İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

OTONOM MOBİL ROBOT

Anabilim Dalı: Disiplinler Arası
Programı: Mekatronik Mühendisliği

YÜKSEK LİSANS TEZİ

Elektronik Müh. Alper TOZAN
518041002

Tez Danışmanı: Doç.Dr. Kenan KUTLU

HAZİRAN 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

OTONOM MOBİL ROBOT

YÜKSEK LİSANS TEZİ

Elektronik Müh. Alper TOZAN
518041002

Tezin Enstitüye Verildiği Tarih : 4 Mayıs 2007
Tezin Savunulduğu Tarih : 13 Haziran 2007

Tez Danışmanı : Doç.Dr. Kenan KUTLU

Diğer Jüri Üyeleri : Prof.Dr. Can ÖZSOY

Prof.Dr. Nejat TUNCAY (Y.Ü.)

Haziran 2007

ÖNSÖZ

Yapmış olduğum bu çalışmada benden yardımlarını esirgemeyen sayın hocam Doç. Dr. Kenan KUTLU'a; maddi ve manevi her zaman yanımda olan annem Gönül TOZAN' a, babam Mümin TOZAN' a, ablam Defne TOZAN' a, şu an yanımızda olmayan dedem Seyfettin ÖZER' e, arkadaşlarım Deniz ER' e, Utku KARAKAYA' a ve Atilla MUHİTTİN' e teşekkürü borç bilirim.

Tezimi, mezun olmamı çok isteyen merhum dedem Seyfettin ÖZER'e ithaf ediyorum.

Mayıs 2007 Alper Tozan

İÇİNDEKİLER

KISALTMALAR **v**
TABLO LİSTESİ **vi**
ŞEKİL LİSTESİ **vii**
SEMBOL LİSTESİ **ix**
ÖZET **x**
SUMMARY **xi**

1. GİRİŞ **1**

2. SİSTEMİN TANIMI **3**
2.1. Kontrol ve Veri Toplama Sistemi (MC) 3
2.1.1. Merkezi Ünitenin Devre Tasarımı 5
2.1.2. Merkezi Ünitenin PCB Tasarımı 6
2.2. Motor Sürücüsü 7
2.2.1. H-Köprüsü 8
2.2.2. Motor Sürücünün Devre Tasarımı 9
2.2.3. Motor Sürücünün PCB Tasarımı 10
2.3. Sensörler 10
2.3.1. Ultrasonik Mesafe Sensörü 11
2.3.2. Manyetik Pusula 13
2.3.3. Akım Sensörü 14
2.3.4. RF Kamera 14

3. RF DATA TRANSMİTER **16**

4. KULLANILAN YAZILIMLAR **19**
4.1. PCW C Compiler 19
4.1.1. ICD 20
4.2. Bilgisayar Arayüzü 20

5. HABERLERŞME PROTOKOLLERİ **23**
5.1. RS-232 Protokolü 23
5.2. I2C Protokolü 24

6. KULLANILAN KONTROL ALGORİTHMALARI **28**
6.1. PID Algoritması 28
6.2. Yazılımda PID Algoritması 29
6.3. Karar Algoritması 30
6.4. Fuzzy Temelli Hız Kontrol Algoritması 30

7. SİSTEMİN MODELLENMESİ **34**
7.1. DC Motor Modeli 34
7.2. Mobil Robot Modeli 36
7.2.1. Kuvvet Modeli 36
7.2.2. Sistemin Hareket Denklemleri 43
7.2.3. Sistemin Kinematik Modeli 45

8. MATLAB SİMÜLASYONLARI **49**
8.1. DC Motor Simülasyonları 49
8.1.1.Yüksüz DC Motor Simülasyonu 50
8.1.2.Yüklü DC Motor Simülasyonu 51
8.1.3. PID' li Yüklü DC Motor Simülasyonu 52
8.2. Mobil Robot Simülasyonları 55
8.2.1. Mobil Robotun Noktasal Dönüş Simülasyonu 57
8.2.2. Mobil Robotun Engelden Kaçış Simülasyonu 59

9. MOBİL ROBOT ÜZERİNDEN ALINAN ÖLÇÜMLER **62**
9.1. Mesafe Ölçümleri 62
9.2. Noktasal Dönüşte Cisimleri Algılama 64
9.3. Noktasal Dönüşte Motor Akımları 65

10. SONUÇLAR **67**

KAYNAKLAR **69**

EKLER **71**
Ek A (RoboTzanV1.38 Software) 71
Ek B (RoboTzanV1.24 Firmware) 82

ÖZGEÇMİŞ **123**

KISALTMALAR

MC	: Microcontroller
DC	: Direct Current
I	: Akım
V	: Voltaj
W	: Watt
PWM	: Pulse Wide Modulation
OPAMP	: Operational Amplifier
ADC	: Analog to Digital Converter
ICD	: Incircuit Debugger
C	: Capacitance
H	: Henry
I/O	: Input / Output
PC	: Personal Computer
Hz	: Hertz
GHz	: Giga Hertz
4WD	: 4 Wheel Drive
LED	: Light Emitting Diode
PLC	: Programmable Logic Controller
Kbit	: Kilobit
SCL	: Serial Clock Line
SDA	: Serial Data

TABLO LİSTESİ

		Sayfa No
Tablo 3.1	RF Kanal Ayarları…….....	16
Tablo 3.2	RF Güç Ayarları ……..	17
Tablo 6.1	Kural Tablosu……...	32
Tablo 6.2	Mikrokontrolör Kural Tablosu…….	33

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1 : 18F252..... 4
Şekil 2.2 : Veri Toplama Sistemi..... 4
Şekil 2.3 : 18F452 I/O..... 5
Şekil 2.4 : ICD Soketi..... 6
Şekil 2.5 : Merkezi Ünite Devre Şeması..... 6
Şekil 2.6 : Merkezi Ünite PCB Layout'u..... 7
Şekil 2.7 : DC Motor 7
Şekil 2.8 : DC Motor Ölçüleri..... 8
Şekil 2.9 : H-Köprüsü Akış Diyagramı..... 9
Şekil 2.10 : Motor Sürücü Devre Şeması..... 10
Şekil 2.11 : Motor Sürücü PCB Layout'u..... 11
Şekil 2.12 : Polaroid Sensör..... 12
Şekil 2.13 : Hitecnic sensör..... 12
Şekil 2.14 : Ses Dalga Yayılımı..... 12
Şekil 2.15 : Manyetik Pusula..... 13
Şekil 2.16 : Akım Sensör'ü Devre Şeması..... 15
Şekil 2.17 : RF Kamera..... 15
Şekil 2.18 : RF Alıcı…..... 15
Şekil 3.1 : RF Alıcı-Verici..... 18
Şekil 3.2 : RF Alıcı-Verici..... 18
Şekil 4.1 : PCW C Compiler..... 19
Şekil 4.2 : ICD Soketi..... 20
Şekil 4.3 : Bilgisayar Arayüzü..... 21
Şekil 5.1 : RS-232 Devre Şeması..... 24
Şekil 5.2 : I2C Bilgi Akışı..... 26
Şekil 5.3 : I2C Akış Diyagramı..... 27
Şekil 5.4 : Mikrokontrolör Üzerindeki I2C Yazılımı..... 27
Şekil 6.1 : Karar Algoritması Akış Diyagramı..... 31
Şekil 6.2 : Üyelik Fonksiyonu..... 33
Şekil 6.3 : PWM Çıkışı…..... 33
Şekil 7.1 : DC Motor Modeli…….....34
Şekil 7.2 : DC Motor Durum Diyagramı..... 36
Şekil 7.3 : DC Motor Blok Diyagramı..... 36
Şekil 7.4 : Paletli Araç Modeli..... 38
Şekil 7.5 : Düşük Hızlarda Paletli Araç Modeli..... 38
Şekil 7.6 : Palet' e Etkiyen Kuvvetler..... 39
Şekil 7.7 : Orta ve Yüksek Hızlarda Paletli Araç Modeli..... 46
Şekil 8.1 : DC Motor Simulink Modeli..... 49
Şekil 8.2 : Girişi Voltajı..... 50
Şekil 8.3 : Hız..... 50

Şekil 8.4 : Akım..... 51
Şekil 8.5 : Giriş Voltajı..... 51
Şekil 8.6 : Akım..... 52
Şekil 8.7 : Yüklü DC Motor Simulink Modeli..... 52
Şekil 8.8 : Giriş Voltajı..... 53
Şekil 8.9 : PID Çıkışı..... 53
Şekil 8.10 : Katedilen Mesafe..... 54
Şekil 8.11 : Motorun Hızı..... 54
Şekil 8.12 : DC Motor Akımı..... 55
Şekil 8.13 : Mobil Robot Simulink Modeli İçyapısı.....56
Şekil 8.14 : Mobil Robot Simulink Modeli…….....57
Şekil 8.15 : Voltaj Girişleri..... 57
Şekil 8.16 : DC Motor Akımı..... 57
Şekil 8.17 : Motor Hızları.....58
Şekil 8.18 : Tekerleklere Etkiyen Kuvvetler..... 58
Şekil 8.19 : Merkezkaç Kuvveti..... 58
Şekil 8.20 : Robotun Toplam Hızı..... 58
Şekil 8.21 : X Yönünde ki Hız.....59
Şekil 8.22 : Y Yönünde ki Hız.....59
Şekil 8.23 : Dönme Hızı.....59
Şekil 8.24 : Kayma Açısı.....59
Şekil 8.25 : Giriş Voltajları..... 60
Şekil 8.26 : DC Motor Akımları.....60
Şekil 8.27 : Motor Hızları.....60
Şekil 8.28 : Tekerleklere Etkiyen Kuvvetler..... 60
Şekil 8.29 : Merkezkaç Kuvveti..... 60
Şekil 8.30 : Robotun Toplam Hızı..... 60
Şekil 8.31 : X Yönünde ki Hız.....61
Şekil 8.32 : Y Yönünde ki Hız..... 61
Şekil 8.33 : Dönme Hızı..... 61
Şekil 8.34 : Kayma Açısı.....61
Şekil 9.1 : Mesafe Ölçümü 1. Test.....62
Şekil 9.2 : Hata 1. Test..... 62
Şekil 9.3 : Mesafe Ölçümü 2. Test..... 63
Şekil 9.4 : Hata 2. Test.....63
Şekil 9.5 : Mesafe Ölçümü 3. Test.....63
Şekil 9.6 : Hata 3. Test..... 63
Şekil 9.7 : Mesafe Ölçümü 4. Test..... 64
Şekil 9.8 : Hata 4. Test.....64
Şekil 9.9 : Mesafe Ölçümü 5. Test.....64
Şekil 9.10 : Hata 5. Test..... 64
Şekil 9.11 : Test Alanı..... 65
Şekil 9.12 : Mesafe Ölçümleri.....65
Şekil 9.13 : Akım Ölçümü.....66

SEMBOL LİSTESİ

$\boldsymbol{A}$: Tekerleklerin yere deyme alanı, m^2
$\boldsymbol{B}$: İki yan tekerlek arasındaki mesafe, m
$\boldsymbol{F_{cent}}$: Merkezkaç kuvveti, N
$\boldsymbol{F_{max}}$: Maksimum çekiş eforu, N
$\boldsymbol{F_{o,i}}$: Sağ ve sol tekerleklerdeki çekiş, N
$\boldsymbol{K}$: Tekerlek deformasyon katsayısı, m
$\boldsymbol{O'}$: Anlık dönüş merkezi
$\boldsymbol{R}$: Dönüş yarıçapı, m
$\boldsymbol{R_{ro,ri}}$: Sağ ve sol tekerleklerdeki boylamsal direnç kuvveti, N
$\boldsymbol{W}$: Normal yük, N
$\boldsymbol{b}$: Tekerlek eni, m
$\boldsymbol{c}$: Yüzey kohezyonu, Nm^{-2}
$\boldsymbol{g}$: Yerçekimi, ms^{-2}
$\boldsymbol{i}$: Tekerleğin kayması, $\%$
$\boldsymbol{i_{o,i}}$: Sağ ve sol tekerleklerdeki kayma, $\%$
$\boldsymbol{l}$: Sağ yada sol iki tekerlek arasındaki mesafe, m
$\boldsymbol{m}$: Kütle, Kg
$\boldsymbol{r}$: Tekerlek yarıçapı, m
α : Kayma açısı, rad
$\dot{\phi}$: Açısal hız; $rads^{-1}$
μ_l : Yatay direnç katsayısı
μ_r : Boylamsal direnç katsayısı
μ : Açısal wall-soil sürtünmesi, rad
$\omega_{o,i}$: Sağ ve sol tekerleklerdeki hız, $rads^{-1}$
ϕ : Robotun açısal yönü, rad

OTONOM MOBİL ROBOT

ÖZET

Projede, insanların giremediği bölgelere girerek bölgeyi araştıracak, elde ettiği görsel ve dijital verileri bilgisayara iletecek ve bilgisayardan gözlenebilecek bir mobil robot gerçekleştirilmiştir.

Mobil robotun tüm kontrol elemanlarının uyumlu bir yapıda bütünleşmesi sağlanmıştır. Bu sayede algılama ve hareket bütünleştirilerek, hesaplamada sensörlerden gelen kaynaklar kullanılmıştır. Robot vereceği kararlarda sensörlere bağımlı kalacağından robota davranışsal yapay zeka kazandırılmıştır. Veriler mobil robot üzerindeki 4 adet ultrasonik mesafe sensörü, 1 adet dijital pusula, 2 akım sensörü ve 1 kameradan elde edilmektedir. Veriler mobil robot üzerinde bulunan bir RF verici ile iletilmektedir. Gönderilen veriler PC üzerinde bulunan bir alıcı vasıtası ile alınıp bilgisayarda işlenmektedir. Ultrasonik mesafe sensörleri sayesinde cisimler algılanmakta, mobil robota olan uzaklıkları belirlenmektedir. Dijital pusuladan yön verisi elde edilmektedir. Mobil robot üzerindeki MC aldığı mesafe ve yön verileri doğrultusunda kendi kararlarını vererek yönünü bulmaktadır. İstenildiği taktirde bilgisayardan verilecek komutlar ile mobil robot durdurulabilmekte ve kamera vasıtası ile objeler gözlenebilmektedir. 2 adet bulunan akım sensörleri vasıtası ile mobil robot istenmeyen akım yüklenmelerini optimize edebilmektir. Bu işlem yardımı ile robotun pil ömrünün uzatılması planlamıştır.

Mobil robotun bilgisayarda modeli oluşturulmuş, değişik senaryolarda simülasyonlar gerçekleştirilmiştir. Elde edilen sonuçlar robot üzerinden ölçülen değerlerle karşılaştırılmıştır. Mesafe sensörünün değişik cisimler üzerinde mesafe ölçümleri yapılıp yorumlanmıştır. Yapılan testler doğrultusunda sensörün ölçüm doğruluğu, objenin yüzeyinin sensöre olan açısı ile değişebileceği sonucuna varılmıştır.

Mobil robot araştırma ve kurtarma görevlerinde kullanılabilecek şekilde tasarlanmıştır. İleriki projelerde değişik kontrol algoritmaları robota adapte edilebilir.

AUTONOMOUS MOBILE ROBOT

SUMMARY

I'm aiming to develop an autonomous 4WD mobile robot which can operates different hazardous environments, collects visual and digital data, transmits these data to PC.

All parts of robot can be designed and assembled as compact as possible. Robot makes its decision base on sensor so that artificial intelligence is added to its firmware. I'm aiming to develop an autonomous 4WD mobile robot which can operates different hazardous environments, collects visual and digital data, transmits these data to PC. These data are collected from 4 ultrasonic range finders, a digital compass, 2 current sensors and an analog camera on mobile robot. Data is transmitted with RF transmitter and received by RF receiver and processed by PC. 4 ultrasonic range finders sense and measure the distance between objects and mobile robot. Direction data is collected from digital compass. Data are processed and objects are localized with computer. Mobile robot can make its own decisions with respect to sensor data. Mobile robot is designed that either works autonomous or manually from computer. In manual mode, operator can give orders to a robot for observing the environment with RF camera. 2 current sensors inform robot about current values and microcontroller optimizes the current for longer battery life time.

Mathematical model of mobile robot is obtained from tracked vehicle model. DC motors which are used in this study, are modeled and adapted into mobile robot model. System is simulated and PID control algorithm is tested. Simulation data is observed and criticized with data collected from mobile robot. Distance measurements are made with different objects width and different objects shaped. It is observed that sensors measurements are changed with surface angle of object.

Mobile robot can be used in search and rescue operations. Also new control algorithms can be adapted for future works.

1 GİRİŞ

Gün geçtikçe artan ve günlük hayatımızın vazgeçilmezleri arasında yerini alan mobil teknolojiler kendilerine sürekli olarak değişik alanlarda yer bulmaktadır. Bugün bu teknolojiler evlerimize giren cep telefonları, radyo cihazları, uydu cihazları, savaş sanayisinde kullanılan bomba imha robotları, arama kurtarma gibi değişik platformlarda kullanılan robotlar ve hatta uzay araştırmalarında bile kullanılan insansız keşif araçları da bu değişik alanlarda kendi varlığının gerekliliğini kabul ettirmiş uygulamalardır.

Mobil araçlardaki sensörler vasıtasıyla çevrelerinde olup biten değişimlerden kolayca haberdar olabilmektedirler.

Mobil robot teknolojisi günümüz Türkiye'sinde kendisine fazla yer bulamamıştır. Bunun sebebi araştırma maliyetinin çok yüksek olması ile gösterilebilir.

Otonom yapıya sahip olması beklenen herhangi bir mobil robotun sahip olması gereken belli başlı özellikler;

- Bulunduğu ortamı tasarlandığı amaç doğrultusunda algılayabilmesi,
- Beklendik ve beklenmedik durumlarda kendi kararlarını verebilmesi,
- Elde ettiği verileri insanlarla paylaşabilip, gerektiğinde kontrol edilebilmesi.

Eğer otonom olması planlanan bir gezgin robotun en önemli sisteminin karar verme mekanizması olduğu düşünülürse, bulunduğu ortamdaki cisimleri algılanmasında sensörlere çok önemli görevler düşmektedir. Sensörler robotun etrafını öğrenmesine olanak sağlamalı, gerçek-zamanlı olmalı, aynı zamanda donanımsal olarak ekonomik olmalıdır.

Robotun mekanik tasarımının iyileştirilmesine olanak tanımak için mekanik bileşenlerinin çok iyi seçilmesi ve yerleştirilmesi (az sürtünme, az boşluk vb.) hem de robotu hareket ettirecek motorların yüksek tork/ağırlık ve yüksek tork/güç özelliklerine sahip olmaları gerekmektedir. Buna ek olarak, bilginin işlenmesi ve karar mekanizmasının çalışması için gerekli işlemsel güç, elektronik donanımına uygun olmalı ve robotun gerçek zamanlı iş yapmasına izin vermelidir. Son olarak ta,

bütün bunları gerçekleştirirken, robotun algıladıklarını paylaşması ve gerektiğinde insan müdahalesine izin vermesi gerekmektedir. Bu proje çerçevesinde, tüm bu gereksinimlere yanıt verebilecek bir robot tasarımı yapılması amaçlanmıştır.

Projeyi gerçekleştirirken şu aşamalar takip edilmiştir.

1. Mobil robotun hangi özelliklere sahip olacağı, hangi işlemleri yapacağı kararlaştırıldı

2. Proje için gerekli donanım ve yazılım belirlenmiştir.

3. İhtiyaçlar temin edilmiştir.

