilmesi için kontrolör için hazır fonksiyonlar kullanılmamıştır. Tezdeki uygulamalarda enkoder ve jiroskop ile geri besleme yapılarak PID kontrol gerçekleştirilmiştir. Mobil robot sistemine farklı sensörler eklenerek benzer kontrol işlemleri gerçekleştirilebilir. Örneğin robotun önüne engel geçmesi durumunda robotun çok ani tepki verip durması

isteniyorsa robota takılan bir mesafe sensöründen geri besleme alınarak PID kontrolör gerçekleştirilebilir.

Sürücü kontrollü robotik sistemlerde sürücünün sürekli robotu izlemesi gerektiği aynı zamanda belirli mesafede işlemlerin gerçekleştiği görülmüştür. Bu sistemlerde sürücü, robottan aldığı verileri doğru değerlendiremeyebilir. Kablosuz bağlantının kopması veya engellenmesi ile robotta belirsiz durumlar oluşabilir. Bu gibi durumlarda robotun doğru karar vermesi için mobil robotlardaki sensörlerden alınan geri beslemeler ile yapılan otomatik kontrol işlemleri büyük önem taşımaktadır.

Son yıllarda teknolojinin gelişmesi ile drone, otonom araçlar, otonom ürünler üzerine çalışmalar hızla artmaktadır. Hızla akıllaşan, otonomlaşan ve robotlaşan dünyamızda robotik sistem üzerine Türkçe kaynakların arttırılması çok önemlidir. Özellikle FPGA tabanlı mikroişlemciler ve sensörler kullanılarak farklı yeteneklere sahip robot tasarımı üzerine çalışmalar ülkemizin teknolojik gelişimine katkı sağlayacaktır.

7. KAYNAKÇA

- Akın Ö.(2016). Ve İnsansı Robotlarımızı Piyasaya Sürüyoruz. Inovax Dergisi 31:2-3, https://inovax.net/inovax/2018/05/inovax_e-dergi_31/#page/3 (erişim tarihi, 14.03.2019).
- Analog Devices (2019). ADXRS450 Overview. <https://www.analog.com/en/products/adxrs450.html?doc=ADXRS450.pdf#product-overview> (erişim tarihi, 16.02.2019).
- AndyMark(2019). AM14U4, Kit of Parts Chassis. <https://www.andymark.com/products/am14u4-kit-of-parts-chassis> (erişim tarihi, 16.03.2019).
- AndyMark (2019a). Toughbox Mini Product Overview. <https://www.andymark.com/products/toughbox-mini-options> (erişim tarihi, 04.02.2019).
- AndyMark (2019b). 120 Amp Breaker Product Overview. <https://www.andymark.com/products/120-amp-breaker>. (erişim tarihi, 05.02.2019)
- AndyMark(2019c). 2.5 in. CIM Motor. <https://www.andymark.com/products/2-5-in-cim-motor> (erişim tarihi, 17.02.2019).
- AndyMark(2019d). Snow Blower Motor. <https://www.andymark.com/products/snow-blower-motor>. (erişim tarihi, 16.03.2019).
- AndyMark (2019e). FRC Radio. <https://www.andymark.com/products/open-mesh-om5p-ac-dual-band-1-17-gbps-access-point-radio> (erişim tarihi, 18.04.2019).
- AndyMark(2019f).Batteries. <https://www.andymark.com/products/set-of-2-batteries-interstate-batteries>. (erişim tarihi, 18.03.2019).
- AndyMark(2019g). CIM Encoder. <https://www.andymark.com/products/cimcoder> (erişim tarihi, 18.03.2019).
- AndyMark (2019h). Analog Devices 2017-Gyro (Adxrs450). <https://firstchoicebyandymark.com/fc-adxrs450> (erişim tarihi, 12.03.2019).
- Anonim(2019). Control System Hardware Components. <http://team358.org/files/programming/ControlSystem2009-/components.php> (erişim tarihi, 05.03.2019).
- Anonim. (2019a). https://upload.wikimedia.org/wikipedia/commons/2/2c/360_controller.svg (erişim tarihi, 20.04.2019).
- Aström K.J. ve Höggund T.(1995). PID Controllers: Theory, Design and Tuning. Instrument Society of America,343 p, USA.
- AutomationDirect(2019). What is a Limit Switch. <https://library.automationdirect.com/what-is-a-limit-switch/> (erişim tarihi, 03.04.2019).
- Boston Dynamics (2019). Boston Dynamics Robots. <https://www.bostondynamics.com/robots> (erişim tarihi, 25.02.2019).
- Cao, Yi. (2019). Learning PID Tuning II: Stability Margin. Mathworks, https://www.mathworks.com/matlabcentral/fileexchange/18561-learning-pid-tuning-ii-stability-margin?s_tid=srchtitle (erişim tarihi, 03.03.2019).