4. Elektronik kartlar tasarlanmış ve sisteme yerleştirilmiştir.

5. Gerekli simülasyonlar yapılmıştır.

6. Simülasyonlar sonucu kontrol algoritması yazılıp mikrokontrolöre yüklenmiştir.

2 SİSTEMİN TANIMI

Mobil robot için non-holonomic 4 çeker bir platform seçilmiştir. Dayanıklı olması için mobil robotun kasası lazer kesim lexanden üretilmiş ve alüminyum çıtalar ile desteklenmiştir. 4 çeker sistem uygulaması robotun kütlesinin fazla olacağından ve yüksek tork gerektireceğinden diferansiyel yerine dört tekerleğe dört ayrı dişli kutusu içeren motor yerleştirilmiştir. İstenilen çekiş gücünü sağlamak için yumuşak plastikten yapılma yüksek esneme katsayısına sahip arazi tipi lastik kullanılmıştır.

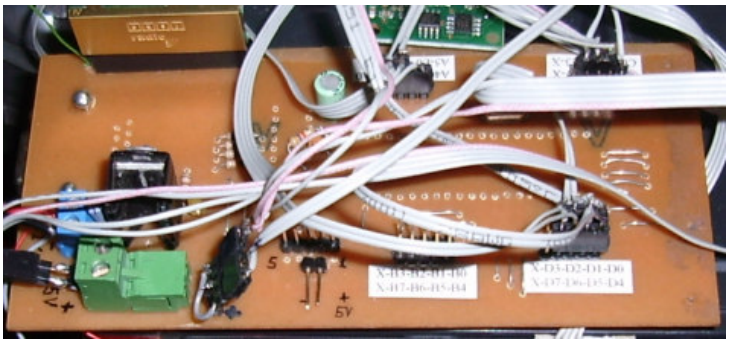
Otonom çalışma için gerekli elektronik devreler zorlu şartlara uyumlu, compact bir dizayna sahip ve yüksek veri işleme gereksinimlerini sağlayabilecek şekilde tasarlanmıştır. Mikrokontrolör olarak Microchip firmasının Pic ailesinden Harward mimarisi ile üretilmiş 18F452 modeli seçilmiştir. Arazide dolaşırken objeleri tespit edip mesafeleri ölçmek için dört tane Devantech firmasının üretmiş olduğu SRF05 modeli ultrasonik mesafe sensörü yerleştirilmiştir. Gidilen yönü algılamak için yine Devantech firmasının üretmiş olduğu CMPS03 dijital pusulası mobil robotun merkezine yerleştirilmiştir. Motorlardan geçen akım miktarını gözlemlemek için kullanılan sensör LMD18200 entegresinde bulunmaktadır. Buradan elde edilen veri mikrokontrolörün iki adet ADC ile okunmaktadır. Mobil robotun önüne görsel teması sağlamak için bir adet 1.2 Ghz Excel Link marka analog kamera yerleştirilmiştir. Görüntü, tünere sahip bir ekranda ya da bilgisayarın tv kartında izlenmektedir. Sensörlerden elde edilen veri Keymark marka 434 Mhz RF alıcı ve veri ile sağlanmaktadır.

2.1 Kontrol ve Veri Toplama Sistemi (MC)

Mobil robot projesinin beyni mikrokontrolördür. Mikrokontrolör mantıksal fonksiyonları bir araya toplayıp içinde mikro işlemci barındıran bir cihazdır. Genelde başlı başına bir sistem olmayıp dışardan klavye, ışık sensörü gibi ilaveler eklenerek amaca ulaşılır. Uygun bir mikrokontrolör, devresi için gerekli olan her şeyi

barındırır. MC lar tek başlarına kullanılabileceği gibi, başka bir sistemi yönetmek amacıyla da kullanılabilir.

Şekil 2.1: 18F252



Şekil 2.2: Veri Toplama Sistemi

Mobil robot projesinde verileri toplamak, işlemek, değerlendirme yapıp karar almak ve bilgisayara yönlendirmek için kullanılmıştır. Mikrokontrolör kullanmanın getirdiği avantajları aşağıdaki gibi sıralayabiliriz.

1. CPU ile yapılacak devrenin gerektirdiği ram, rom, I/O, dekoder, enkoder, adc, dac gibi dış bileşenleri içinde barındırarak yerden zamandan ve emekten tasarruf sağlar.

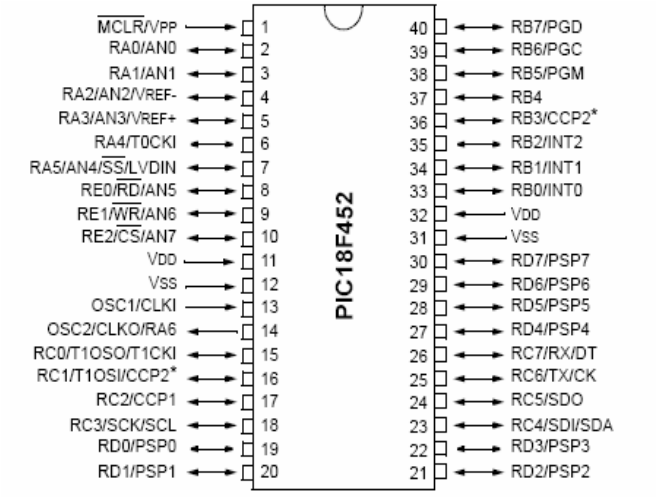
2. Assembly, C, Basic, Pascal gibi bir çok programlama dili ile programlana bilirler.

3. 100000 sefere kadar yeniden programlanabilirler.

4. PLC' ler kadar yer kaplamazlar.

5. Maliyeti az, temini kolaydır.

Mobil Robot projesinde merkezi ünite için MicroChip firmasının 18F452 mikrokontrolörü seçilmiştir. (Şekil 2.3)

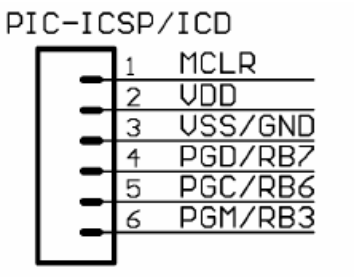


Şekil 2.3: 18F452 I/O

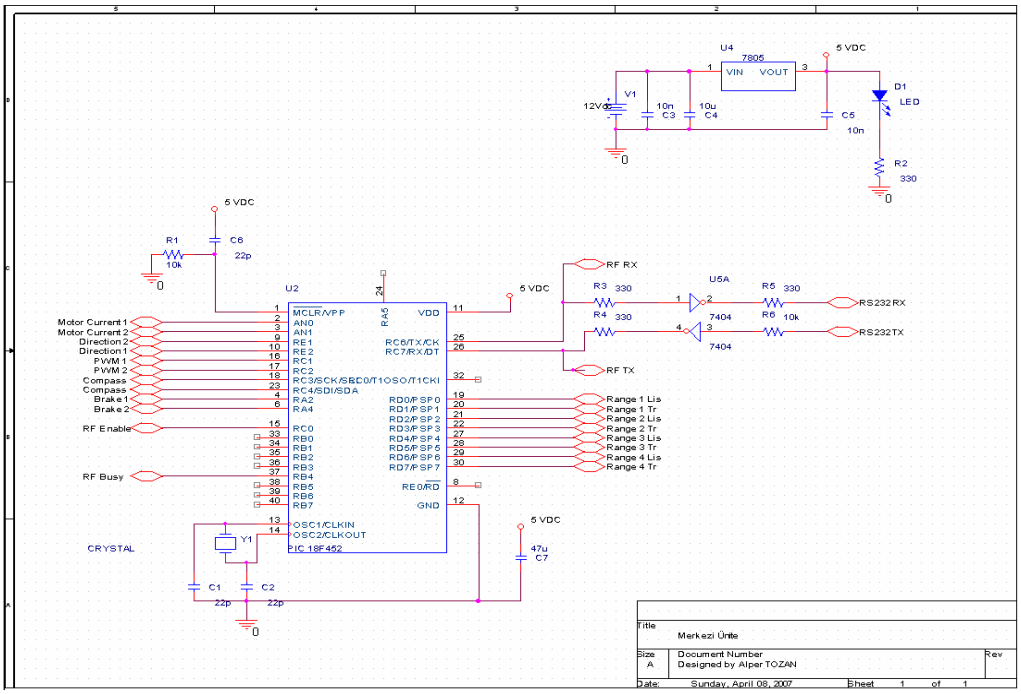
Sistem 0 ile 40 Mhz arasında çalışma olanağı sunmaktadır. 33 adet I/O' u bulunmaktır. Bunların sekiz tanesi analog giriş olarak ayarlanabilmektedir. Sistemde iki adet 16 bitlik 1 adet 8 bitlik timer bulunmaktadır. Timerlar vasıtası ile ultrasonik sensörler için zaman ölçümü yapılabilmektedir. Sistemin kararlılığını artırmak amaçlı WDT kullanılmıştır. Toplanan verileri iletmek amaçlı üzerinde hem RF modülü hem de 9 pin RS-232 soketi bulunmaktadır. Merkezi ünite beslemesi 12 VDC ile yapılmakta ve üzerinde üç adet 5 VDC, bir adet 9 VDC çıkışı bulunmaktadır. Üniteye yazılım yüklemek ve debug yapabilmek için ICD/ICSP soketi yerleştirilmiştir. (Şekil 2.4)

2.1.1 Merkezi Ünitenin Devre Tasarımı

Merkezi üniteyi tasarlarken mümkün mertebe ufak, sarsıntılara, darbelere dayanıklı ve ilerde eklenebilecek sensör ve diğer bileşenlerle genişleme imkanı sağlayabilecek bir yapıya sahip olması planlanmıştır. Devre ilk önce Orcad programında tasarlanmış bais voltaj ve akımları için programda simülasyon yapılmıştır. Devre şeması şekil 2.1 de görülmektedir.



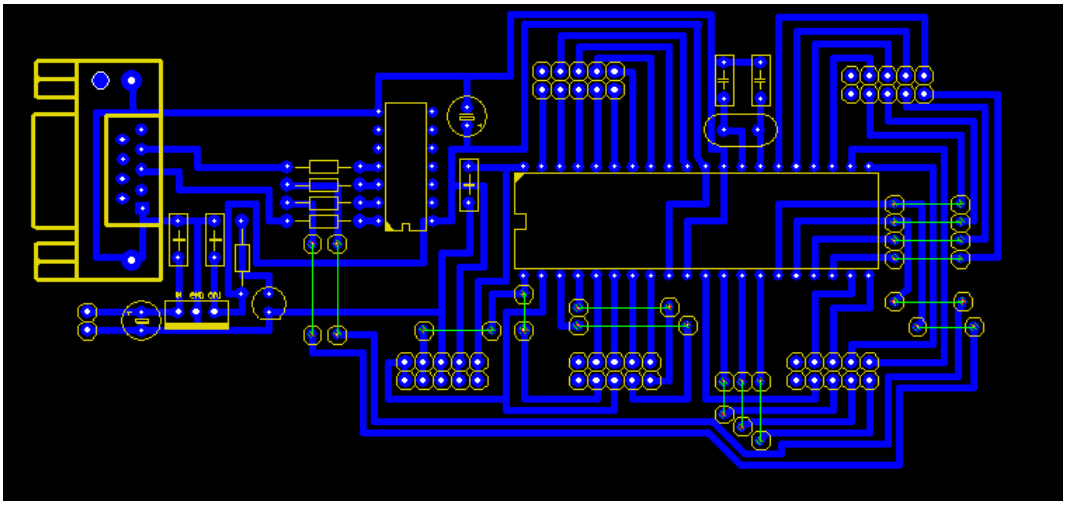
Şekil 2.4: ICD Soketi



Şekil 2.5: Merkezi Ünite Devre Şeması

2.1.2 Merkezi Ünitenin PCB Tasarımı

Merkezi ünitenin PCB sini tasarlarken sistemin yakınında bulunan RF vericinin 434 Mhz lik frekansı ve kameranın 1.2 Ghz lik frekansından etkilenmiyecek bir yapıda tasarlanmış veri yolları mümkün mertebe yüksek frekanslı bölgelerden uzak tutulmuştur. Devre üzerinden aktarılan bilgilerin devrenin besleme hatlarından etkilenmemesi için veri yolları ve besleme hatları PCB nin değişik yerlerine yerleştirilmişlerdir. PCB tasarımında Sprint-Layout programı kullanılmıştır. PCB şeması şekil 2.2 de görülmektedir.



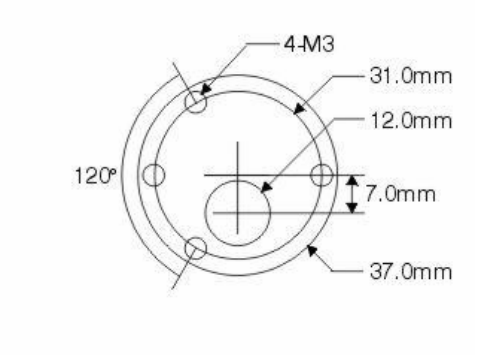
Şekil 2.6 : Merkezi Ünite PCB Layout'u

2.2 Motor Sürücüsü

Robotu mobilize etmek için dört adet 50:1 dişli kutusuna sahip 175 rpm dönüş hızında DC motor kullanılmıştır. (Şekil 2.7)

Şekil 2.7: DC Motor

Her bir motor yüksüz çalıştırıldığında 200 mA akım çekmektedir. Şaftın kilitlendiği durumlarda ise çekilen akım motor başına 3.8 A' e çıkmaktadır. Bu değer hazır motor sürücü entegrelerinin çoğunun ulaşamayacağı bir değer olduğundan dolayı motorlar en fazla %70 kapasitede çalıştırılmaktadır. Motorları sürmek için tasarlanan devre sürekli olmak kaydı ile motor başına 3 A' i sağlayabilmektedir. Robotun kalkış ve yön değiştirişlerinde oluşabilecek anlık akımlardan zarar görmemesi için anlık 6 A' i sağlayabilecek şekilde tasarlanmıştır.



Şekil 2.8: DC Motor Ölçüleri

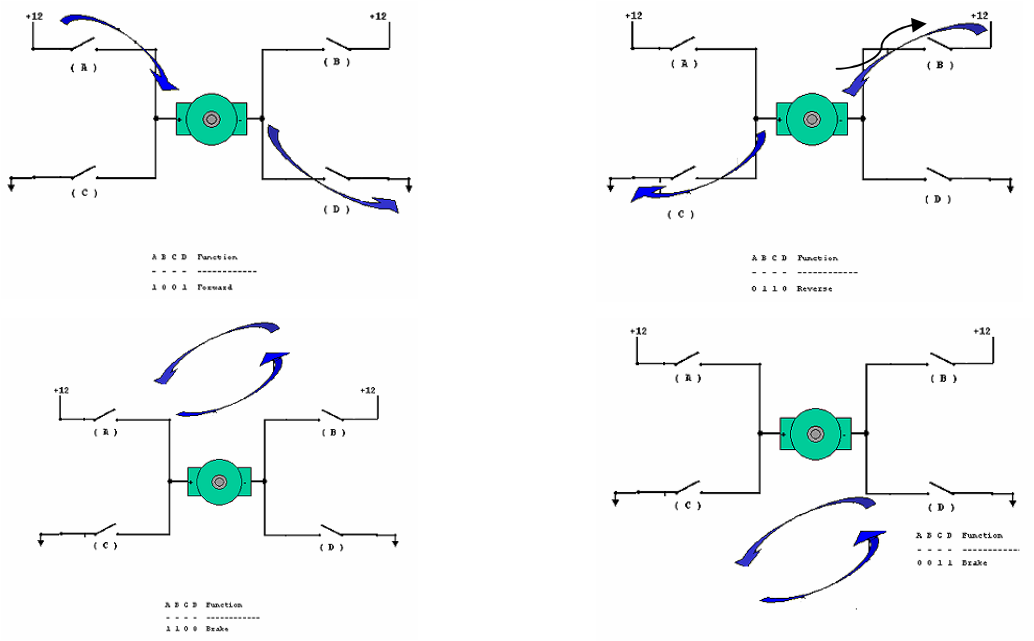
Motor sürücüsü başlık 2.2.2 de açıklanmış olup motorlar için 12 VDC, logic girişler için 5 VDC beslemeye ihtiyaç duymaktadır. Motor sürücüsü üzerinde dört motor çıkışı, her çift motor için bir logic hız girişi, bir logic yön girişi ve bir logic fren girişi bulunmaktadır. Fren işlemi motorların yön değiştireceği zaman uygulanmakta, motor üzerinde birikmiş olan enerjiyi motor üzerinde harcayarak devreye zarar vermesi engellenmiştir. Motor sürücü üzerinde iki adet akım çıkışı bulunmakta ve her birinin hassasiyeti 377 µA/A dir. Akım çıkışlarına bağlanan 2.2 $k\Omega$ luk direnç vasıtası ile 0-5 V a çevrilmekte ve mikrokontrolör vasıtası ile okunmaktadır.

2.2.1 H-Köprüsü

Motor sürücü yapı olarak H-köprüsü olarak tasarlanmış. H-köprüsü ismi devrenin H harfine benzemesinden gelmektedir. Genel olarak DC motorları sürmek için kullanılmaktadır. En temel hali ile dört adet anahtarlama elemanı ve motordan oluşmaktadır.

Motoru ileri yönde döndürmek için A ve D anahtarlarını kapalı B ve C anahtarlarını açık konuma getirmek gerekmektedir. Bu konumda motoru döndürmek için gerekli akım A anahtarı üzerinden akıp motora ulaşmakta ve D anahtarı üzerinden devreyi tamamlamaktadır.

Motoru geri yönde döndürmek için B ve C anahtarlarını kapalı A ve D anahtarlarını açık konuma getirmek gerekmektedir. Bu konumda motoru döndürmek için gerekli akım B anahtarı üzerinden akıp motora ulaşmakta ve C anahtarı üzerinden devreyi tamamlamaktadır.



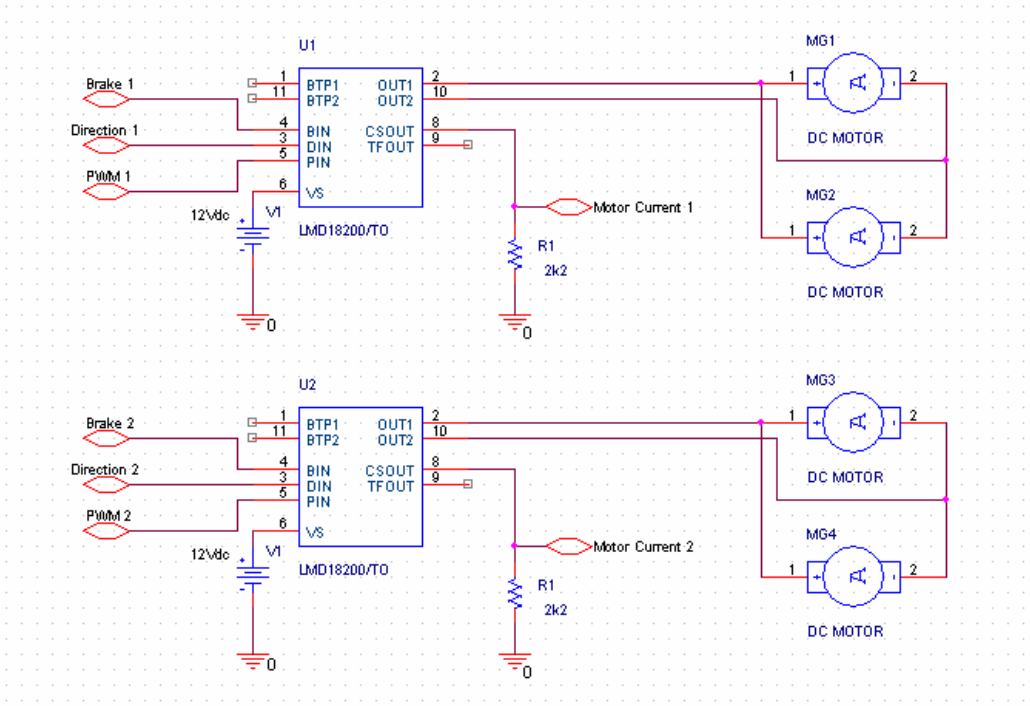
Şekil 2.9: H-Köprüsü Akış Diyagramı

Eğer A ve B anahtarını kapalı C ve D anahtarlarını açık konuma getirdiğimizde ya da A ve B anahtarını açık C ve D anahtarını kapalı konuma getirdiğimizde motor fren işlemi yapmakta ve üzerinde biriken enerjiyi kendi üzerinde harcamaktadır. Buradaki işlem sadece motoru durdurmak değil aynı zamanda motor üzerinde oluşan akımın devreye zarar vermesini önlemektir.

2.2.2 Motor Sürücüsünün Devre Tasarımı

Motor sürücüsünü tasarlarken göz önünde tutulan en önemli kriter dört adet DC motorun ihtiyaç duyduğu akımı sağlayabilmesi olmuştur. Robotun ufak olması ve yüksek akıma ihtiyaç duyması tasarım aşamasında karşılaşılan en büyük zorluklardan biridir. Yapılan araştırmalar sonucu hem ufak yapıya sahip hem de ihtiyaç duyulan akımı karşılayabilecek LMD18200 entegresi kullanılmıştır. Entegre mimari olarak H-köprüsü ne sahiptir. Anahtarlama elemanı olarak Mos-FET ler kullanılmıştır.

Devre şeması ORCAD programında çizilmiş ve bais voltaj ve akımları için simülasyon yapılmıştır. (Şekil 2.6)



Şekil 2.10: Motor Sürücü Devre Şeması

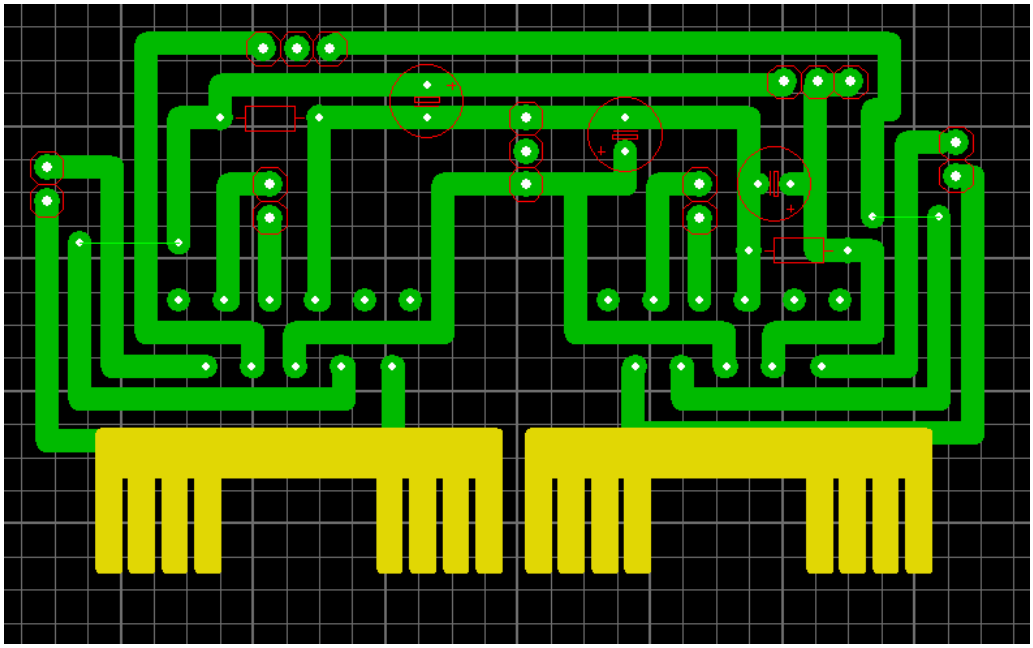
2.2.3 Motor Sürücüsünün PCB Tasarımı

Motor sürücü ünitesinin PCB' sini tasarlarken motorların ihtiyaç duyduğu akımların geçtiği yollar ısınma ve yanma olasılıkları göz önünde bulundurulmuş ve akımların geçtiği yollar geçen akımları karşılayabilecek kalınlıkta yapılmıştır. Geçen akımlardan oluşacak manyetik alanın logic seviyedeki yön, hız ve fren bilgilerini etkilememesi için bilgi yolları ile akım yolları birbirlerinden uzak tutulmuştur. PCB tasarımında Sprint-Layout programı kullanılmıştır. PCB şeması şekil 2.7 de görülmektedir

2.3 Sensörler

Mobil robotlar dış dünyayı algılamak için sensörlerini kullanırlar. Bilindiği gibi robotlarda kullanılan sensörler doğal canlılardan esinlenerek tasarlanmışlardır. Örneğin kedilerin bıyıklarından esinlenerek dokunmatik sensörleri, yarasaların gece görüşünde kullandığı ultrasonik seslerden esinlenerek ultrasonik sensörler geliştirilmiştir. Sensörler bir dış uyarıyı işlenebilen, ölçülebilen elektrik sinyallerine dönüştürürler. Sensörlerin verilerini kullanabilmek için her tip sensörün uygun bir

arayüzle robotun kontrol kartına bağlanması gerekir. Mobil robotu otonom hale getirmek ve beş değişik parametreyi algılayabilmek için dokuz adet sensör kullanılmıştır.



Şekil 2.11: Motor Sürücü PCB Layout'u

2.3.1 Ultrasonik Mesafe Sensörü

Ultrasonik ses insan kulağının algılayamadığı yüksek frekanslı seslerdir. 20 kHz ila 500 kHz frekans arası ses dalgaları bu sınıftandır. Ultrasonik sesler doğrusal (direksiyonel) yayılım özelliklidir. Sesin frekansı yükseldikçe, doğrusal yayılım özelliği artar. Bu özellik sayesinde, bir kaç cm' den 30m' ye kadar olan mesafeler ultrasonik aygıtlarla ölçülür. Sesin deniz seviyesindeki yayılma hızı 346 m/s dir. Bu yayılma hızı yükseklik, ısı, nem oranı ve atmosfer basıncına göre değişebilir. Örneğin ses hızları 0 'C (273k) da 330 m/s; 18 'C de 341 m/s, 20 'C oda sıcaklığında ve kuru havada ise 343 m/s dir.

Mobil robot projesinde kullanılan ultrasonik sensörde 40kHz frekanslı bir ses atım sinyali, bir sensör (Verici) aracılığıyla gönderilir. Ultrasonik ses dalgası önündeki bir nesneye, bir engele çarpıp geri yankılandığında diğer bir sensör (Alıcı) tarafından algılanır. Bir elektronik devre aracılığıyla gönderilen ses atımının çıkışından engele çarpıp geri gelen eko sesin alınması arasındaki zaman süresini sayarak arada geçen

zaman ölçülür. Bu zaman ikiye bölünür, ses hızı ile çarpılarak uzaklık ölçümü yapılır.

Şekil 2.12: Polaroid Sensör

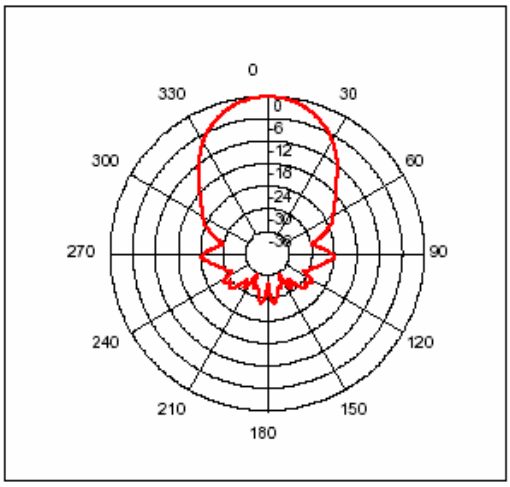
Şekil 2.13: Hitecnic sensör

$$D = \frac{Vt}{2}$$

$D =$ Uzaklık

$V =$ Ses Hızı

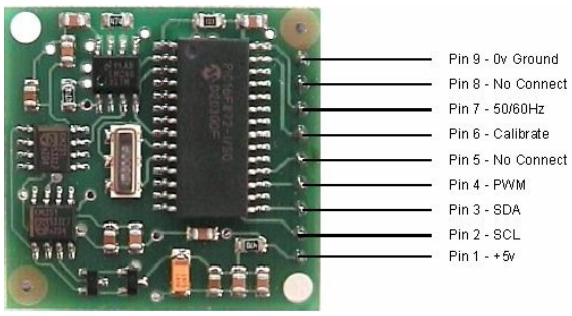
$t =$ Ses atımının çıkışından engele çarpıp eko olarak algılanana kadar geçen zaman;



Şekil 2.14: Ses Dalga Yayılımı

2.3.2 Manyetik Pusula

Manyetik pusulalar MEMS teknolojisi ile üretilmiş dijital sensörlerdir. Çalışma prensibi iğneli pusulardaki gibi dünyanın manyetik alanına dayanır. İçerdiği mekanik bileşen dünyanın manyetik alanından etkilenir ve hareket eder. Bu hareket entegrenin içerdiği elektronik bileşenler tarafından algılanır ve dijital bilgiye çevrilir. Bu bilgi kontrol devresi tarafından desteklenen iletişim protokolü vasıtasıyla elde edilir.



Şekil 2.15 Manyetik Pusula

Projede manyetik pusula hem yön bilgisini elde etmek hem de cisimlerin robota hangi açıda olduğunu belirlemek için kullanılmıştır. Seçilen manyetik pusula bilgi iletimini PWM ya da I2C protokolleri üzerinden iletebilme yeteneğine sahiptir. İlk kez çalıştırılma esnasında gereken kalibrasyon işlemi I2C protokolü üzerinden yapılmış ve gerçeğe daha yakın değerler alınması sağlanmıştır.

Manyetik pusulanın PWM çıkışı için veri aralığı 0-255, I2C protokolü için veri aralığı 0-3599 olduğundan daha hassas bilgi alabilmek açısından I2C protokolü seçilmiştir.

Avantajları

1 Çözünürlüğünün yüksek oluşu

2 20 mA lik düşük güç tüketimi

3 I2C protokolü üzerinden 1 Mhz e kadar veri transferi

4 Düşük maliyeti

Dezavantajları

1 Mobil bir platforma yerleştirildiğinden dolayı meyilli arazilerde doğru bilgi ölçümleyememe olasılığı

2 Araç üzerinde bulunan dört adet DC motorun oluşturduğu manyetik alandan etkilenme olasılığı

2.3.3 Akım Sensörü

Motorların üzerinden geçen akımları ölçmek için LMD18200 entegresinin üzerinde bulunan akım sensörleri kullanılmıştır. Bu sensörler 0-6 A aralığında 377 µA/A hassasiyetinde çalışabilmektedirler. Sensörler akım çıkışlı olduğundan elde edilen akım değerini voltaj bilgisine çevirmek için %1 hassasiyetinde 2.2 $k\Omega$ luk direnç kullanılmıştır. Dolayısı ile 0-6 A aralığındaki akım ölçümünü 0-5 volt aralığına indirgenmiştir. Elde edilen analog akım bilgisi mikrokontrolör üzerinde bulunan ve 10 bitlik bir ADC yardımı ile okunmaktadır. ADC yolu ile elde edilen bilginin hassasiyeti ise aşağıdaki formülden elde edilmektedir.

$$Hassasiyet = \frac{V}{2^n} \qquad \textbf{(2.2)}$$

V değeri ölçülecek maksimum değer, n ise ADC nin bit sayısıdır. ADC miz 10 bitlik ve ölçeceğimiz maksimum 5 V = 6 A olduğuna göre;

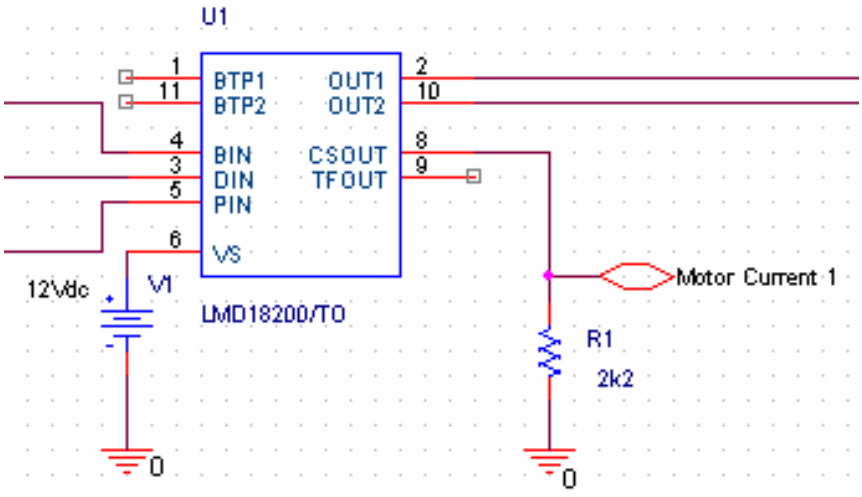
$$Hassasiyet = \frac{6}{2^{10}} \cong 0,00585A \qquad \textbf{(2.3)}$$

Görüldüğü gibi burada elde edilen hassasiyet sonucu 5.85 mA' dir. Bu da demektir ki 0-6 A arasında algılayabildiğimiz en ufak değişiklik 5.85 mA ile sınırlıdır.

2.3.4 RF Kamera

Mobil Robotta, kamera robottan görsel bilgi almak için kullanılmıştır. Kameraya entegre mikrofonu sayesinde çevredeki sesleri de bilgisayara aktarabilmektedir. Görüntü formatı Pal formatındadır. Görüntüyü 1.2 Ghz frekansından alıcıya yollamaktadır. 50 Hz gibi yüksek bir tarama frekansına sahip olduğundan dolayı

robotun bulunduğu ortam akıcı olarak gözlemlenebilmektedir. Seçilen kameranın çözünürlüğü ise 628X582 dir. Dikkat edilmesi gereken nokta kameranın vericisinin kapalı alanlarda 50 m, açık alanlarda 100 metre gibi bir kapsama alanı olmasıdır. Bu değerler geçildiği taktirde görüntü aktarımı aksamaya başlamaktadır. RF kamera için gerekli olan 9 VDC besleme gerilimi 7809 regülatör entegresinden elde edilmektedir.



Şekil 2.16: Akım Sensörü Devre Şeması

Şekil 2.17: RF Kamera

Şekil 2.18: RF Alıcı

3 RF DATA TRASMITER

Sensörlerden elde edilen bilgiler mikrokontrolör vasıtasıyla işlenip RF verici üzerinden bilgisayara iletilir. Kullanılan RF vericinin fabrika çıkış frekansı 434 Mhz olmasına rağmen ileride eklenebilecek diğer RF modülleri olasılığını düşünerek, frekansların çakışmasını engellemek amaçlı 10 değişik kanaldan veri iletebilecek şekilde ayarlanabilmektedir. RF vericinin kanalı yardımcı bir program yardımı ile ayarlanmaktadır. Kanalı ayarlamak için ER_CMD#CX komutunu X gelecek yere kanal numarasını yazarak vericiye yollamak gerekmektedir. Kanal numaraları ve kanalların frekansları aşağıdaki tabloda verilmiştir. (Tablo 3.1)

Tablo 3.1: RF Kanal Ayarları

RF Kanal Ayarları		
Komut	**Kanal No**	**Frekans**
ER_CMD#C0	Kanal 0	433.23 MHz
ER_CMD#C1	Kanal 1	433.30 MHz
ER_CMD#C2	Kanal 2	433.45 MHz
ER_CMD#C3	Kanal 3	433.55 MHz
ER_CMD#C4	Kanal 4	433.68 MHz
ER_CMD#C5	Kanal 5	433.83 MHz
ER_CMD#C6	Kanal 6	433.88 MHz
ER_CMD#C7	Kanal 7	434.00 MHz
ER_CMD#C8	Kanal 8	434.15 MHz
ER_CMD#C9	Kanal 9	434.35 MHz

RF alıcı-vericilerde kullanılan anten boyu 16.7 cm dir. Anten boyu frekansla ters orantılı olduğundan dolayı frekans arttıkça anten boyuda kısalmaktadır. Anten üzerinden gönderilen radyo frekansının gücü maksimum 10 mW tır. Bu değer yine yardımcı bir program kullanarak 1 mW tan 10 mW a kadar 10 değişik güç seviyesine ayarlanabilir. Ayarlanan güç seviyesi ne kadar yüksek olursa veri iletim mesafesi de o kadar artacaktır. 10 mW güçte açık alanda ortalama veri iletim mesafesi 250 m dir.

Radyo frekansının gücü, ER_CMD#PX komutunu X gelecek yere ayarlamak istediğimiz güç seviyesine denk gelecek rakamı yazıp vericiye yollamak gerekmektedir. Güç seviyeleri aşağıdaki tabloda verilmiştir. (Tablo 3.2)

Tablo 3.2: RF Güç Ayarları

RF Güç Ayarları	
Komut	**Güç Seviyesi**
ER_CMD#P0	1 mW
ER_CMD#P1	2 mW
ER_CMD#P2	3 mW
ER_CMD#P3	4 mW
ER_CMD#P4	5 mW
ER_CMD#P5	6 mW
ER_CMD#P6	7 mW
ER_CMD#P7	8 mW
ER_CMD#P8	9 mW
ER_CMD#P9	10 mW

RF verici, 2400 baud ile 38400 baud arasında standart baud oranlarını desteklemesi yanında istendiğinde kullanıcı tarafından tanımlanmış baud oranlarını da iletebilmektedir. İletim protokolü 1 başlama bit' i, 8 data bit' i ve 1 bitiş bit' i olarak ayarlanmıştır. Kendi içinde bulunan 180 byte' lık buffer sayesinde 1 seferde maksimum 180 byte gönderebilmektedir. Transfer zamanı ise aşağıdaki formül ile hesaplanmaktadır.

$$t = 13,2mS + (n.Byte).0,8mS + 1,6mS \qquad \textbf{(2.4)}$$

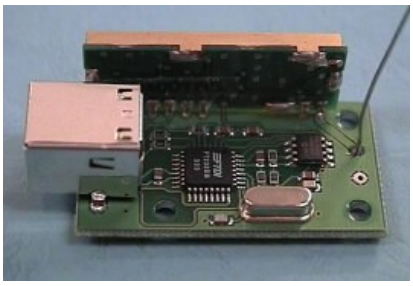
Formülde t zamanı, n ise gönderilen byte adetini belirtmektedir. Bilgi gönderimi sırasında vericinin Busy bacağı logic 1 seviyesine çıkar ve 1 seviyesinde kaldığı sürece gönderilecek başka veriler kaybolur. Mikrokontrolör üzerindeki yazılımda sürekli vericinin Busy bacağı kontrol edilmekte ve Busy bacağının logic 0 yani meşgul olmadığı zamanlarda veri gönderilmektedir.