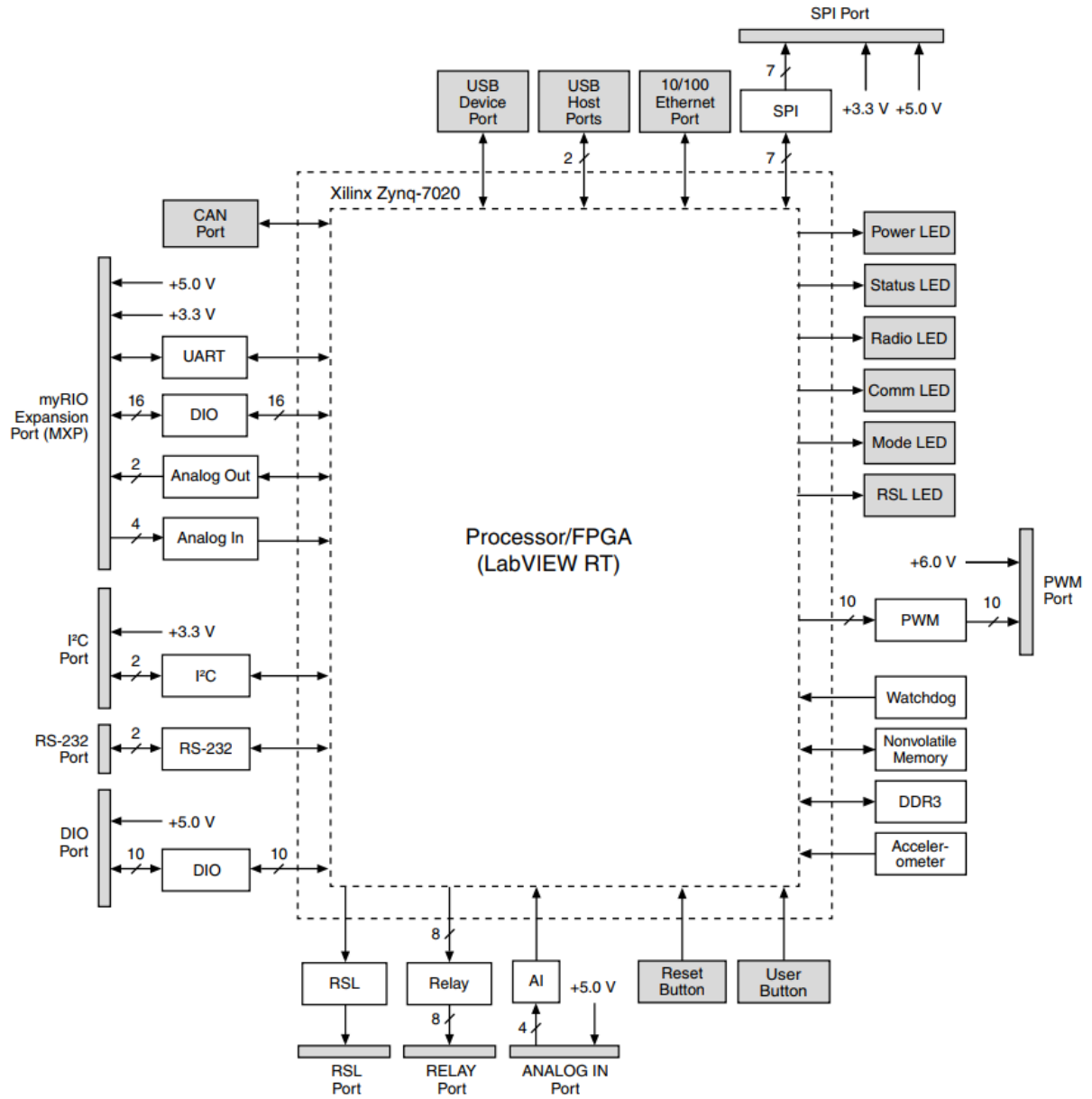
- Copeland, Brian R(2008). The Design of PID Controllers using Ziegler Nichols Tuning. Educypedia, http://educypedia.karadimov.info/library/Ziegler_Nichols.pdf (erişim tarihi, 03.03.2019).
- CTR Electronics (2019). Power Distribution Panel. <http://www.ctr-electronics.com/downloads/pdf/PDP%20User's%20Guide.pdf> (erişim tarihi, 10.02.2019).
- CTR Electronics (2019a). Voltage Regulator Module User's Guide. <http://www.ctr-electronics.com/VRM%20User's%20Guide.pdf> (erişim tarihi, 10.02.2019).
- CTR Electronics(2019b). Talon SRX User's Guide. <http://www.ctr-electronics.com/Talon%20SRX%20User's%20Guide.pdf> (erişim tarihi, 11.02.2019).
- CTR Electronics(2019c). Pneumatic Control Module User's Guide. <http://www.ctr-electronics.com/downloads/pdf/PCM%20User's%20Guide.pdf> (erişim tarihi, 10.02.2019).
- Cui Global (2019). What is an Encoder. <https://www.cui.com/product-spotlight/capacitive-absolute-encoders-amt20-series> (erişim tarihi, 23.03.2019).
- Cui Global (2019a). What's the Difference Between an Incremental Encoder's PPR, CPR, and LPR. <https://www.cui.com/blog/what-is-encoder-ppr-cpr-and-lpr> (erişim tarihi, 23.03.2019).
- Dalcı, K.B.(2004). Bulanık kazanç ayarlamalı nonlinear PI kontrolörün kalıcı mıknatıslı doğru akım motor-dinamo sistemine uygulanması. Doğu Üniversitesi Dergisi, 5(1): 71-88.
- DARPA (2019). Darpa Robotics Challenge Finals. <https://archive.darpa.mil/roboticschallenge/> (erişim tarihi, 09.02.2019).
- Dynapar (2019). Quadrature Encoder Overview. https://www.dynapar.com/technology/encoder_basics/quadrature_encoder/ (erişim tarihi, 25.02.2019)
- EBSO (2015). Sanayi 4.0 Uyum Sağlamayan Kaybedecek. Ege Bölgesi Sanayi Odası, http://www.ebso.org.tr/ebsomedia/documents/sanayi-40_88510761.pdf. (erişim tarihi, 26.02.2019).
- FIRST (2019). Vision and Mission. <https://www.firstinspires.org/about/vision-and-mission> (erişim tarihi, 02.03.2019).
- FIRST (2019a). What is FIRST Robotics Competition. <https://www.firstinspires.org/robotics/frc> (erişim tarihi, 02.03.2019).
- Honda (2019). History of Asimo. <https://asimo.honda.com/asimo-history/> (erişim tarihi, 02.04.2019).
- Honda (2019a). Asimo Innvations. <https://asimo.honda.com/innovations/> (erişim tarihi, 02.04.2019).
- Honeywell (2019) Micro Switch Overview. <https://sensing.honeywell.com/V7-2B17D8-048-general-purpose-basic-switches> (erişim tarihi, 05.04.2019).
- Kansagara, R. (2018). PID Controllers: Working, Structure and Tuning Methods. Circuitdiges, <https://circuitdigest.com/article/what-is-pid-controller-working-structure-applications> (erişim tarihi, 07.04.2019).