Veri yollama protokolünde her bir bilgi başına 4 byte gönderilmektedir. 1. byte verinin ne olduğunu içermekte, 2. byte verinin MSB byte' ı, 3. byte verinin LSB

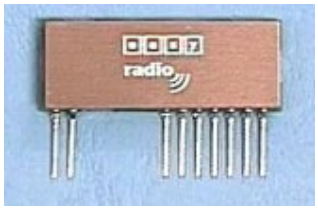
byte' ı ve son byte gönderilen bilginin doğru olup olmadığını kontrol etmek amaçlı kontrol toplamıdır (CHECKSUM).

CheckSum algoritması ilk 3 byte' ın toplamının 4. byte yazılmasından oluşmaktadır. Gönderilen bilgi, bilgisayar arayüzü için yazılan programda kontrol edilmekte ve toplamlar uyuştuğu taktirde doğru bilgi olduğu anlaşılmaktadır.

Kullanılan RF birim vericinin yapısında olduğu için bilgi akışı mikrokontrolör ile kısıtlı kalmayıp bilgisayardan da mobil robota bilgi akışı sağlanabilmektedir.



Şekil 3.1: RF Alıcı-Verici



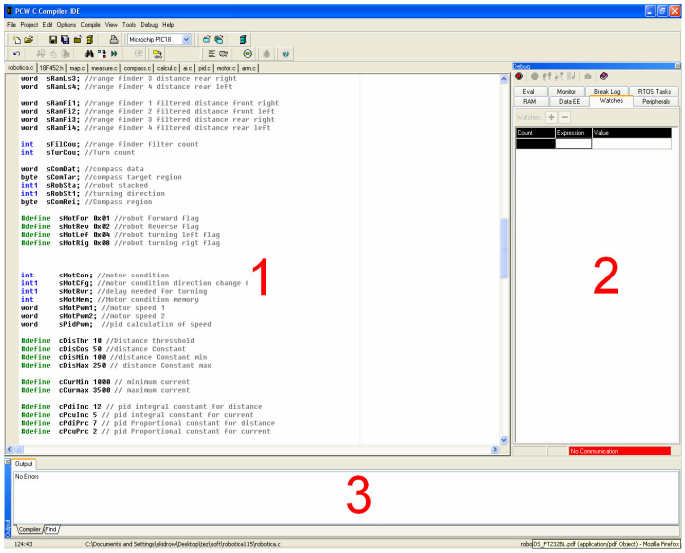
Şekil 3.2: RF Alıcı-Verici

4 KULLANILAN YAZILIMLAR

Projenin yazılım kısmını gerçekleştirirken iki ana program kullanılmıştır. Bunlardan birincisi mikrokontrolör üzerinde yazılım yazma amaçlı diğeri mikrokontrolörün elde ettiği verileri bilgisayarda gözlemek amaçlı kullanılmıştır.

4.1 PCW C Compiler

Mobil robot üzerindeki mikrokontrolörü programlarken C dili kullanılmıştır. Program PCW C Compiler programı ile yazılmış ve geliştirilmiştir. Bu programın seçilmesindeki amaç programın kullanıcı dostu olması ve programı adım adım koşturabilme olanağı sunmasıdır. Programın üç ana ekranı bulunmaktadır. Ana ekran (1. Ekran) programın yazıldığı ekrandır. Burada programı koştururken istenilen noktaya durma noktası eklemek mümkündür. Program o noktaya geldiğinde otomatik olarak duracak ve o zamana kadar elde ettiği verileri kullanıcıya gösterecektir. Program şekil 4.1 de görülmektedir.



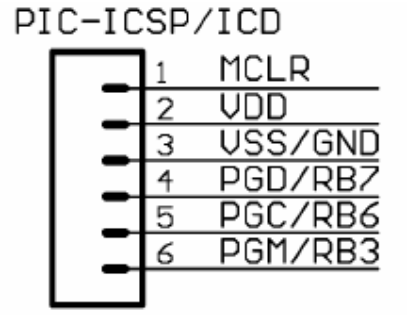
Şekil 4.1: PCW C Compiler

2. ekran debug yaparken incelenmek istenilen parametrelerin girildiği ekrandır. Program durdurulduğu ya da herhangi bir dur noktasına geldiği zaman bu ekrana girilen parametreler geçen zamana kadar elde ettiği bilgileri göstermektedir. Bir sonraki dur noktasına gelene kadar bilgiler buradan gözlemlenebilmektedir. Bir sonraki dur noktasına geldiğinde bilgiler tazelenmektedir.

3. ekran bu projede olduğu gibi uzun yazılımlarda belirlenmiş parametreleri yazılım içinde aratmaya yaramaktadır.

4.1.1 ICD

PCW C Compiler programı icd (In-Circuit Debugger) devresi ile birlikte kullanılmaktadır. Bu devre üzerinde PIC 16F876 mikrokontrolörünü barındırmaktadır. Programdan gelen komutlar doğrultusunda çalışmakta ve DEBUG işlemini destekleyen mikrokontrolörlerde programı adım adım çalıştırmaya yaramaktadır. Proje için bu devrenin şeması internet üzerinden bulunmuş ve gerçeklenmiştir. Devre robota kendi üzerinde bulunan ICD/ICSP konnektöründen bağlanmaktadır. (Şekil 4.2)



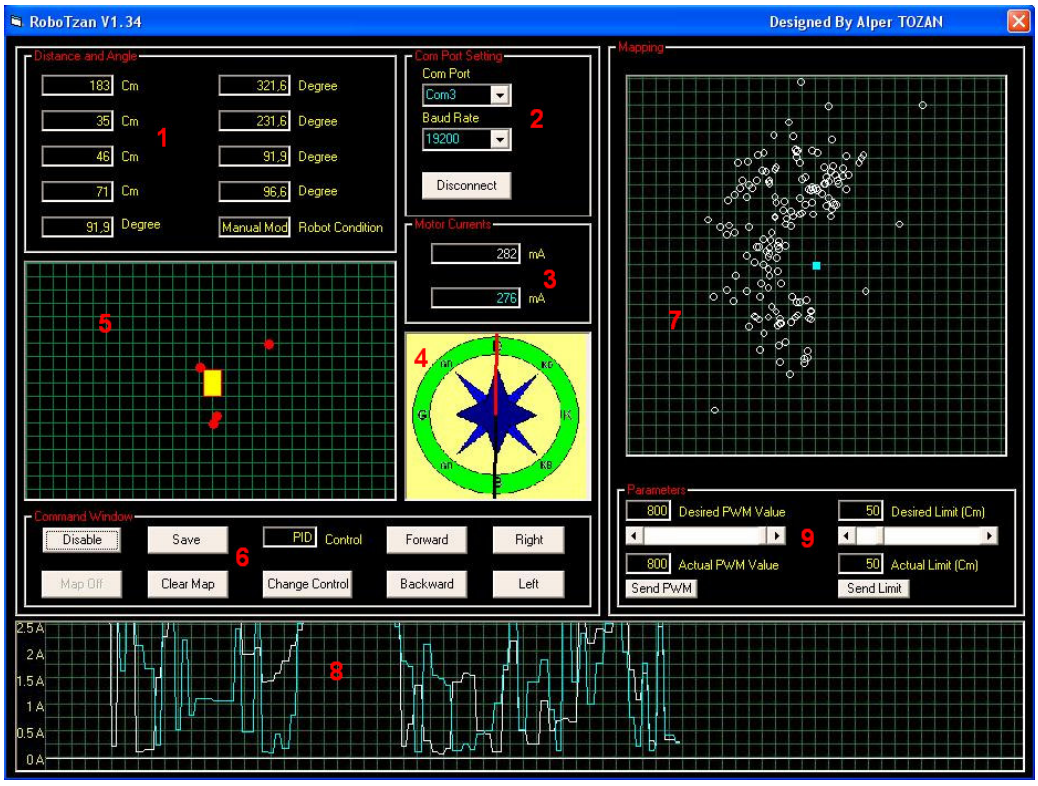
Şekil 4.2 ICD Soketi

Bu konektör vasıtası ile hem DEBUG işlemi hem de programlama yapılmaktadır.

4.2 Bilgisayar Arayüzü

Mobil robot ile bilgisayar arasındaki bilgi alışverişinin gözlemlenmesi için MicroSoft Visual Basic programı ile bir program oluşturulmuştur. Programın içyapısında gelen

bilgilerin doğruluğu CheckSum algoritması ile kontrol edilmektedir. Programın arayüzü şekil 4.3 de görülmektedir.



Şekil 4.3 Bilgisayar Arayüzü

1 nolu kısım algılanan cisimlerin robota göre açısal değerini, mesafesini, kuzey bilgisini ve robotun yaptığı işlemi belirtmektedir. Mesafe ölçümlerinin birimi Cm, açısal değerler derece olarak ölçülmektedir.

2 nolu kısım robotla bilgisayar arayüzünün arasında gerçekleşecek olan bilgi akışı için gerekli ayarlamaların yapılmasına olanak sağlayan ekrandır. Com Port kutusu bağlanmak istediğiniz Com Port' unu seçmekte, Baud Rate kutusu bağlanmak istediğiniz hızı belirlemektedir. Bu kutulardan gerekli ayarlamalar yapıldıktan sonra Connect butonuna basıldığı taktirde bağlantı gerçekleşmektedir. Bilgi akışının güvenliği açısından bağlantı kesilmeden parametrelerde ayarlama yapılması kısıtlanmıştır. Eğer yeniden yapılandırma işlemi yapılmak isteniyorsa bağlantıyı kesmek ve ondan sonra ayarları değiştirmek gerekmektedir.

3 nolu kısım motor sürücüsünden gelen akım değerlerini gözlemlemek için kullanılmaktadır. Gözlenen parametreler mobil robotun sağ ve sol tarafındaki motor akımlarıdır. Ölçülen birimler mA olarak belirtilmiştir.

4 nolu kısım kuzey bilgisini analog olarak gözlemlemek için yapılmıştır. İbrenin kırmızı ucu kuzeyi göstermektedir.

5 nolu kısım belirlenen cisimlerin uzaklıklarının ve yönlerinin görsel olarak algılanması için yapılmıştır. Basit bir mapping algoritması içermektedir. Ekranda görünen her bir kare 45 cm yi ifade etmektedir.

6 nolu kısımda görünen Enable butonu RF protokolü üzerinden robot ile iletişime geçtiğimiz zaman ortaya çıkmaktadır. Enable butonuna basıldığı taktirde H 0X05 bilgisi robota iletilmekte ve robot bu bilgiyi aldığı taktirde karşılığında H 0X0B bilgisini bilgisayar üzerindeki yazılıma iletmektedir. Bilgisayar üzerindeki yazılım bilgiyi doğruladığı taktirde robot el ile kumanda moduna geçmektedir. Bu durumda 5 nolu kısımda görünen diğer butonlar aktif hale gelmektedir. Bu ekrandan robota ileri, geri, sağ ve sol komutları verilebileceği gibi "Change Control" butonu ile hız kontrol algoritmasını PID den Fuzzy ye ya da Fuzzy den PID ye değiştirme imkanı sunmaktadır. "Map On" butonu robotu kendi merkezinde 360º döndürüp algılanan değerleri 7 nolu ekrana yerleştirmektedir. "Clear Map" butonu ile 7 nolu ekranı temizleme imkanı bulunmaktadır. "Save" butonu 7 ve 8 nolu ekranlardaki değerleri resim formatında kaydetmeye yaramaktadır.

7 nolu kısım "Map On" butonuna basıldığında robotun üzerinde bulunan mesafe sensörlerinden gelen verilerin yerleştirilip basit bir harita oluşturulduğu ekrandır.

8 nolu kısım motor sürücüden gelen akım değerlerini zaman ekseninde çizdirmek için kullanılmaktadır. Gözlenen parametreler mobil robotun sağ ve sol tarafındaki motor akımlarıdır.

9 nolu kısım robot elle kumandaya geçtiğinde aktif olan bir kısımdır. Robotun hızını ve kritik dönüş noktası buradaki parametrelerle oynayarak ayarlama olanağı sunmaktadır.

5 HABERLEŞME PROTOKOLLERİ

Mobil robot projesinde kullanılan sensörler ve mikrokontrolör tarafından işlenmiş verilerin iletimi ile ilgili belli başlı veri iletim protokolleri uygulanmıştır. Bunlar aşağıdaki gibi sıralanabilir;

1. RS232

2. I2C

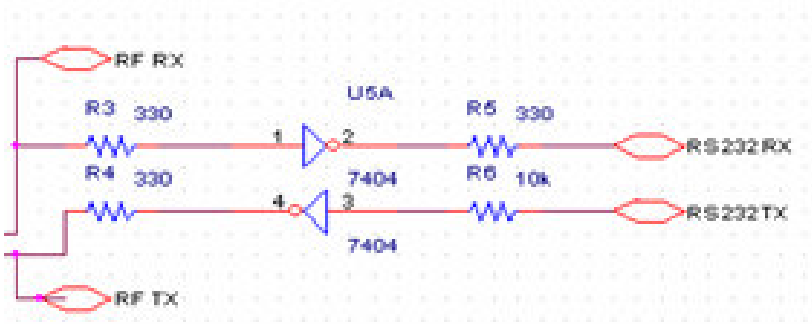
5.1 RS232 Protokolü

RS-232 *EIA* (*Electronic Industries Association*) tarafından geliştirilmiş bir standarttır. +15V ve -15V arasında iki voltaj seviyesi kullanarak 15 metre civarındaki birimler ile haberleşmek için kullanılır. Modem, klavye ya da terminal gibi kısa mesafelerdeki birimlere sayısal veri aktarmak için geliştirilmiştir. Veri karakter olarak taşınır. İletim seri yapılır (bitler ardışıl gönderilir).

İletimin *eşzamansız* (*asynchronous*) olması nedeniyle gönderici ve alıcının koordine olması gerekmez. Gönderen birim belli bir formatta hazırlanan veriyi hatta aktarır. Alıcı ise devamlı olarak hattı dinlemektedir, verinin gelişini bildiren işareti aldıktan sonra gelen veriyi toplar ve karakterleri oluşturur. Her karakterin yedi bitten oluşması gelen verinin işlenmesinde kolaylık sağlar. RS-232' de, eksi voltaj seviyesi 1, artı voltaj seviyesi 0 anlamındadır. Hattın boş olduğu eksi voltaj seviyesi ile ifade edilir. Veri gönderileceği voltajın artı değere çekilmesi (0, başlangıç biti) ile ifade edilir ve ardından yedi bitlik karakter verisi gönderilir. Her bit için voltaj belli bir süre aynı seviyede tutulur. Gönderici ve alıcı birimler bu süreye göre ayarlanmıştır. Her karakterin sonuna bir bitiş biti (1) eklenir.

Tasarlanan merkezi ünitede RS232 için standart olarak kullanılan MAX232 entegresi yerine yapılan araştırma ve denemeler sonucunda 74HC04 logic tersleyici entegresinin kullanılabileceği belirlenmiş ve dış komponentleri azaltmak amaçlı

kullanılmıştır. 74HC04 entegresinin RS232 protokolünde kullanılması için gerektirdiği üç adet 330 Ω luk direnç ve bir adet 10 $k\Omega$ luk direnç şekil 5.1 de belirtildiği gibi devreye yerleştirilmiştir.



Şekil 5.1 : RS-232 Devre Şeması

Yapılan analizlere ve deneylere göre 0 Volt u bilgisayarın Com Port' u logic 1 olarak algılamakta ve +5 Volt u logic 0 olarak algılamaktadır. Yapılan literatür araştırmalarında da maksimum logic sıfır algılama eşiği +3 volt olarak tanımlanmış olarak bulunmuştur.

Yapılan testlerde iki ayrı devre kullanılmıştır. Tasarlanan iki ayrı devrenin bir tanesinde MAX232 kullanılmış diğerinde 74HC04 kullanılmıştır. Elde edilen sonuçlarda 74HC04' ün MAX232 ile eşdeğer sonuç verdiği tespit edilmiş ve tasarıda kullanılmıştır.

5.2 I2C Protokolü

I2C veri yolu 1980lerin basında Philips Semiconductors tarafından geliştirilmiştir. İlk amacı bir televizyon cihazındaki merkezi işlemci ile harici entegrelerini birbirine kolayca bağlamak için geliştirilmiştir.

Televizyon, video, müzik seti gibi çok sayıda entegre devre kullanılan malzemelerde entegreler arasında çok sayıda bakır yol bulunması PCB maliyetlerini arttırdığı gibi Electromagnetic Interference (EMI) (manyetik alanda etkilenme) ve Electrostatic Discharge (ESD) (elektrostatik ani deşarj) problemlerini de meydana getirmektedir. Bu problemleri gidermek için Philips Labs in Eindhoven (Hollanda) iki hatlı I2C veri yolu geliştirmiştir. De-Facto haline gelen bu veri yolu günümüzde Xicor, ST

Microelectronics, Infineon Technologies, Intel, Texas Instruments, Maxim, Atmel, Analog Devices ve diğer büyük elektronik şirketleri tarafından desteklenmektedir.

1982'deki ilk I2C yayımından sonra çok popüler olması ve birçok alanda kullanılması I2C'nin geliştirilmesi ihtiyacını doğurmuştur. Bu ihtiyaç doğrultusunda 1992'de yeni bir I2C standardı yayınlanmıştır. Bunda hızlı modu (FAST MODE) ve 10 bit adreslemeyi içeren yeni gelişmeler mevcuttur.

Bu çalışmayla veri yolu hızı 400 Kbit/s'a yükseltildi ve sistem gürültüsüne limit konuldu. Girişlere schimitt tetikleyicisi yerleştirildi, çıkışlara eğim kontrol edici devre elemanları eklendi.

Her kullanılan entegrenin kendine özel bir adresi vardır. I2C destekleyen entegre sayısı arttıkça ilk 7 bitlik adres yetmemeye başlamış ve 10 bitlik yeni bir adresleme tanımlanmıştır. Önceden rezerve ayrılan bir adres grubu işaretleyici olarak kullanılmıştır. Buna genişletilmiş adresleme adı verilmiştir. Bu yeni adresleme önceki standarda tamamen geriye dönük uyumlu çalışır olarak tasarlanmıştır.

Gelişen uygulamalarla birlikte 400 kbit/s hızda yetmemeye başlayınca 3.4 Mbit /s lık son derece hızlı yeni bir mod geliştirilmiş ve buna high speed mod adı verilmiştir.

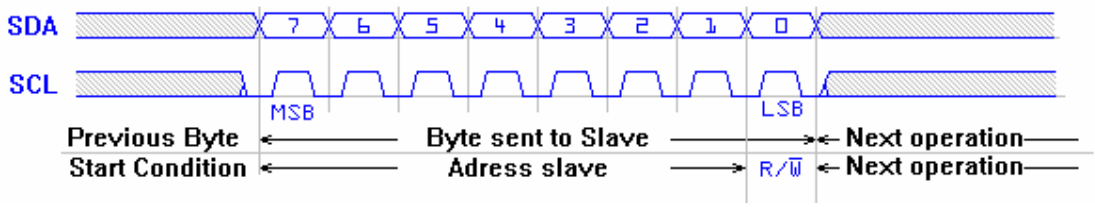
Bu yüksek hızına rağmen yeni cihazlar geriye dönük olarak fast mode yada standart I2C ile iletişim kurabilme yeteneğine sahip olarak tasarlanmıştır.

I2C veri yolu iki aktif iletken ve bir toprak bağlantısından oluşur. SCL ve SDA denen aktif iletken teller iki yönlüdür. SDA: Serial Data Line yani seri data hattı, SCL: Serial Clock Line yani seri saat hattıdır. Mikrokontrolöre bağlı her bir entegrenin kendine özel bir adresi vardır. Kullanım amacına bağlı olarak bu entegrelerin her birisi veri yolu üzerinde alıcı yada gönderici olabilir.

I2C veri yolu ÇOK MASTERLI bir veri yoludur. Bu birden fazla kendiliğinden veri transferi başlatabilen entegre veri yoluna bağlanabileceği anlamına gelmektedir. I2C protokolüne göre yürütülmekte olan veri transferini başlatan entegreye BUS MASTER diğer tüm entegrelere BUS SLAVE denir. BUS MASTERLAR genellikle mikrokontrolör veya denetleyicilerdir.

I2C' nin veri iletim hızı gönderilen bilginin frekansı ile ters orantılıdır. Örnek vermek gerekirse 1 Mhz gibi yüksek hızlarda veri aktarım mesafesi herhangi bir buffer kullanılmadığı taktirde 8 ila 10 metre iken 500 Hz lik bir veri iletiminde bu mesafe 100 m civarına çıkmaktadır.

Mobil robot projesinde I2C protokolü dijital pusuladan kuzey bilgisini almak için kullanılmaktadır. Veriyi alabilmek için mikrokontrolör başla komutunu yollar ve arkasından dijital pusulanın bulunduğu adres olan 0xC0 hex bilgisini yollar. Burada dikkat edilmesi gereken nokta adres bilgisini gönderirken son bit yazma ya da okuma işlemini belirler. 0 yazma işlemini belirtirken 1 okuma işlemini belirtmektedir.



Şekil 5.2 : I2C Bilgi Akışı

Adres bilgisini yolladıktan sonra okumak istediği pusula verisinin register adresi olan 0x02 hex bilgisini I2C üzerinden pusulaya gönderir ve I2C protokolünü tekrardan başlatır. Tekrardan dijital pusulanın adresini SDA üzerinden yollar. Veri okumak istediğini adresin sonuna eklediği 1 bilgisinden anlaşılmaktadır.(0xC1) Okumak istediğimiz verinin boyutu iki byte olduğundan 2 okuma komutu ile veri alınır. Veri alışverişi durdurma komutu ile bitirilir. Veri alımının akış diyagramı ve proje için yazılan kodun bir örneği aşağıdaki şekil 5.4 de verilmiştir.

Şekil 5.3: I2C Akış Diyagramı

```
i2c_start();

i2c_write(0xC0);      //Device address (Write)

i2c_write(0x02);      //Data to device

i2c_start();             //Restart

i2c_write(0xC1);     // Device address (Read)

datah=i2c_read();   //read from slave

datal=i2c_read(0);  //read from slave

i2c_stop();             //stop i2c
```

Şekil 5.4: Mikrokontrolör Üzerindeki I2C Yazılımı

6 KULLANILAN KONTROL ALGORİTMALARI

Mobil robot projesinde iki temel kontrol algoritması kullanılmıştır. Bunlardan biri PID diğeri karar verme algoritmasıdır.

6.1 PID Algoritması

PID algoritması robotta hızı kontrol etmek amaçlı kullanılmıştır. 50 ile 100 cm arasında obje algıladığı taktirde devreye girmektedir. Eğer 100 cm mesafede önünde bir obje görmediği taktirde motorların hızını ayarlanmış en üst düzeye çıkarır. Tanımlanmış mesafeler içerisinde obje algılamış ise objeyi algıladığı taraftaki motorları PID den gelen sonuca göre daha hızlı diğer taraftaki motorları daha yavaş çalıştırarak aracı durdurmadan yumuşak dönüşler yapması hedeflenmiştir.

Akım kontrolü için de PID yazılmış fakat sisteme entegre edilmemiştir. İleri dönük çalışmalarda eklenmesi göz önüne alınacaktır.

PID için genel formül aşağıda verilmiştir.

$$U = K_p.e + K_i.\int e + K_d.\frac{d}{dt}e \qquad \textbf{(6.1)}$$

Formülde görünen e ayarlanmış değer ile okunan değer arasındaki farkı belirtmektedir. K_p PID' nin orantısal sabitini, K_i integratör sabitini, K_d ise türev sabitini belirtmektedir. PID' nin çıkışı U yukarıdaki formülden elde edilir.

PID kontrolün P, PI ve PD kontrole göre avantajlarını şöyle sıralayabiliriz.

1 Oransal kontrolde oluşan offset integral yardımıyla ortadan kaldırılır.

2 İstenen değerin çok üstüne yada altına çıkması (overshoot ve undershoot) türev algoritması ile önlenir.

3 Katsayıların düzgün ayarlanması ile mükemmel bir kontrol gerçekleştirilebilir.

6.2 Yazılımda PID Algoritması

PID algoritmasını bir mikrokontrolörde gerçekleştirmek istediğimizde yukarıdaki formülde bazı değişiklik ve eklemeler yapmak gerekmektedir. Elde edilen formül aşağıda yer almaktadır.

$$U = K_p.e_2 + K_i.(Int + e_2) + K_d.(e_2 - e_1) \quad \textbf{(6.2)}$$

Genel formülden farklı olarak burada e_1 ve e_2 diye iki farklı hata mevcuttur. e_2 şuan ki hatayı belirtirken e_1 bir önceki örnekleme zamanında elde edilen hatayı belirtmektedir. Şimdiki hatadan bir önceki hatayı çıkarttığımız takdirde bu bize genel PID formülündeki türev etkisini oluşturur.

$$\frac{d}{dt}e = (e_2 - e_1) \quad \textbf{(6.3)}$$

(6.2) deki formülde gözüken Int, mikrokontrolör içinde tanımlanmış 32 bitlik bir tamsayıdır. Gelen hata sürekli bu tamsayıya toplanarak integral etkisi oluşturulmuştur. Programın bir noktada kilitlenip Int tamsayısının dolup sisteme olumsuz etki yapmaması için belirli bir noktaya limitlenmiştir.

Yazılımdaki PID' nin oransal kısmı ise tamamen genel formüle uyduğundan dolayı bir değişiklik geçirmemiştir. Elde edilen hataların PID katsayıları ile çarpılmasından PID' nin çıkışı olan U elde edilmektedir.

Mikrokontrolörün içindeki PID yazılımının üç girişi bir çıkışı mevcuttur. Bu üç giriş robotun önünde bulunan birbirlerinden 45 derece farklı yöne bakan üç adet ultrasonik sensörden gelen mesafe verileridir. Hata olarak hangi sensörden gelen verinin kabul edeceği, verilerin birbiri ile karşılaştırılması ile belirlenir. Üç sensörün verisi de aynı önemi taşıdığından karşılaştırma işlemi kolaylaşmıştır.

Gelen üç verinin hepsi tanımlanmış maksimum ve minimum limit değerleri ile karşılaştırılır. Mobil robot tanımlanmış mesafe değerine 0 dan yaklaşırken PID üç sensörden elde edilen minimum hatayı alırken, tanımlanmış değeri geçtiğinde üç

sensörden elde edilen maksimum hatayı göz önüne alıp PID algoritmasına yerleştirip U çıkışını elde eder.

Elde edilen U, bir interpolasyondan geçerek PWM değeri elde edilir ve mobil robotun sağ ve sol motorlarının tanımlı PWM değerlerine artı ya da eksi olarak etki eder.

6.3 Karar Algoritması

Karar algoritması PID algoritmasının sağladığı yumuşak dönüşlerin yeterli gelmediği ve mobil robotun giderek cisme tehlikeli bir şekilde yaklaştığı zaman devreye giren bir algoritmadır. Öndeki üç arkadaki bir sensörden aldığı veriler doğrultusunda karar verir.

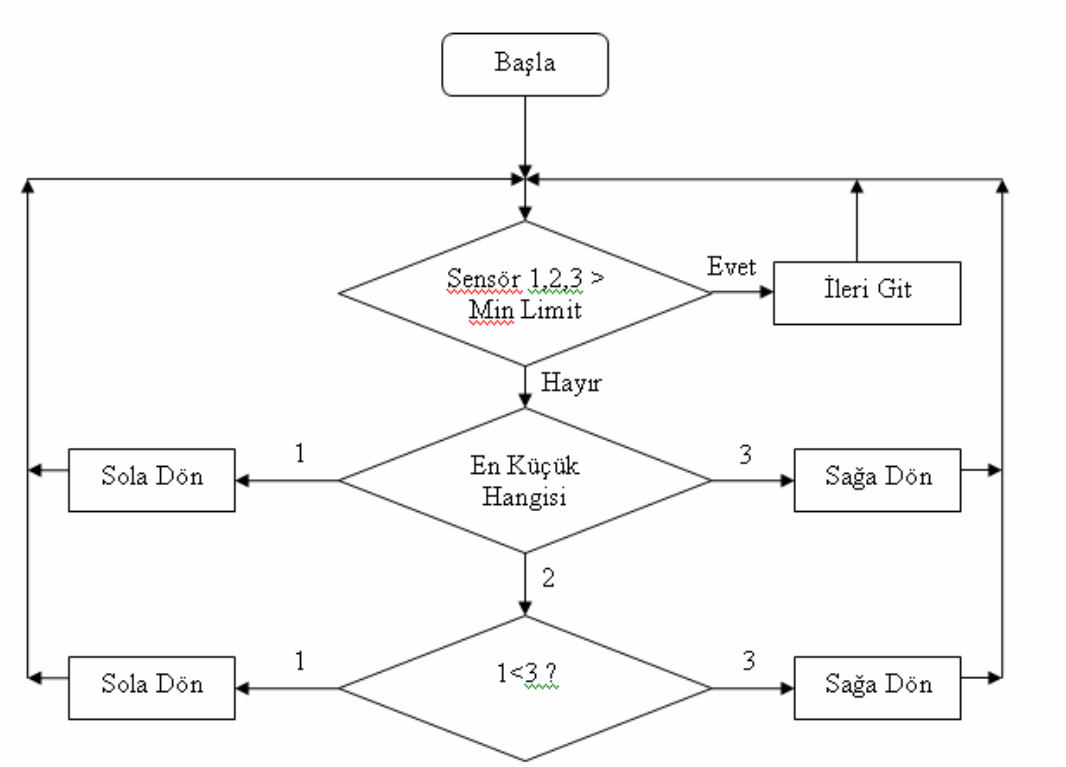
Eğer öndeki üç mesafe sensöründen gelen verilerin hiç biri limitin altında değil ise mobil araç ileri yönde tanımlanmış maksimum hızla ilerlemektedir. Bu zaman zarfında mobil robot sağda, solda yada ileride bir cisim algıladığında, algıladığı cismin yönü doğrultusunda bulunan tekerleklerdeki hızı arttırarak diğer yöndeki tekerleklerin hızını azaltarak mobil robotu kontrol etmektedir.

Fakat algılanan mesafe, tehlikeli limit noktasına girdiği zaman karar algoritması devreye girer ve motorun yönünü değiştirip tekerleklerdeki hız değerlerini ayarlar.

Bazı durumlarda sağ ve soldaki cisimleri algılamak için yerleştirilen sensörlerin limitler içinde bir obje algılamadığı, fakat öndeki sensörün cisim algıladığı durumlarda karar algoritması sağ ve sol sensörlerden elde ettiği verileri inceleyip en uzakta cisim algılanan yöne dönmektedir. Karar algoritması ile ilgili basit bir akış diyagramı şekil 6.1 görülmektedir.

6.4 Fuzzy Temelli Hız Kontrol Algoritması

PID ile yapılmış olan hız kontrol algoritmasında kullanılan karar algoritması fuzzy temelli bir algoritmadır. Kendi içinde kural tablosu bulunmaktadır (Tablo 6.2). Temel olarak bu karar tablosu üzerinden kararlar vermekte ve çevresindeki cisimlerden kaçınarak yolunu bulmaktadır.



Şekil 6-1: Karar Algoritması Akış Diyagramı

Hız kontrolü için robotun önünde bulunan 3 sensörden gelen bilgilerden yararlanılmıştır. Gelen her bir bilgi şekil 6.2 de görünen üyelik fonksiyonuna sokulmuştur. Elde edilen değer PWM skalasına göre oranlanmıştır. Kural tablosundan çıkan sonuca göre sağ ve sol motor PWM değerleri oluşturulmuştur. Temel olarak bir motorun alacağı PWM değeri şekil 6.3 de görülmektedir.

Fuzzy temelli hız ve karar algoritmasının kural tablosu tablo 6.1 de görülmektedir. Bu kural tablosunun mikrokontrolör üzerinde oluşturulmuş şekli tablo 6.2 deki gibidir.

Algoritma tam olarak fuzzy algoritmasının gereksinimlerini karşılamamasının sebebi robotun bilgisayardan bağımsız çalışması ve üzerinde bulunan mikrokontrolörün işlem gücünün bir bilgisayar kadar olmamasıdır. Dolayısı ile işlem zamanını azaltmak ve robotun alacağı kararları zamanında alabilmesi için geleneksel fuzzy algoritması kısaltılmış ve fuzzy mantığına sahip bir algoritma geliştirilmiştir.

Tablo 6.1 Kural Tablosu

x_1	x_2	x_3	$(x_1 > x_2)$	$(x_1 < x_2)$	Sonuç
Büyük	Büyük	Büyük	Fark etmez	Fark etmez	Hızlı İleri
Büyük	Büyük	Orta	Fark etmez	Fark etmez	İleri Sol
Büyük	Büyük	Küçük	Fark etmez	Fark etmez	Sol
Büyük	Orta	Büyük	Evet	Hayır	İleri Sol
Büyük	Orta	Büyük	Hayır	Evet	İleri Sağ
Büyük	Orta	Orta	Fark etmez	Fark etmez	İleri Sol
Büyük	Orta	Küçük	Fark etmez	Fark etmez	Sol
Büyük	Küçük	Büyük	Evet	Hayır	Sol
Büyük	Küçük	Büyük	Hayır	Evet	Sol
Büyük	Küçük	Orta	Fark etmez	Fark etmez	Sol
Büyük	Küçük	Küçük	Fark etmez	Fark etmez	Sol
Orta	Büyük	Büyük	Fark etmez	Fark etmez	İleri Sağ
Orta	Büyük	Orta	Evet	Hayır	İleri Sol
Orta	Büyük	Orta	Hayır	Evet	İleri Sağ
Orta	Büyük	Küçük	Fark etmez	Fark etmez	Sol
Orta	Orta	Büyük	Fark etmez	Fark etmez	İleri Sağ
Orta	Orta	Orta	Evet	Hayır	İleri Sol
Orta	Orta	Orta	Hayır	Evet	İleri Sağ
Orta	Orta	Küçük	Fark etmez	Fark etmez	Sol
Orta	Küçük	Büyük	Fark etmez	Fark etmez	Sağ
Orta	Küçük	Orta	Evet	Hayır	Sol
Orta	Küçük	Orta	Hayır	Evet	Sağ
Orta	Küçük	Küçük	Fark etmez	Fark etmez	Sol
Küçük	Büyük	Büyük	Fark etmez	Fark etmez	Sağ
Küçük	Büyük	Orta	Fark etmez	Fark etmez	Sağ
Küçük	Büyük	Küçük	Evet	Hayır	Sol
Küçük	Büyük	Küçük	Hayır	Evet	Sağ
Küçük	Orta	Büyük	Fark etmez	Fark etmez	Sağ
Küçük	Orta	Orta	Fark etmez	Fark etmez	Sağ
Küçük	Orta	Küçük	Evet	Hayır	Sol
Küçük	Orta	Küçük	Hayır	Evet	Sağ
Küçük	Küçük	Büyük	Fark etmez	Fark etmez	Sağ
Küçük	Küçük	Orta	Fark etmez	Fark etmez	Sağ
Küçük	Küçük	Küçük	Evet	Hayır	Sol
Küçük	Küçük	Küçük	Hayır	Evet	Sağ

Küçük = Mesafe Ölçümü < 50 Cm

Orta = 50 Cm < Mesafe Ölçümü < 100 Cm

Büyük = Mesafe Ölçümü > 100 Cm

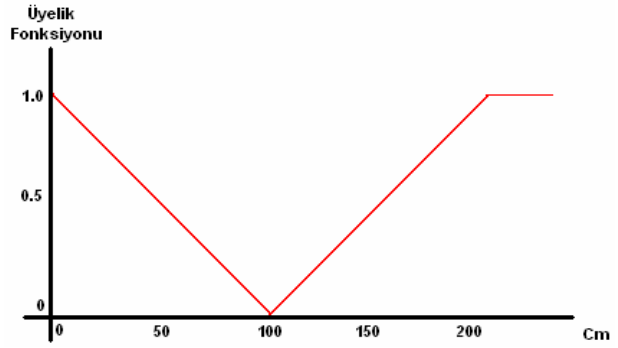
Tablo 6.2 Mikrokontrolör Kural Tablosu

Mesafe Bilgileri	Karar
(x_1 >100)&(x_2 >100)&(x_3 >100)	Hızlı İleri
(x_1 >100)& (x_2 >50)&(100> x_3 >50)	İleri Sol
(100> x_1 >50)&(x_2 >50)&(x_3 >100)	İleri Sağ
(100> x_1 >50)&(x_2 >50)&(100> x_3 >50) (x_1 > x_2)	İleri Sol
(100> x_1 >50)&(x_2 >50)&(100> x_3 >50) (x_1 < x_2)	İleri Sağ
(x_1 >50)&(x_2 >50)&(x_3 <50)	Tam Sol
(x_1 <50)&(x_2 >50)&(x_3 >50)	Tam Sağ
(x_1 <50)&(x_3 <50)& (x_1 > x_3)	Tam Sol
(x_1 <50)&(x_3 <50)& (x_1 < x_3)	Tam Sağ
(x_2 <50)&(x_1 > x_3)	Tam Sol
(x_2 <50)&(x_1 < x_3)	Tam Sağ

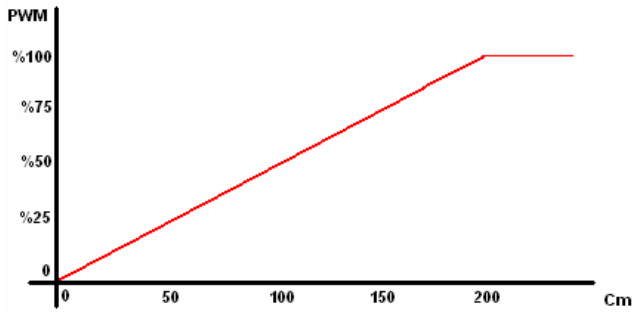
x_1 =Sol mesafe sensörü Ölçümü (Cm)

x_2 =Orta mesafe sensörü Ölçümü (Cm)

x_3 =Sağ mesafe sensörü Ölçümü (Cm)



Şekil 6.2 Üyelik Fonksiyonu



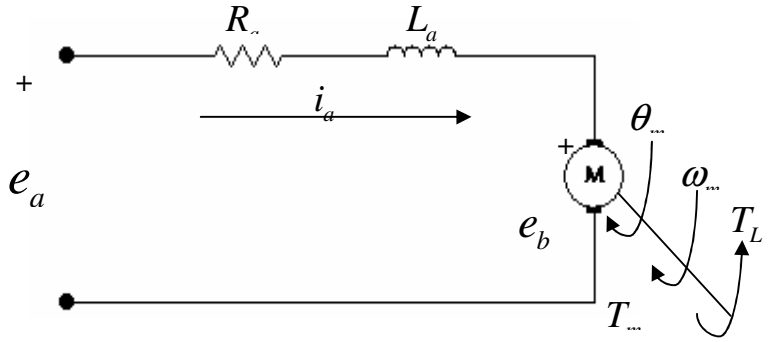
Şekil 6.3 PWM çıkışı

7 SİSTEMİN MODELLENMESİ

Mobil robot projesinde kullanılan motorların ve sistemin matematik formüleri çıkartılmış ve simülasyonları yapılmıştır. Çıkan sonuçlar sistemden alınan veriler ile karşılaştırılıp yorumlanmıştır.