- Kuo, Benjamin C. (1999) Otomatik Kontrol Sistemleri. Yedinci Baskı. Çeviren Prof. Dr. Atilla Bir. Literatür Yayınları, 933, İstanbul.
- Курганов С.М.(2016).СИСТЕМА ПЛАНИРОВАНИЯ ТРАЕКТОРИИ И УПРАВЛЕНИЯ ДВИЖЕНИЕМ ДВУХКОЛЕСНОГО МОБИЛЬНОГО РОБОТА (İki tekerlekli mobil robot için planlama ve trafik yönetim sistemi). <http://elib.spbstu.ru/dl/2/v16-1652.pdf/download/v16-1652.pdf>. (erişim tarihi, 25.02.2019).
- Microsoft (2019). LifeCam HD-3000. <https://www.microsoft.com/accessories/en-us/products/webcams/lifecam-hd-3000/t3h-00011#details>. (erişim tarihi, 25.02.2019).
- Mohamed Emam F(2015). Control Systems I Lecture 11 PID. http://www.aast.edu/pheed/staffadminview/pdf_retrieve.php?url=351_27435_EE411_2015_1__1_1_0%20%20EE411%20Lec11%20PID.pdf&stafftype=staffcourses (erişim tarihi, 25.04.2019).
- National Instruments(2019). PID Theory Explained. <http://www.ni.com/en-tr/innovations/white-papers/06/pid-theory-explained.html>. (erişim tarihi, 02.03.2019).
- National Instruments (2019a). roboRIO Advanced Robotics Controller. <http://www.ni.com/en-tr/support/model.roborio.html> (erişim tarihi, 02.03.2019).
- Jenkis, Hodge (2016). Tuning for PID Controllers. Mercer University, http://faculty.mercer.edu/jenkins_he/documents/TuningforPIDControllers.pdf (erişim tarihi, 19.03.2019).
- REV Robotics (2019). Spark Motor Controller Quick Start Guide. <http://www.revrobotics.com/content/docs/REV-11-1200-QS.pdf> (erişim tarihi, 11.03.2019).
- Smith, A (2014). Digital Life in 2025 AI, Robotics and the Future of Jobs. PewResearchCenter, <https://www.pewresearch.org/wp-content/uploads/sites/9/2014/08/Future-of-AI-Robotics-and-Jobs.pdf>. (erişim tarihi, 10.03.2019).
- Tsoy T, Sabirov S, Magid E (2017). Towards Effective Interactive Teaching and Learning Strategies in Robotics Education. 2017 10th International Conference on Developments in eSystems Engineering (DeSE), Intelligent Robotics Department, Higher Institute of Information Technology and Information Systems, Republic of Tatarstan, Russia.
- VEX Robotics (2019). VEX Robotics Motor Data. <https://motors.vex.com/vexpro-motors/bag-motor> (erişim tarihi, 11.03.2019).
- WPI (2019). RoboRIO Brownout and Understanding Current Draw. https://wpilib.screenstepslive.com/s/currentCS/m/cs_hardware/l/289498-roborio-brownout-and-understanding-current-draw (erişim tarihi, 18.04.2019).
- WPI (2019a). Status Light Quick Reference. https://wpilib.screenstepslive.com/s/currentCS/m/cs_hardware/l/144972-status-light-quick-reference (erişim tarihi, 09.04.2019).
- WPI(2019b). FRC Software Component Overview. https://wpilib.screenstepslive.com/s/currentCS/m/getting_started/l/144981-frc-software-component-overview (erişim tarihi, 09.04.2019).

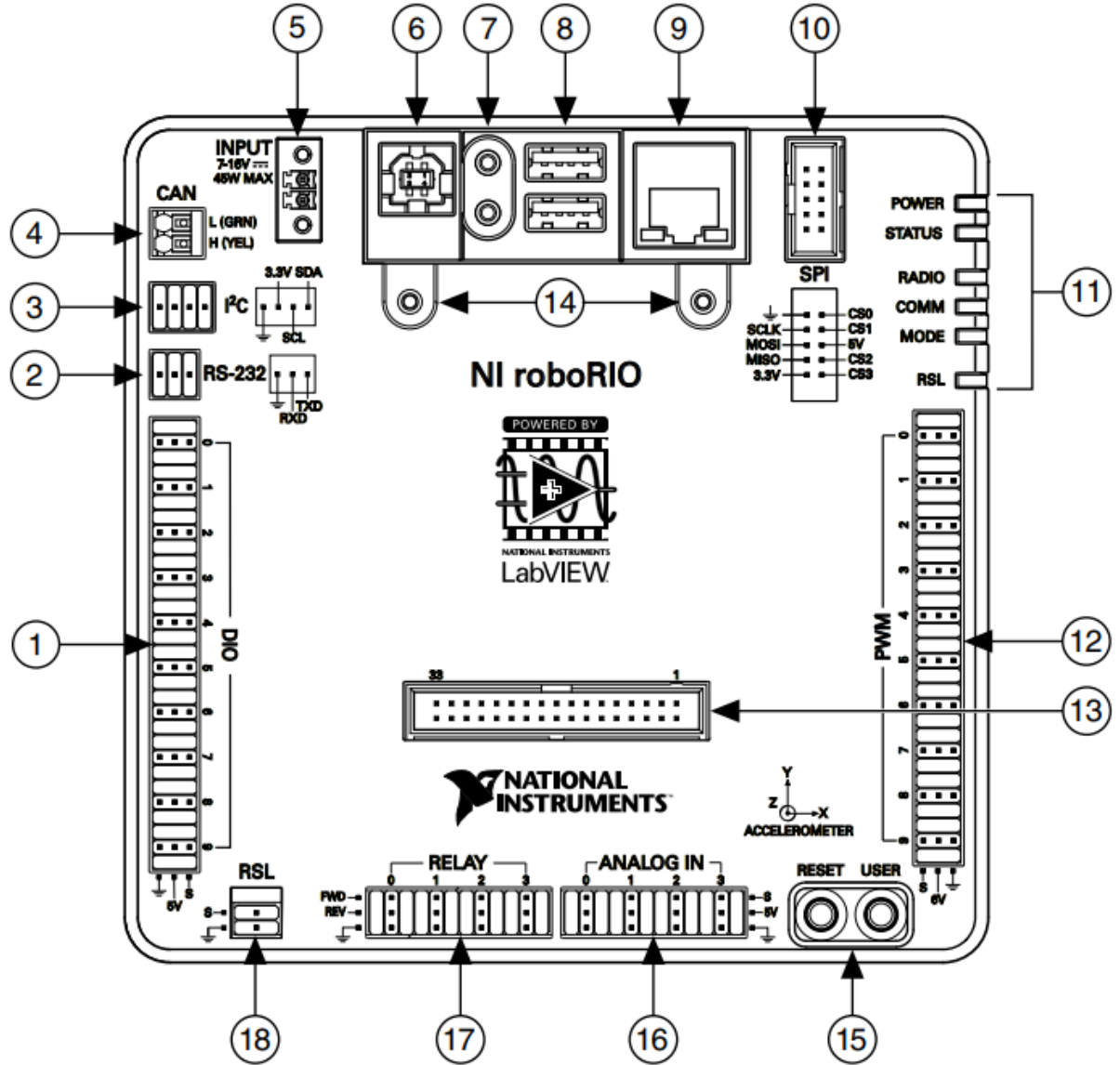
- WPI (2019c). Getting Started With The Control System.
https://media.screensteps.com/exported/Wpilib/2078/119063/Getting_Started_with_the_Control_System_.pdf (erişim tarihi, 09.04.2019).
- WPI (2019d). What is WPILib.
<https://wpilib.screenstepslive.com/s/currentCS/m/java/l/599696-what-is-wpilib> (erişim tarihi, 13.04.2019).
- WPI (2019e). FRC Java Programming.
https://media.screensteps.com/exported/Wpilib/2078/120335/FRC_Java_Programming.pdf (erişim tarihi, 29.03.2019).
- Zhao, Z.Y., Tomizuka, M. & ISAKA, S. (1993) Fuzzy gain scheduling of PID controllers. IEEE Transactions on Systems, Man and Cybernetics, 23(5):1392-1398.



EK 1 NI roboRIO BLOK DİYAGRAMI



EK 2 NI roboRIO GİRİŞ ÇIKIŞ PORTLARI



1. Dijital giriş ve çıkış portu (DIO)
2. RS-232 portu
3. I2C portu
4. CAN portu
5. Güç Girişi
6. USB aygıt portu
7. USB tutucusu
8. USB Host portları
9. Ethernet portu
10. SPI portu
11. LED' ler
12. PWM portları
13. MXP portu
14. MXP tutma bağlantısı
15. Kullanıcı ve durum butonları
16. Analog giriş portu
17. Röle portu
18. Robot sinyal lambası(RSL) portu