7.1 DC Motor Modeli

Mobil robot projesinde robota hareket kazandıracak unsur olan DC motorlar olduğundan kullanılan DC motorları modellemek sistem modelinde önemli yer tutmaktadır.



Şekil 7.1: DC Motor Modeli

DC motorun eş değer devresi şekilde görülmektedir. Voltaj kaynağına seri bir R_a, ona seri bağlanmış bir L_a bulunmaktadır. e_b motorun dönüşü ile armatür de oluşan EMF voltajıdır.

Motor uygulanan $e_a(t)$ voltajı ile kontrol edilir. Lineer analiz yaklaşımı ile motorun oluşturduğu tork, mıknatıs ile stator arasındaki hava boşluğunda oluşan flux ve armatürden geçen akım ile doğru orantılıdır. Dolayısı ile tork formülünü;

$$T_m = K_m(t)\phi i_a(t) \qquad \textbf{(7.1)}$$

şeklinde yazabiliriz. ϕ sabit olduğundan dolayı formül;

$$T_m = K_i i_a(t) \tag{7.2}$$

şekline dönüşür. K_i N-m/A biriminden tork sabitidir. Besleme voltajı $e_a(t)$ yi de formüle eklediğimiz taktirde DC motor modeli;

$$\begin{aligned}
&\frac{di_a(t)}{dt} = \frac{1}{L_a}e_a(t) - \frac{R_a}{L_a}i_a(t) - \frac{1}{L_a}e_b(t) \\
&T_m = K_i i_a(t) \\
&e_b(t) = K_b \frac{d\theta_m(t)}{dt} = K_b \omega_m(t) \\
&\frac{d^2\theta_m(t)}{dt^2} = \frac{1}{J_m}T_m(t) - \frac{1}{J_m}T_L(t) - \frac{B_m}{J_m}\frac{d\theta_m(t)}{dt}
\end{aligned} \tag{7.3}$$

Modelde $T_L(t)$ yükün sürtünme torku dur. Besleme voltajı $e_a(t)$ uygulandığında akım $i_a(t)$ oluşmaya başlar. Akım $i_a(t)$ torku $T_m(t)$ oluşturur. Torkta $T_m(t)$ açısal hızı $\omega_m(t)$ ve kat edilen açısal mesafeyi $\theta_m(t)$ oluşturur.

Modelin denklemlerini matris formunda yazacak olursak;

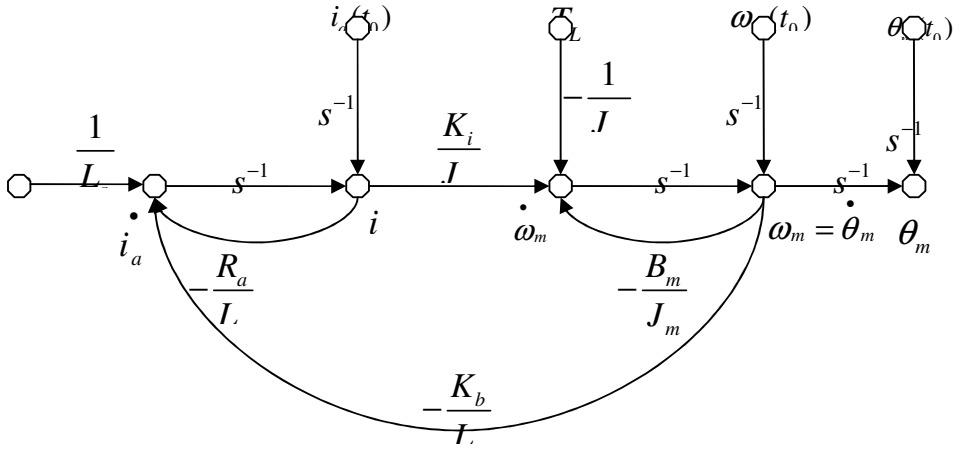
$$\begin{bmatrix} \frac{di_a(t)}{dt} \\ \frac{d\omega_m(t)}{dt} \\ \frac{d\theta_m(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} & 0 \\ \frac{K_i}{J_m} & -\frac{B_m}{J_m} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} e_a(t) - \begin{bmatrix} 0 \\ \frac{1}{J_m} \\ 0 \end{bmatrix} T_L(t) \tag{7.4}$$

şekline gelmektedir.

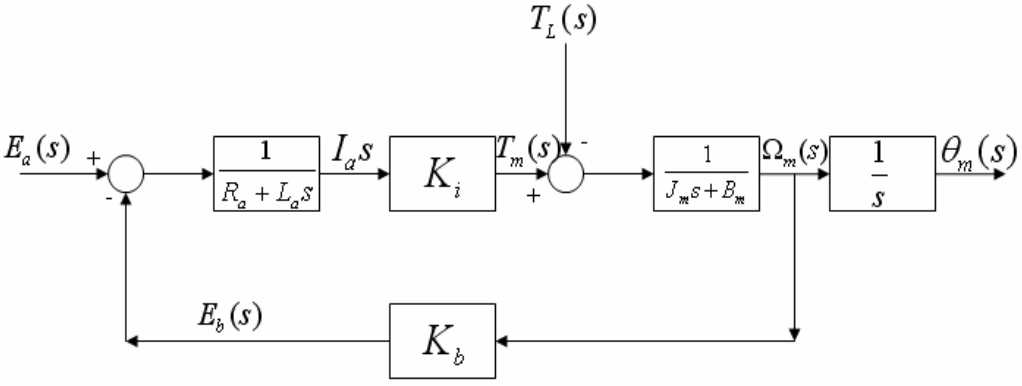
Buradan transfer fonksiyonuna geçersek eğer;

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{K_i}{L_a J_m s^3 + (R_a J_m + B_m L_a)s^2 + (K_b K_i + R_a B_m)s} \tag{7.5}$$

formül 7.5' i elde etmiş oluruz.



Şekil 7.2: DC Motor Durum Diyagramı



Şekil 7.3: DC Motor Blok Diyagramı

7.2 Mobil Robot Modeli

Yapılan literatür araştırmaları sonucunda gerçekleşmiş olan mobil robot paletli araç modeline uymaktadır. Bu sistemin modeli oluşturulurken ve ileride yapılacak olan simülasyonlarda paletli araç modeli kullanılmıştır

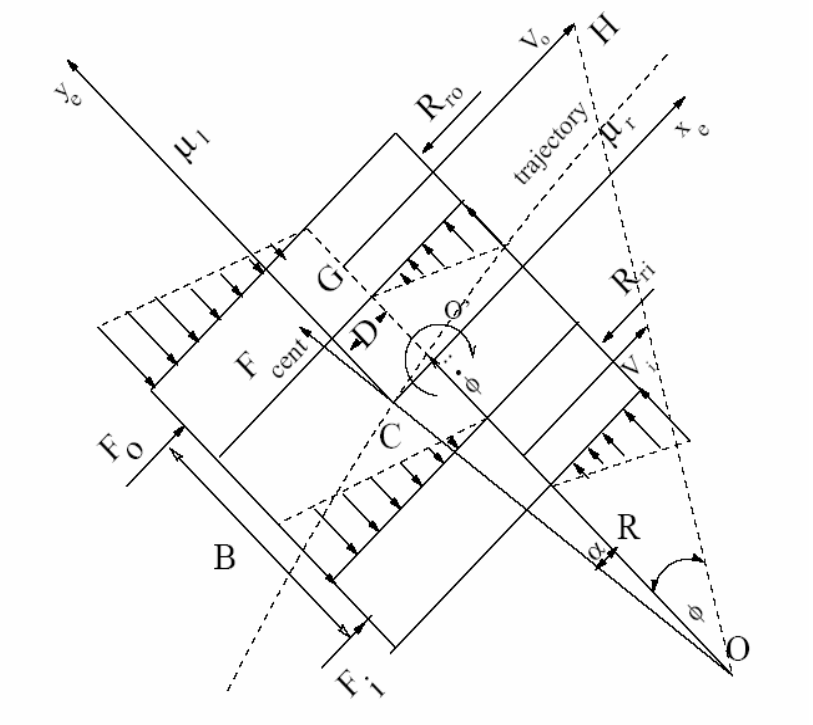
7.2.1 Kuvvet Modeli

Paletli araçlarda normal araçlarda olduğu gibi ön tekerleklerin açısını değiştirerek yön değiştirilmemektedir. Mobil robotumuzda olduğu gibi paletli araçların büyük çoğunluğunda yön kontrolü kayma yönlendirmesi (skid steering) yardımı ile yapılmaktadır. Bu yöntemde araç paletlere uygulanan farklı itme kuvvetleri yardımı

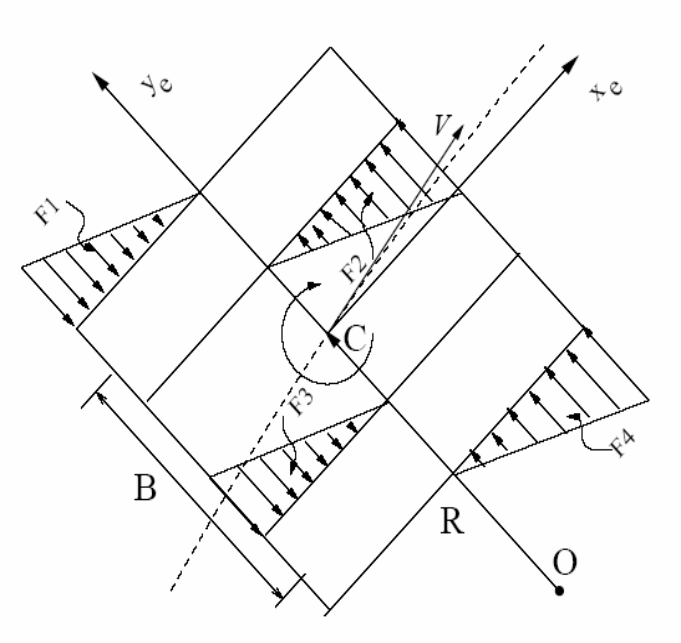
ile döndürülür. İtme kuvvetlerinin farklı olmaları bir dönme momenti oluşturur. Bu dönme momenti paletlerin yüzeyde kaymalarından oluşan yanal dönme dirençlerini yenecek noktaya ulaştığında dönme işlemi gerçekleşmiş olur.

Doğada kuvvetler dinamik yada statik olarak bulunurlar. Statik kuvvetler araç düz bir doğrultuda sabit hızla ilerlerken etki eder. Aracın paletleri ile yol arasındaki sürtünme statik bir kuvvettir. Yapılan hızlanmalar yada yavaşlamalar ise araca dinamik kuvvetlerin etkimesine yol açar. Aracın bir yöne dönmesi merkez kaç kuvvetinin aracın kütle merkezine etkimesine ve aracı yana doğru savurmaya yol açar. Dinamik kuvvetlerin hepsi aracın kütle merkezine etkiler. şekil 7.4 de paletli bir aracın orta ve yukarı hızlarda etkiyen kuvvetler görülmektedir. F_o ve F_i paletlere etkiyen itme kuvvetleri, R_{ro} ve R_{ri} yüzeyin paletlere etkilediği boylamsal direnç, μ_r ve μ_l boylamsal ve yatay direnç katsayıları, B iki arka tekerlek arasındaki mesafe, F_{cent} atalet (inertial) kuvveti, α da kayma açısıdır. Şekil 7.4 te araç anlık O^t merkez noktasında R yarıçapı ile sağa dönmektedir. Dönüş sırasında paletin O^t anlık merkez noktasına yakın yerlerine daha az kuvvet etkilerken uzak noktalarına daha çok kuvvet etkilemektedir. Zeminin yapısını homojen olarak kabul edersek palete etkiyen yanal kuvvetler üçgen yapıya sahip bir dağılım gösterirler. Yanal direnç, aracın dönme yönüne ters yönde bir M_r dönüş direnç momenti oluşturur.

Şekil 7.4 te orta ve yukarı hızlarda O^t anlık merkez noktasından D kadarlık kaymaya yol açar bu da dönüş esnasında α kayma açısının oluşmasına sebebiyet verir. Düz bir hatta sabit bir hızla ilerlerken α kayma açısı 0, sağa yada sola dönüşlerde ise pozitif yada negatif açı değerleri almaktadır. Dönüş sırasında aracın merkez noktasına merkez kaç kuvveti etkiler. Bu etkiyen merkez kaç kuvveti nokta dönüşü yapıldığında yada çok düşük hızlarda dönüş yapıldığında sıfıra yakın değerler almaktadır. Merkez kaç kuvveti ihmal edilirse yanal kuvvetlerin palete olan etkisi homojen olarak şekil 7.5 de görüldüğü gibi etkimektedir. Dolayısı ile F1 ile F4, F2 ile F3 bir birine eşdeğer olacaktır. Dönüş hızı artırıldığında merkez kaç kuvvetinin paletlere olan ek etkisinden dolayı bu eşitlik bozulmakta ve şekil 7.6 da ki hale gelmektedir.



Şekil 7.4: Paletli Araç Modeli



Şekil 7.5 Düşük Hızlarda Paletli Araç Modeli

Eğer araca etkiyen merkez kaç kuvveti göz önüne alınıp aracın paletlerine etkiyen kuvvetler F1, F2, F3 ve F4 hesaplanmak istendiğinde;

$$F_I = \frac{mg\mu_l}{l} - \frac{m\dot{\phi}^2 R}{2l}$$

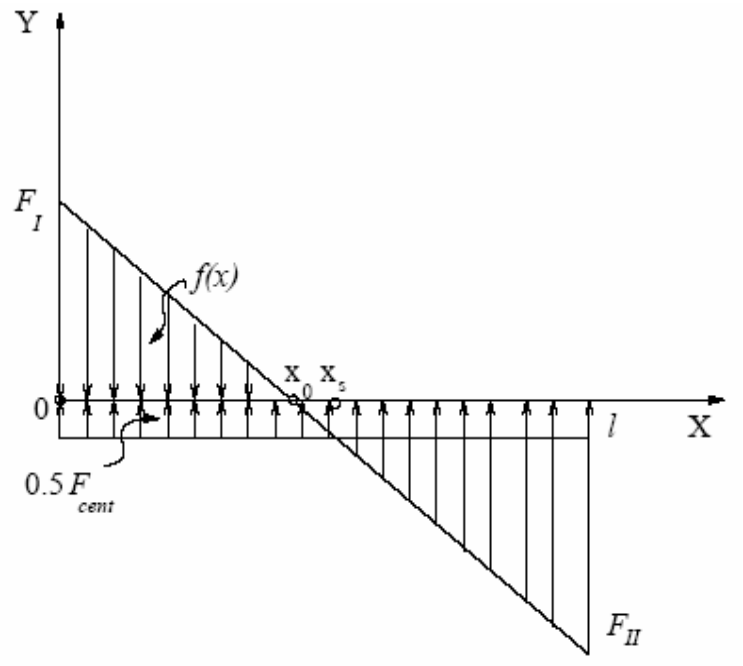
$$F_{II} = -\frac{mg\mu_l}{l} - \frac{m\dot{\phi}^2 R}{2l} \qquad \textbf{(7.6)}$$

$$f(x) = -\frac{2mg\mu_l}{l^2} + \frac{mg\mu_l}{l} - \frac{m\dot{\phi}^2 R}{2l}$$

$$x_0 = \frac{l}{2} - \frac{\phi^2 Rl}{4g\mu_l}$$

F_I ve F_{II} paletin başlangıç ve bitiş noktasında palete etkiyen kuvvetler, $f(x)$ paletin boyu l üzerinde herhangi bir noktaya etkiyen kuvveti hesaplamak için kullanılan formül ve x_0 etkiyen merkez kaç kuvvetinden dolayı yeri değişen 0 yanal kuvvet noktasının palet üzerindeki yeni yerini belli etmektedir.

Bu formüllerden yola çıkılarak aşağıdaki denklemler yazılabilir.



Şekil 7.6 Palete Etkiyen Kuvvetler

$$M_y = \int_0^l xf(x)dx$$

$$A = \int_0^l f(x)dx \qquad \textbf{(7.7)}$$

$$x_s = \frac{M_y}{A}$$

Bu denklemleri iki duruma göre açabiliriz.

1. Durum $F_l \geq 0$

$$M_y = \int_0^l xf(x)dx$$

$$M_y = \int_0^{x_0} xf(x)dx - \int_{x_0}^l xf(x)dx$$

$$M_y = \frac{2mg\mu_l}{3l^2}(l^3 - 2x_0^3) + \frac{mg\mu_l}{2l}(2x_0^2 - l^2) + \frac{m\dot{\phi^2}R}{4l}(l^2 - 2x_0^2)$$

$$M_y = \frac{mg\mu_l l}{4} + \frac{m\dot{\phi^2}Rl}{8} + \frac{m\dot{\phi^4}Rl}{16g\mu_l} - \frac{m\dot{\phi^6}R^3 l}{96g^2\mu_l}$$

$$A = \int_0^l f(x)dx$$

$$A = \int_0^{x_0} f(x)dx - A = \int_{x_0}^l f(x)dx$$

$$A = \frac{mg\mu_l}{2} + \frac{m\dot{\phi^4}R^2}{8g\mu_l}$$

$$x_s = \frac{l}{2} + \frac{\dot{\phi^2}Rl}{4g\mu_l} - \frac{\dot{\phi^6}R^3 l}{24g^2\mu_l^2(\frac{g\mu_l}{2} + \frac{\dot{\phi^4}R^2}{8g\mu_l})} \qquad \textbf{(7.8)}$$

$$x_s \approx \frac{l}{2} + \frac{\dot{\phi^2}Rl}{4g\mu_l}$$

2. Durum $F_l \le 0$

$$
\begin{aligned}
M_y &= -\int_0^l xf(x)dx \\
M_y &= \frac{mg\mu_l l}{6} + \frac{m\dot{\phi}^2 Rl}{4} \\
A &= \frac{m\dot{\phi}^2 Rl}{2} \\
x_s &= \frac{l}{2} + \frac{g\mu_l l}{3\dot{\phi}^2 R}
\end{aligned}
\qquad \textbf{(7.9)}
$$

2. durum da merkez kaç kuvveti o kadar büyük $f(x)$ pozitif bir bileşen getiremez ve gerekli yatay kuvveti oluşturamadığından dolayı araç kontrolden çıkar. Bu durum ancak çok yüksek hızlarda ve çok düşük yanal dirençli μ_l ortamlarda dönüş yapıldığında gerçekleşir.

Hesaplanmış anlık merkez noktası O' gösteriyor ki α kayma açısı arasın ağırlığından bağımsız, yanal direnç μ_l, paletin boyu l ve dönüş hızı $\dot{\phi}$ ile alakalıdır.