EK 3 ACCESS POINT KONFIGÜRASYON AYARLARI

FRC Radio Configuration Utility

File Tools Help

Team Number: **Robot Name:**

WPA Key: **Firewall:** ☐

Radio: **BW Limit:** ☒

Mode: **Load Firmware**

Configure



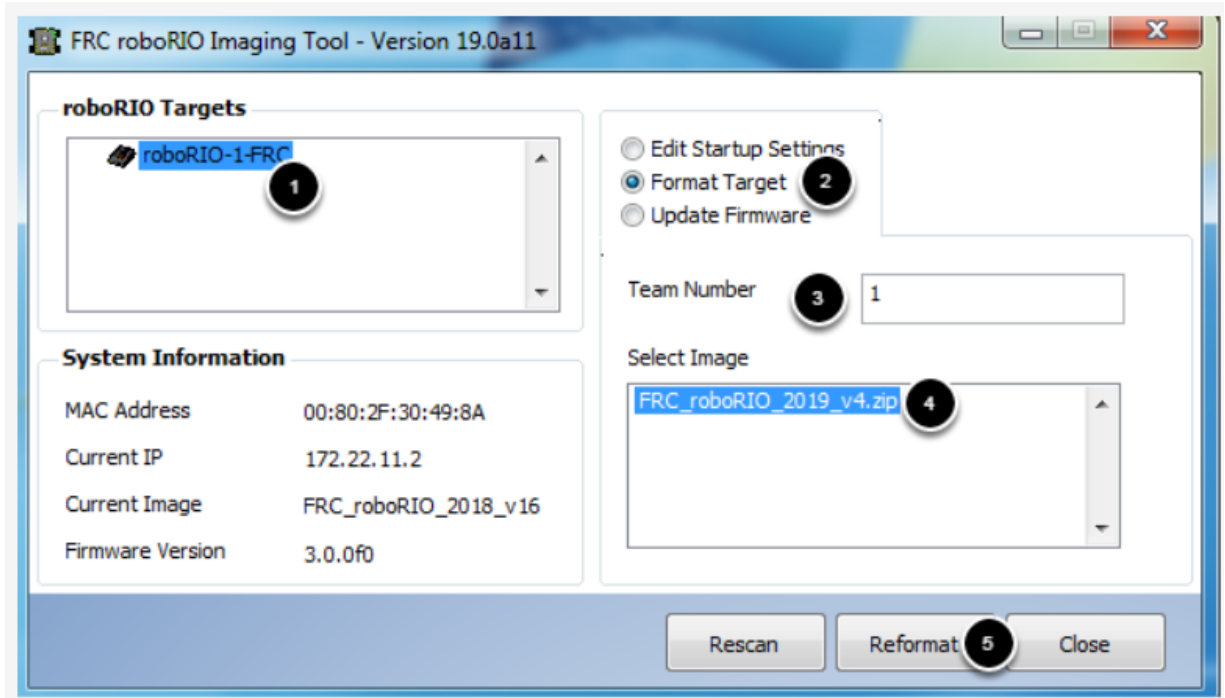
To program your wireless bridge:

- 1) Connect power and Ethernet to the wireless bridge.
- 2) Make sure to use the "802.3af" Ethernet port as shown above.
- 3) Wait for the Power light to turn and stay solid.
- 4) Enter your team number, and a WPA key (optional), above
- 5) Press "Configure", the process should take 15-60 seconds

If asked to reset your wireless bridge:

- 1) If Event WPA Kiosk: Follow on-screen prompts
- 1) If Radio Config Utility: Select OpenMesh
- 2) Unplug power from the radio
- 3) Press the "Load Firmware" button.
- 4) Follow the on-screen prompts.