Dönme direncinin momenti M_r yanal kuvvetlerin entegralini alarak hesaplanır. Simetriden dolayı dönme direncinin momenti;

$$
\begin{aligned}
M_r &= 4\int_0^{l-x_s} -xf(x+x_s)dx \\
M_r &= 4\int_0^{l-x_s} x\left(\frac{2mg\mu_l}{l^2}x + \frac{2mg\mu_l}{l^2}x_s - \frac{mg\mu_l}{l} + \frac{m\dot{\phi}^2 R}{2l}\right) \\
M_r &= 4\int_0^{l-x_s} x\left(\frac{2mg\mu_l}{l^2}x + \frac{m\dot{\phi}^2 R}{2}\right)dx \\
M_r &= \frac{mgl\mu_l}{3} - \frac{m\dot{\phi}^4 R^2 l}{4g\mu} + \frac{m\dot{\phi}^6 R^3 l}{12g^2\mu_l^2}
\end{aligned}
\qquad \textbf{(7.10)}
$$

Elde edilen sonuçlar gösteriyor ki M_r dönme direncinin momenti, araca etkiyen yanal kuvvetlerden ve merkez kaç kuvvetinden oluşmakta. Aracın nokta dönüşü yaptığını ya da yavaş hızlarda döndüğünü varsayarsak M_r dönme direncinin momenti;

$$M_r = -4\int_0^{\frac{l}{2}} x f_i(x + x_0)dx$$

$$M_r = -4\int_0^{\frac{l}{2}} x\left[\frac{2mg\mu_l}{l^2}(x+\frac{l}{2}) - \frac{mg\mu_l}{l}\right]dx$$

$$M_r = \frac{8mg\mu_l}{l^2}\int_0^{\frac{l}{2}} x^2 dx \qquad \textbf{(7.11)}$$

$$M_r = \frac{8mg\mu_l}{3l^2}x^3\Big|_0^{\frac{l}{2}}$$

$$M_r = \frac{mgl\mu_l}{3}$$

Ve

$$f_i(x) = \frac{2mg\mu_l}{l^2}x + \frac{mg\mu_l}{l} \qquad \textbf{(7.12)}$$

Yanal kuvvetlerin palet boyunca dağılımıdır.

Diğer bir sorun ise direncin aracın paletlerine olan dağılımıdır. Araç ileri yönde ilerlerken ya da dönüşlerde iki paletin aynı yöne döndüğü durumlarda boylamsal direncin paletlere eşit dağıldığı var sayılır. Fakat dönüş bir paletin durdurulup diğer paletin hareket ettirilerek yapıldığı takdirde hareket eden palet aracı döndürebilmek için sadece boylamsal direnci yenmekle kalmayıp dönüş direncinin momentinde yenmek zorundadır. Aynı durum paletlerin aynı yöne döndüğü durumlarda da geçerlidir. Dönüş işleminin gerçekleşmesi için dışarıda kalan palet boylamsal direnci yenmesi gerektiği gibi iç taraftaki paletin aracın merkezinde ürettiği momenti de yenmesi gerekmektedir. Araç noktasal dönüş yapıyorsa ya da dönüş yarıçapı $R \le \frac{B}{2}$ den küçük ise ancak o zaman iki palette direnci yenmeye katkıda bulunur. Bu durumda dönüş direncinin momenti mesafe ile orantısal olarak paletin ortasından

dışarıya doğru yayılır. Eğer dönüş yarıçapı $R > \frac{B}{2}$ ise o zaman dönmek için gerekli kuvveti sadece dış tarafta kalan palet üretmektedir. Dolayısı ile bu nokta α kayma açısını ve paletlere etki eden kuvveti hesaplarken önem kazanmaktadır.

Boylamsal dirençler R_o ve R_i;

$$R_{o,i} = \frac{W \mu_r}{2} \tag{7.13}$$

formülünden hesaplanır. Formülde ki μ_r boylamsal direnç sabitidir.

7.2.2 Sistemin Hareket Denklemleri

Aracın düz bir zeminde, ϕ yönünde, sabit bir hızla $v_c = \dot{x}_e$ ve $\dot{\phi}$ açısal hızında döndüğünü varsayalım. v_c ile x_e arasında kalan α açısı kayma açısı olarak adlandırılır. Sistemin hareket denklemleri aşağıdaki şekilde yazılabilir.

$$\begin{aligned} m\ddot{x}_e &= F_o + F_i - F_{cent}\sin(\alpha) - R_{ro} - R_{ri} \\ m\ddot{y}_e &= F_{cent}\cos(\alpha) - \mu_l W \\ I_z\ddot{\phi} &= M_Q - M_r \end{aligned} \tag{7.14}$$

Araca etkiyen merkez kaç kuvveti F_{cent}

$$F_{cent} = m\dot{\phi}^2 R \tag{7.15}$$

ve kütle merkezinin etrafındaki moment M_Q

$$M_Q = \frac{\left[(F_o - R_{ro}) - (F_i - R_{ri})\right]B}{2} \tag{7.16}$$

M_r dönüş direncinin momenti, I_z de aracın e_z e göre atalet momenti.

Kayma yüzdemiz i;

$$i = \frac{\frac{K}{l}}{\ln(\frac{F}{F_{\max}})} \tag{7.17}$$

Formüldeki $F_{\max}$ paletteki maksimum çekiş eforu. $F_{\max}$ da;

$$F_{\max} = 0.5(Ac + W\tan(\varphi)) \tag{7.18}$$

Genel olarak formülleri toparladığımızda;

$$\begin{aligned}
M_r &= \frac{Wl\mu_l}{3} \\
R_{ro} &= \frac{W\mu_r}{2} \\
R_{ri} &= \frac{W\mu_r}{2} \\
i_o &= \frac{\frac{K}{l}}{\ln(\frac{F_o}{Ac + W\tan(\varphi)})} \\
i_i &= \frac{\frac{K}{l}}{\ln(\frac{F_i}{Ac + W\tan(\varphi)})} \\
R &= \frac{B\left[\omega_o(1-i_o) + \omega_i(1-i_i)\right]}{2\left[\omega_i(1-i_i) - \omega_o(1-i_o)\right]} \\
\alpha &= \arctan(\frac{\dot{\phi}^2 l}{4g\mu_l}) \\
\dot{\phi} &= \frac{r\left[\omega_i(t)(1-i_o) - \omega_o(t)(1-i_i)\right]}{B}
\end{aligned} \tag{7.19}$$

formüllerde geçen F_o ve F_i iç ve dış paletlerdeki itme kuvvetleri, ω_o ve ω_i paletleri döndüren sistemin açısal hızlarıdır. Aracın dönüş hızı $\dot{\phi}$ herhangi bir sensör vasıtası ile ölçülebileceği gibi üstteki formülden de çıkartılabilir.

Elimizdeki bilgiler doğrultusunda 7.14 teki denklemi düzenleyecek olursak;

$$\begin{aligned}
\ddot{x}_e &= \frac{F_o + F_i - m\dot{\phi}^2 R\sin(\alpha) - \mu_r W}{m} - \frac{2M_r}{B} \\
\ddot{y}_e &= \dot{\phi}^2 R\cos(\alpha) - \mu_l g \\
\ddot{\phi} &= \left[\frac{(F_o - F_i) - (R_{ro} - R_{ri})}{2} B - \frac{Wl\mu_l}{3}\right]\frac{1}{I_z}
\end{aligned} \tag{7.20}$$

Dolayısı ile aracın kinematik ve dinamik denklemleri elde edilmiş olur. Eğer biz bu denklemleri matris formda yazmak istersek elde edeceğimiz matris;

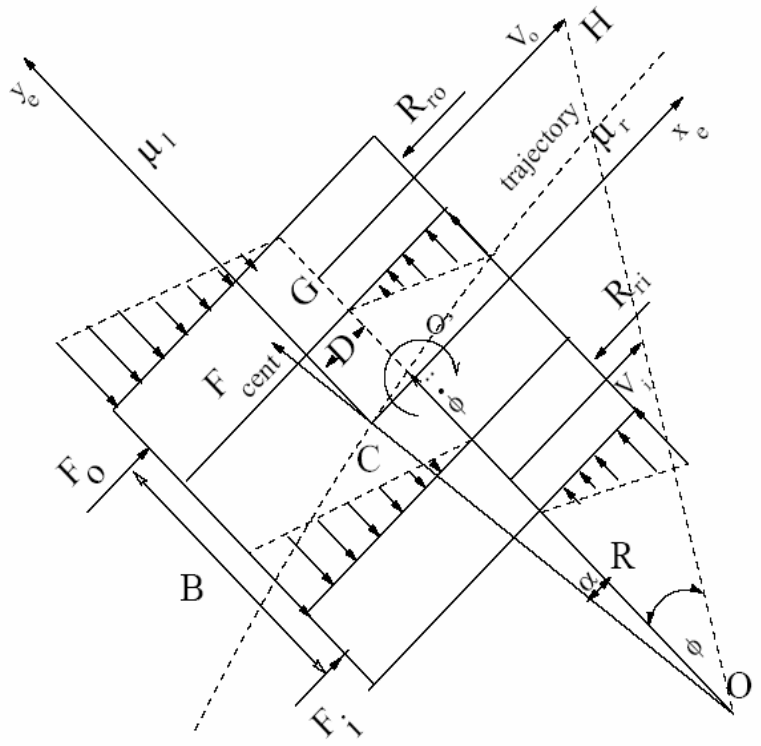
$$\frac{d}{dt}\begin{bmatrix} x_e \\ y_e \\ \phi \\ \dot{x}_e \\ \dot{y}_e \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\left[\omega_o(1-i_o) + \omega_i(1-i_i)\right] \\ \dot{y}_e \\ \frac{r\left[\omega_i(t)(1-i_o) - \omega_o(t)(1-i_i)\right]}{B} \\ \frac{F_o + F_i - m\dot{\phi}^2 R\sin(\alpha) - \mu_r W}{m} - \frac{2M_r}{B} \\ \dot{\phi}^2 R\cos(\alpha) - \mu_l g \\ \left[\frac{(F_o - F_i) - (R_{ro} - R_{ri})}{2} B - \frac{Wl\mu_l}{3}\right]\frac{1}{I_z} \end{bmatrix} \tag{7.21}$$

Buradan da dinamik ve kinematik denklemleri matris formunda yazarsak formül 7.21' i elde ederiz.

7.2.3 Sistemin Kinematik Modeli

Aşağıdaki grafikte aracın iki boyutlu kinematik modelini gösteren parametreler bulunmaktadır.

$$\frac{d}{dt}\begin{bmatrix} x_e \\ y_e \\ \phi \\ \dot{x}_e \\ \dot{y}_e \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\left[\omega_o(1-i_o)+\omega_i(1-i_i)\right] \\ \dot{y}_e \\ \frac{r\left[\omega_i(t)(1-i_o)-\omega_o(t)(1-i_i)\right]}{B} \\ \frac{(F_o+F_i)}{m}-\dot{\phi}^2 R\sin(\alpha)-g\mu_r-\frac{2M_r}{B} \\ \dot{\phi}^2 R\cos(\alpha)-\mu_l g \\ \left[\frac{B\left[(F_o-F_i)-(R_{ro}-R_{ri})\right]}{2I_z}-\frac{mgl\mu_l}{3I_z}\right] \end{bmatrix} \quad \textbf{(7.21)}$$



Şekil 7.7 Orta ve Yüksek Hızlarda Paletli Araç Modeli

Şekil 7.7 de araç sağa dönmektedir. Dış taraftaki ya da soldaki palet o iç taraftaki yâda sağdaki palet i ile tanımlanmıştır.

Paletlerdeki kaymayı sıfır olarak kabul edersek paletlerin hızı;

$$V_o = r\omega_o$$
$$V_i = r\omega_i \quad \textbf{(7.22)}$$

formül 7.22 de ki gibi olur. Burada r paleti döndüren mekanizmanın yarıçapı ω_o ve ω_i ise paleti döndüren mekanizmanın açısal hızıdır. Eğer yukarıdaki formüle boylamsal kayma yüzdelerini de eklersek elde edeceğimiz formül;

$$V_o = r\omega_o(1-i_o)$$
$$V_i = r\omega_i(1-i_i) \quad \textbf{(7.23)}$$

şekline gelir. Buradan yola çıkarak aracın ileri yöndeki bileşke hızı;

$$V_x = \frac{V_o + V_i}{2}$$
$$V_x = \frac{r\left[\omega_o(1-i_o) + \omega_i(1-i_i)\right]}{2} \quad \textbf{(7.24)}$$

V_o ile V_i arasındaki hız farkından dolayı ϕ açısı aşağıdaki gibi arctan fonksiyonu olarak yazılabilir.

$$\phi = \arctan(\frac{GH}{OG}) = \arctan(\frac{(V_o - V_i)t}{B}) \quad \textbf{(7.25)}$$

Formülde t zamanı, B iki arka tekerlek arasındaki mesafeyi belirtmektedir. Eğer ϕ açısını ufak zaman aralıkları ile türevini alırsak;

$$\dot{\phi} = \frac{\Delta V}{B}$$
$$\dot{\phi} = \frac{r\left[\omega_i(1-i_i) - \omega_o(1-i_o)\right]}{B} \quad \textbf{(7.26)}$$

Buradan yola çıkarak aracın x ve y eksenlerindeki hızları;

$$\dot{x} = \frac{r}{2}\left[\omega_o(1-i_o) + \omega_i(1-i_i)\right]\cos(\phi)$$
$$\dot{y} = \frac{r}{2}\left[\omega_o(1-i_o) + \omega_i(1-i_i)\right]\sin(\phi) \qquad \textbf{(7.27)}$$
$$\dot{\phi} = \frac{r\left[\omega_i(1-i_i) - \omega_o(1-i_o)\right]}{B}$$

Yukarıdaki denklemlere α kayma açısını da ilave ettiğimizde;

$$\dot{x} = \frac{r}{2}\left[\omega_o(1-i_o) + \omega_i(1-i_i)\right]\left[\cos(\phi t) - \sin(\phi t)\tan(\alpha t)\right]$$
$$\dot{y} = \frac{r}{2}\left[\omega_o(1-i_o) + \omega_i(1-i_i)\right]\left[\sin(\phi t) + \cos(\phi t)\tan(\alpha t)\right] \qquad \textbf{(7.28)}$$
$$\dot{\phi} = \frac{r\left[\omega_i(1-i_i) - \omega_o(1-i_o)\right]}{B}$$

Araç düz bir doğrultuda sabit hızda ilerlediği varsayılırsa;

$$\dot{i}_o = 0$$
$$\dot{i}_i = 0 \qquad \textbf{(7.29)}$$
$$\dot{\alpha} = 0$$

formüllerini elde ederiz.

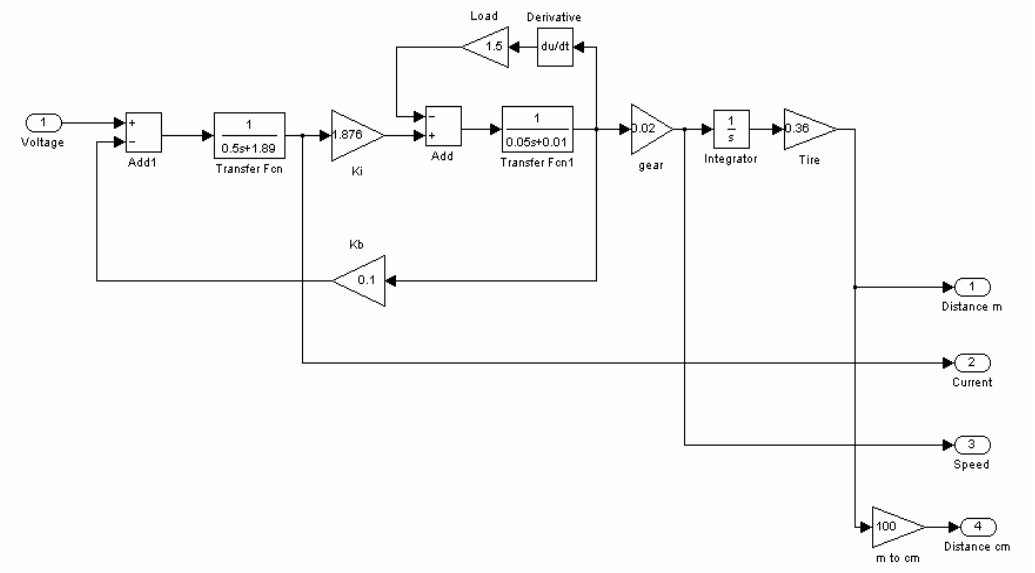
8 MATLAB SIMULINK SİMÜLASYONLARI

Bölüm 7 de çıkartılan matematiksel modeller Matlab Simulink' de gerçekleştirilerek çeşitli şartlar altında birçok simülasyon yapılmıştır. Elde edilen veriler yorumlanmış ve sistemin iyileştirilmesinde önemli katkılar sağlamıştır.

8.1 DC Motor Simülasyonları

İlk aşamada DC motor modeli MATLAB Simulink' de gerçekleştirildi. Model ilk simülasyonda yüksüz olarak denenmiş ve sonuçlar alınmıştır. İkinci aşamada motora aracın ağırlığının ¼ ü kadar yük bağlanıp simülasyon tekrarlanmıştır. Üçüncü aşamada mikrokontrolördeki PID algoritması yerleştirilip 50 cm de duracak şekilde simüle edilmiştir.

Şekil 8.1 de Simulink modeli gözükmektedir.

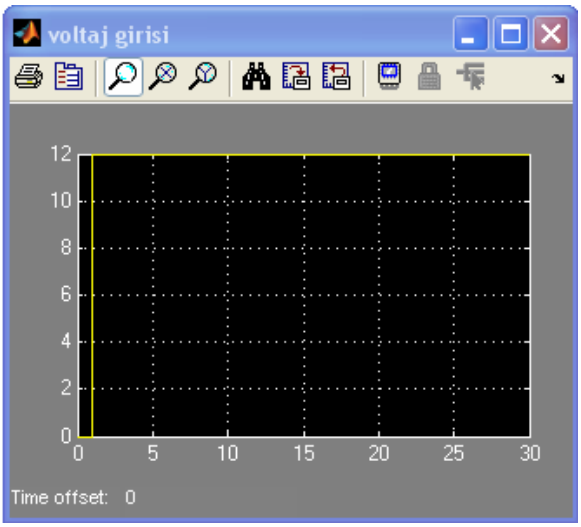


Şekil 8.1 DC Motor Simulink Modeli

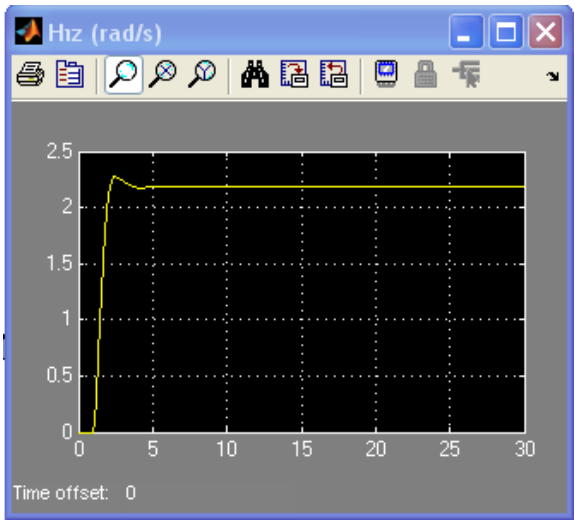
8.1.1 Yüksüz DC Motor Simülasyonları

Yüksüz olarak çalıştırıldığında ve girişine 12 voltluk step voltaj verdiğimizde elde edilen veriler aşağıda verilmiştir.