EK 4 ROBORIO IMAGE İŞLEMLERİ



EK 5 ROBOT KODU

```
// Kütüphane dosyaları eklendi
package frc.robot;
import edu.wpi.first.wpilibj.TimedRobot;
import edu.wpi.first.wpilibj.smartdashboard.SmartDashboard;
import edu.wpi.first.wpilibj.Joystick;
import edu.wpi.first.wpilibj.drive.DifferentialDrive;
import edu.wpi.first.wpilibj.Spark;
import edu.wpi.first.wpilibj.Compressor;
import edu.wpi.first.wpilibj.DigitalInput;
import edu.wpi.first.wpilibj.Timer;
import edu.wpi.first.wpilibj.SpeedController;
import edu.wpi.first.cameraserver.CameraServer;
import edu.wpi.first.wpilibj.Encoder;
import edu.wpi.first.wpilibj.PWMTalonSRX;
import edu.wpi.first.wpilibj.Solenoid;
import org.opencv.core.Mat;
import org.opencv.imgproc.Imgproc;
import edu.wpi.cscore.CvSink;
import edu.wpi.cscore.CvSource;
import edu.wpi.cscore.UsbCamera;
import edu.wpi.first.wpilibj.ADXRS450_Gyro;
import edu.wpi.first.wpilibj.AnalogInput;

// Ana program
public class Robot extends TimedRobot
{
    Compressor c = new Compressor();
    Solenoid pnomatik = new Solenoid(0);
    public ADXRS450_Gyro gyro = new ADXRS450_Gyro();
    PWMTalonSRX m_left = new PWMTalonSRX(0);
    PWMTalonSRX m_right = new PWMTalonSRX(1);
    DifferentialDrive myRobot = new DifferentialDrive(m_left, m_right);
    Joystick kumanda_kolu = new Joystick(0);
    public SpeedController lift_motor1 = new Spark(2);
    public SpeedController lift_motor2 = new Spark(3);
    private SpeedController kol_sagmotor = new Spark(4);
    private SpeedController kol_solmotor = new Spark(5);
    private SpeedController tutucu_sagmotor = new Spark(6);
    private SpeedController tutucu_solmotor = new Spark(7);
    public static final double kPulsesPerRotation = 20; // Enkoder ayarları
    public static final double kDistancePerPulse = 0.125; // Enkoder ayarları
    public final static int kCIMcoderDioPortA = 0; // enkoder A portu DIO 0 a bağlı
    public final static int kCIMcoderDioPortB = 1; // enkoder B portu DIO 1 e bağlı
    private Encoder cimcoder;
    DigitalInput sinir_anahtari = new DigitalInput(9);
    Timer timer = new Timer();
    double Kp, Kd, Ki, Kp2, Ki2, Kd2;
```

```

double olculen_konum, referans_konum, hata, turev, toplam_hata;
double p_out, i_out, d_out, pid_out, son_hata;
double olculen_aci, referans_aci, hata2, turev2, toplam_hata2;
double p_out2, i_out2, d_out2, pid_out2, son_hata2;
int kademe=5;

// Bu metod robot ilk çalıştırıldığında başlatılır
@Override
public void robotInit() {
    cimcoder = new Encoder(kCIMcoderDioPortA,
kCIMcoderDioPortB,false,Encoder.EncodingType.k4X);
    cimcoder.setDistancePerPulse(kDistancePerPulse);
    cimcoder.reset();
    getDistanceTraveled();
    c.start(); // kompresör başlatıldı
    c.setClosedLoopControl(true);
    gyro.calibrate(); // jiroskop kalibrasyonu yapıldı
    gyro.reset(); // jiroskop resetlendi
    referans_konum=0;
    p_out=0; i_out=0; d_out=0;
    Kp = 0.5; Ki =0.5; Kd = 0.05;

    new Thread(() -> { //Kamera ayarları yapıldı
        UsbCamera camera = CameraServer.getInstance().startAutomaticCapture();
        camera.setResolution(640, 480);
        CvSink cvSink = CameraServer.getInstance().getVideo();
        CvSource outputStream = CameraServer.getInstance().putVideo("Blur", 320, 240);
        Mat source = new Mat();
        Mat output = new Mat();
        while(!Thread.interrupted()) {
            cvSink.grabFrame(source);
            Imgproc.cvtColor(source, output, Imgproc.COLOR_BGR2GRAY);
            outputStream.putFrame(output);
        }
    }).start();
}

// Otonom başlangıç ayarları
@Override
public void autonomousInit()
{
    timer.reset();
    timer.start();
    cimcoder.reset();
    gyro.reset();
    olculen_aci = gyro.getAngle();
}

// Robot otonom olarak çalışır
@Override

```

```

public void autonomousPeriodic() // otonom mod
{
    if (timer.get() < 40.0) { //40 saniye otonom olarak çalış
        pid_robot(); //robotu düz bir yörüngede gitmesini sağla
        goster_aci(); //değişkenleri SmartDashboard ekranında göster
    }
    else {
        myRobot.arcadeDrive(0.0, 0.0);
    }
}

// Robot sürücü tarafından tuş takımı ile kontrol edilir
@Override
public void teleopPeriodic() // Sürücü modu
{
    //robotun yön hareketlerini sağla
    myRobot.arcadeDrive(-kumanda_kolu.getY(), kumanda_kolu.getX() );
    robot_kontrol();//robot bileşenleri çalıştırılır
    lift_kontrol();//otomatik lift konumlandırmasını yapar
}

// robotun tuş takımı ile kontrol işlemleri yapılır
public void robot_kontrol()
{
    // liftin tuş takımı ile hareketi sağlandı
    if(kumanda_kolu.getTwist()>0.03) { // sol tuş(LT)-lift aşağı
        lift_motor1.set(-0.3*kumanda_kolu.getTwist());
        lift_motor2.set(-0.3*kumanda_kolu.getTwist());
        kademe=5;
    }
    else if(kumanda_kolu.getThrottle()>0.03) { // sağ tuş (RT)-lift yukarı
        lift_motor1.set(kumanda_kolu.getThrottle());
        lift_motor2.set(kumanda_kolu.getThrottle());
        kademe=5;
    }
    else {
        if(kademe==5) {
            lift_motor1.set(0.2); // lift sabit tutuldu
            lift_motor2.set(0.2); // lift sabit tutuldu
        }
    }
    // robot kol hareketleri
    if(kumanda_kolu.getRawButton(1)) {
        kol_solmotor.set(0.7); // kolu aşağı indirir
        kol_sagmotor.set(-0.7); // kolu aşağı indirir
    }
    else if(kumanda_kolu.getRawButton(2)) {
        kol_solmotor.set(-1); // kolu yukarı kaldırır
        kol_sagmotor.set(1); // kolu yukarı kaldırır
    }
}

```

```

else {
    kol_solmotor.set(0); // kol motorları pasif
    kol_sagmotor.set(0); // kol motorları pasif
}
// nesne tutma sistemi
if(kumanda_kolu.getRawButton(5)){ //paketi bırak
    tutucu_solmotor.set(-1); // paket sol motoru dışarı yönde döner
    tutucu_sagmotor.set(1); // paket sağ motoru dışarı yönde döner
}
else if(kumanda_kolu.getRawButton(6)){ // paketi al
    tutucu_solmotor.set(0.6); // paket sol motoru içeri yönde döner
    tutucu_sagmotor.set(-0.6); // paket sağ motoru içeri yönde döner
}
else {
    tutucu_solmotor.set(0); // paket sol motoru pasif
    tutucu_sagmotor.set(0); // paket sağ moturu pasif
}
// pnömatik sistem kontrolü
if(kumanda_kolu.getRawButton(4)){
    pnomatik.set(true); // selonoid aktif yapıldı
}
else {
    pnomatik.set(false); // selonoid pasif yapıldı
}
}
// Liftin referans değere konumlanması sağlanır
public void lift_kontrol() // Sadece lift konumlandırmasını yapar
{
    if(kumanda_kolu.getPOV()==0.0) { kademe=3; } // lift kademe 3
    else if(kumanda_kolu.getPOV()==90.0) { kademe=2; } // lift kademe 2
    else if(kumanda_kolu.getPOV()==180.0) { kademe=1; } // lift kademe 1
    else if(kumanda_kolu.getPOV()==270.0) { kademe=0; } // lift kademe 0

    if(kademe==3) referans_konum=60;
    else if(kademe==2) referans_konum=40;
    else if(kademe==1) referans_konum=20;
    else if(kademe==0) referans_konum=0;
    if(kademe==3||kademe==2||kademe==1||kademe==0)
    {
        pid_lift();
        goster_lift(); //değişken durumları SmartDashboard da gösterilir
    }
    // sınır anahtarı kontrolü
    if(sinir_anahtari.get() == false){
        cimcoder.reset(); // enkoder resetlendi
    }
}

// Lift için PID hesaplanır
public void pid_lift()

```

```

{
  Kp = 0.5; Ki = 0.5; Kd = 0.05; //parametreler belirlendi
  olculen_konum= getDistanceTraveled();//lifte bağılı enkoderden veri okundu
  hata = referans_konum - olculen_konum; //hata hesaplandı
  turev=(hata-son_hata)/0.02; //türev işlemi yapıldı
  toplam_hata= toplam_hata + hata*0.02; //integral toplamı hesaplandı
  son_hata= hata; //hata içeriği son_hata olarak belirlendi
  if(hata>7) toplam_hata=0; //integral toplamı sıfırlandı

  p_out=hata*Kp; //oransal çıkış hesaplandı
  i_out=toplam_hata*Ki; //integral çıkış hesaplandı
  d_out=turev*Kd; //türev çıkış hesaplandı
  pid_out = p_out + i_out + d_out; //PID çıkış hesaplandı

  if(pid_out >= 1) pid_out= 1; //PID çıkış sınırlandı
  else if(pid_out < - 0.1) pid_out= - 0.1; //PID çıkış sınırlandı

  lift_motor1.set(pid_out); //PID değeri lift motoruna gönderildi
  lift_motor2.set(pid_out); //PID değeri lift motoruna gönderildi
}

// Robotun düz yörüngede hareketi için PID hesaplanır
public void pid_robot()
{
  referans_aci=0; //referans açı 0 olarak belirlendi
  Kp2 = 0.08; Ki2 =1; Kd2 = 0.05; //pid Kontrolör parametreleri belirlendi
  olculen_aci= gyro.getAngle(); //açı değeri alındı

  hata2 = referans_aci - olculen_aci; // hata2 hesaplandı
  turev2=(hata2-son_hata2)/0.02; // türev işlemi yapıldı
  toplam_hata2 = toplam_hata2 + hata2*0.02; // toplam hata hesaplandı
  son_hata2 = hata2; // hata2 değeri son_hata2 değişkenine aktarıldı
  if(hata2<-10||hata2>10) {toplam_hata2=0; } // integral toplamı sıfırlandı

  p_out2=hata2*Kp; // oransal çıkış hesaplandı
  i_out2=toplam_hata2*Ki; // integral çıkış hesaplandı
  d_out2=turev2*Kd; // türev çıkış hesaplandı
  pid_out2 = p_out2 + i_out2 + d_out2; // PID çıkış hesaplandı

  if(pid_out2 >= 0.8) pid_out2= 0.8; // çıkış sınırlandı
  else if(pid_out2 < -0.8) pid_out2=-0.8; // çıkış sınırlandı

  myRobot.arcadeDrive(0.65, pid_out2); // robotun hareketi sağlandı
  myRobot.arcadeDrive(0.65, pid_out2); // robotun hareketi sağlandı
  Timer.delay(0.004);
}

// Lifte bağılı enkoder tanımlaması yapıldı
public double getDistanceTraveled(){
  return cimcoder.getDistance();
}

```

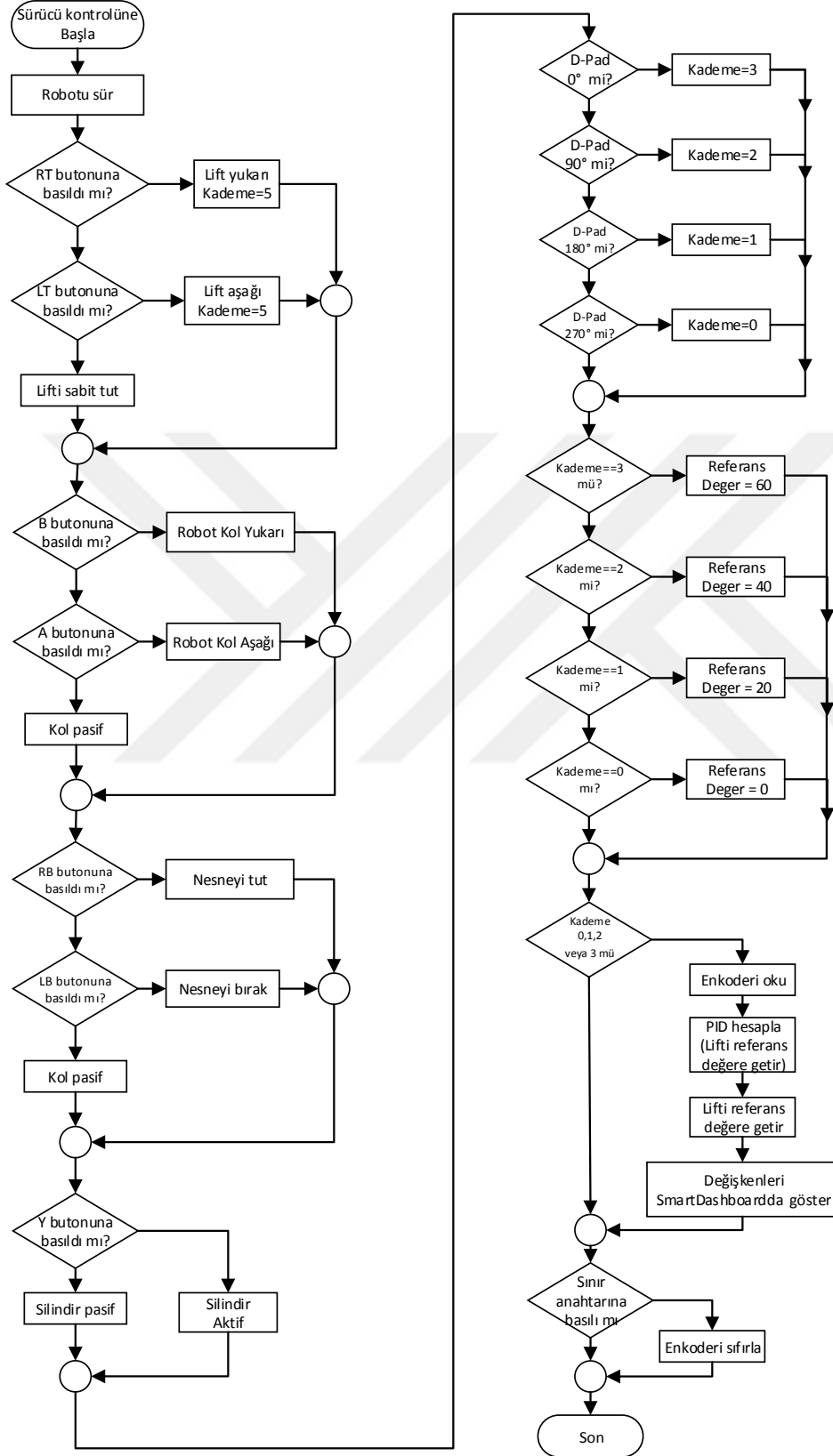
```

}

// Değişken değerleri SmartDashboard ekranında gösterildi
void goster_lift()
{
    SmartDashboard.putNumber("Konum-Zaman",-1*getDistanceTraveled());
    SmartDashboard.putNumber("Pout",p_out);
    SmartDashboard.putNumber("Iout",i_out);
    SmartDashboard.putNumber("Dout",d_out);
    SmartDashboard.putNumber("Kp:",Kp); *
    SmartDashboard.putNumber("Ki:",Ki);
    SmartDashboard.putNumber("Kd:",Kd);
    SmartDashboard.putNumber("Hata-Zaman",hata);
}
void goster_aci()
{
    SmartDashboard.putNumber("Aci-Zaman", gyro.getAngle());
    SmartDashboard.putNumber("Pout ",p_out2);
    SmartDashboard.putNumber("Iout ",i_out2);
    SmartDashboard.putNumber("Dout ",d_out2);
    SmartDashboard.putNumber("Kp: ",Kp2);
    SmartDashboard.putNumber("Ki: ",Ki2);
    SmartDashboard.putNumber("Kd: ",Kd2);
    SmartDashboard.putNumber("Hata-Zaman ",hata2);
}
}

```


EK 6 ROBOT SÜRÜCÜ KONTROLÜ AKIŞ DİYAGRAMI



ÖZGEÇMİŞ

Abidin CAN, Sakarya Üniversitesi Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü, Elektronik Öğretmenliği programını 2008’de tamamlamış ayrıca 2015’te Tekirdağ Namık Kemal Üniversitesi, Çorlu Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği bölümünü bitirmiştir. Lisans eğitimi stajını Almanya’nın Dortmund şehrinde yapmıştır. 2010’da Milli Eğitim Bakanlığı’nda Elektronik Öğretmenliğine başlamış ve halen çalışmaya devam etmektedir. Öğretmenliği sırasında birçok ulusal ve uluslararası proje yarışmasına danışmanlık yapmış bu yarışmalarda önemli dereceler alınmıştır.