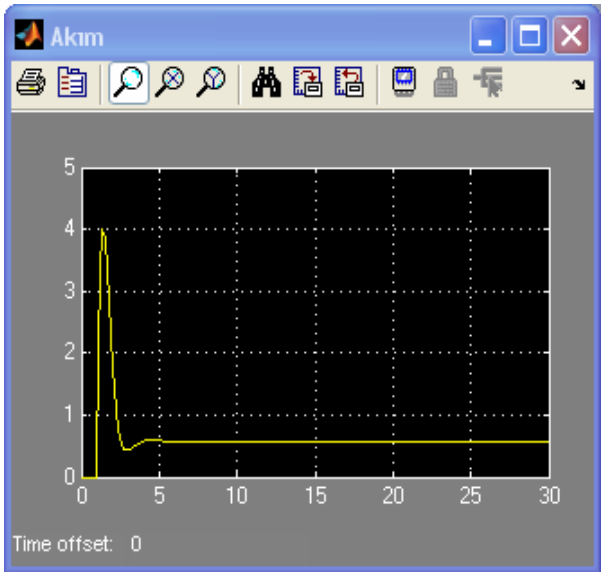
Modele bağlı yük devreden çıkartılmış ve bir saniye gecikmeli genliği 12 olan step girişi verilmiştir. Simülasyon da giriş voltajı, motor hızı ve akımı gözlemlenmiştir. Elde edilen sonuçlar şekil 8.2, şekil 8.3, şekil 8.4' de gözlemlenmektedir.



Şekil 8.2 Giriş Voltajı



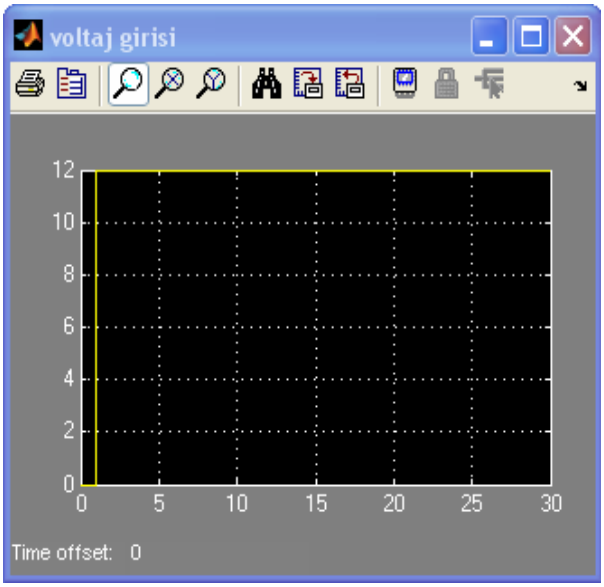
Şekil 8.3 Hız



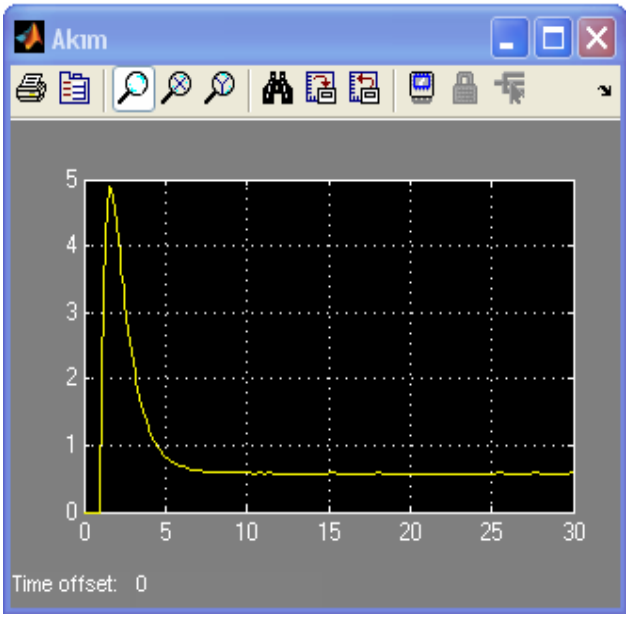
Şekil 8.4 Akım

8.1.2 Yüklü DC Motor Simülasyonları

Motora aracın ağırlığının ¼ ü kadar yük (1.5 kg) bağlayıp simülasyonu tekrarladığımızda elde edilen veriler aşağıdadır.



Şekil 8.5 Giriş Voltajı

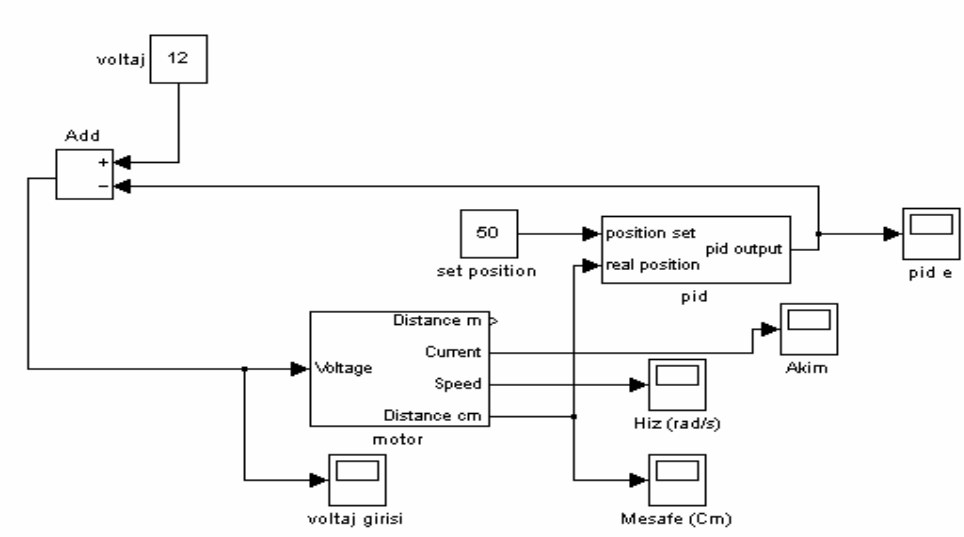


Şekil 8.6 Akım

Motora yükün bağlanması ile motorun nominal hızına ulaşmasının uzadığı, çekilen anlık peak akımın arttığı ve zamanının uzadığı gözlemlenmiştir.

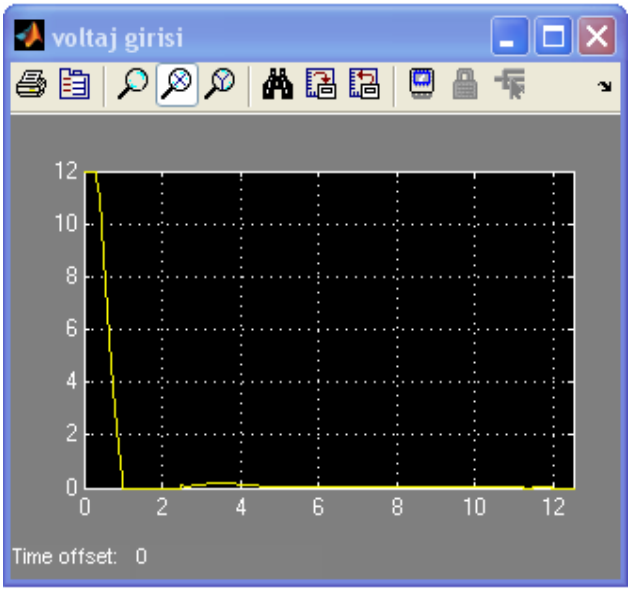
8.1.3 PID' li Yüklü DC Motor Simülasyonları

Motor simülasyonuna PID bloğunu ekleyip motoru 50 cm de duracak şekilde modeli oluşturduğumuzda Simulink modeli aşağıdaki gibi olmuştur.

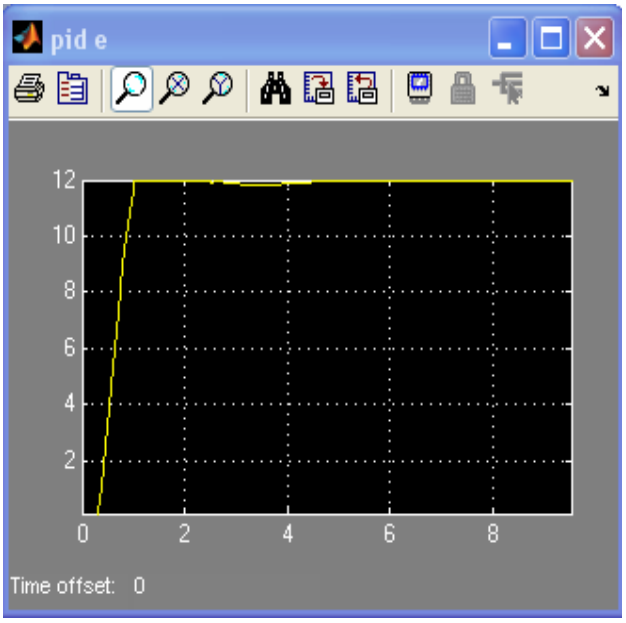


Şekil 8.7 Yüklü DC Motor Simulink Modeli

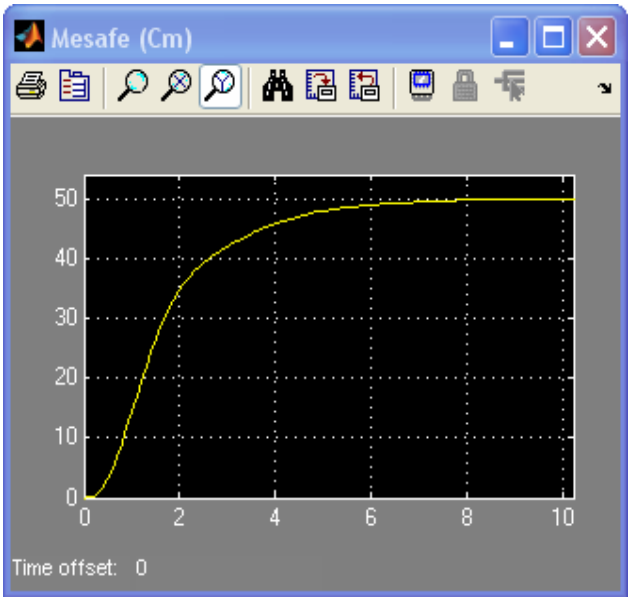
Elde edilen simülasyon verileri aşağıdadır.



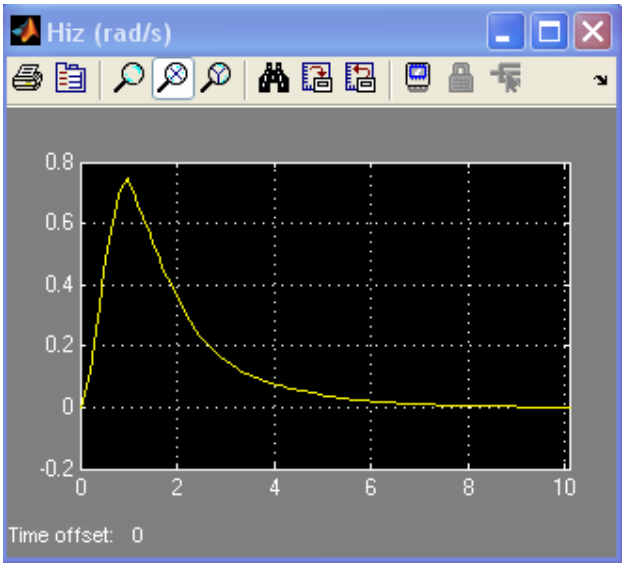
Şekil 8.8 Giriş Voltaj' ı



Şekil 8.9 PID Çıkışı

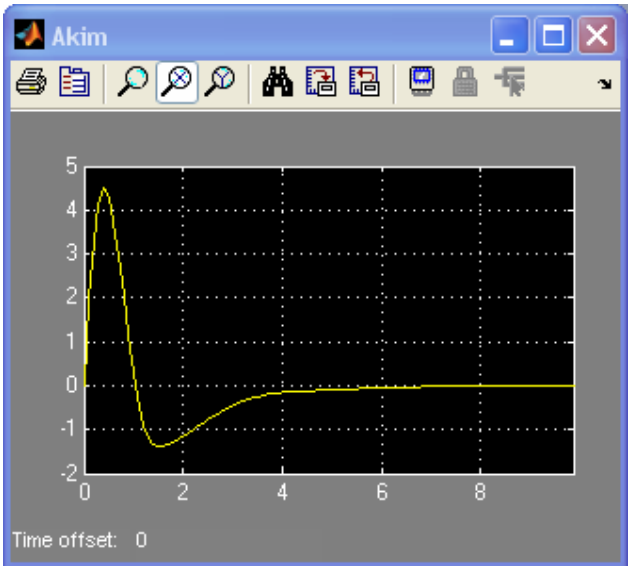


Şekil 8.10 Katedilen Mesafe



Şekil 8.11 Motorun Hızı

Simülasyonda kullanılan PID, sistemde kullanılan PID ile aynı mantıkta çalışmaktadır. 50 cm' lik bölgeye girdiğinde çalışmaya başlamakta ve motorun hızını besleme voltajını azaltarak kontrol etmektedir. Simülasyon limit bölgeden başlatılmış ve 50 cm sonra duracak şekilde parametreler verilmiştir. Motor durma noktasına belirli bir hızla yaklaşmış ve sonra yavaşlamaya başlamıştır. Motorun kalkış anında yüksek bir peak akım çektiği gözlenmiş belirli bir zaman sonunda yükün ataleti ile üzerinde biriken akımı beslemeye geri göndermiştir.

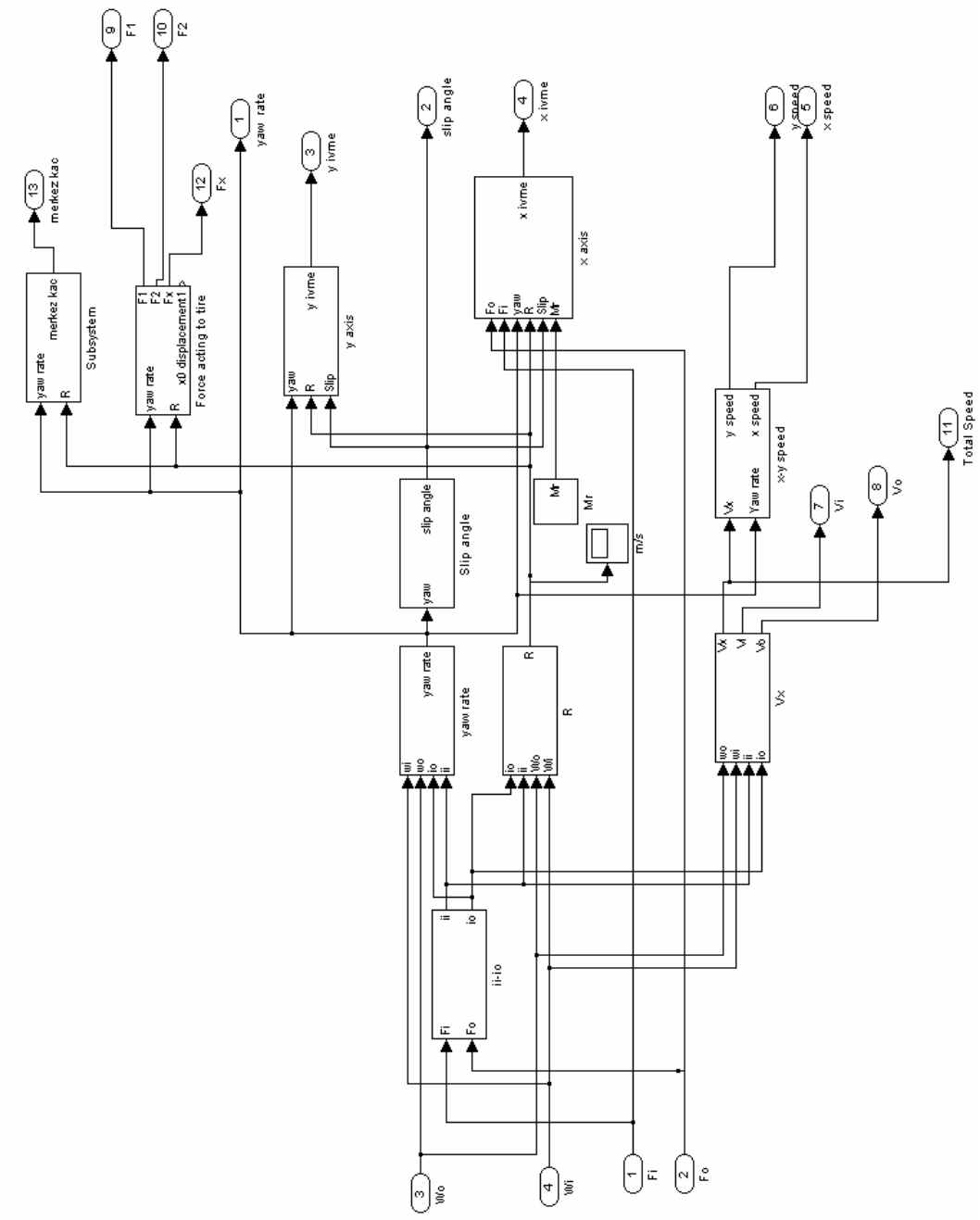


Şekil 8.12 DC Motor Akımı

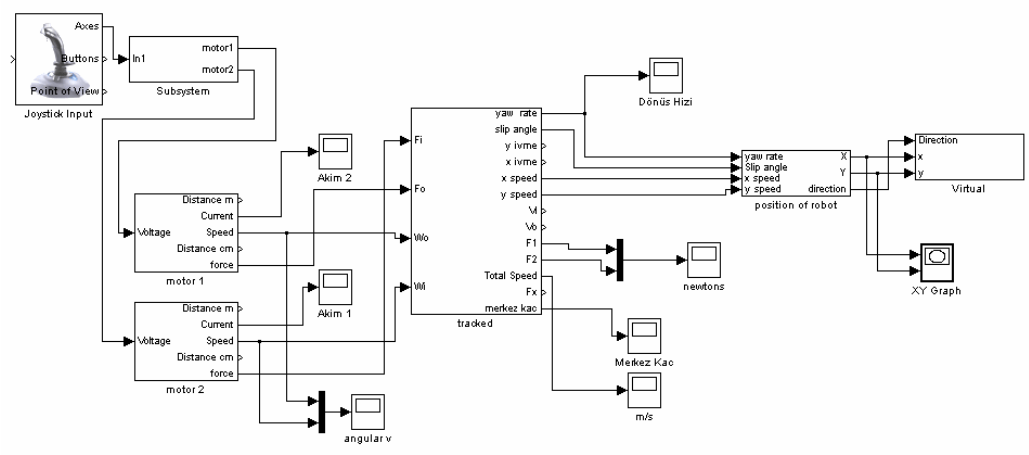
Gerçek sistemde bu durum devreleri bozabileceğinden besleme voltajı 0 değerine çok yaklaştığında fren işlemi yapılıp motorun üzerinde biriken akımı kendi üzerinde harcayarak elektronik donanım güvenceye alınmıştır. Yapılan simülasyonda durması gereken nokta 50 cm olarak ayarlanmış olmasına rağmen motorun ataletinden dolayı 50,3 cm de durabilmiştir. Hata oranı % 0,6 gibi bir orana sahiptir. Elde edilen verilerden tasarlanan PID' nin motorun hızını iyi kontrol ettiği sonucuna varabiliriz.

8.2 Mobil Robot Simülasyonları

Mobil robot simülasyonunda paletli araç modeli kullanılmış ve araca uyarlanmıştır. Simülasyonda daha önce oluşturulmuş olan DC motor modeli kullanılmıştır. Simülasyon noktasal dönüş, 20 metre ilerdeki cismi algıladığında sola dönüş ve serbest simülasyon olarak üç değişik koşulda yapılmıştır. Robotun davranış şeklini ve hareketlerini daha kolay anlayabilmek açısından simülasyona Simulink Virtual Toolbox' ından simülasyon verilerine göre davranan animasyon eklenmiştir. Simulink modeli şekil 8.13 ve şekil 8.14 de görülmektedir.



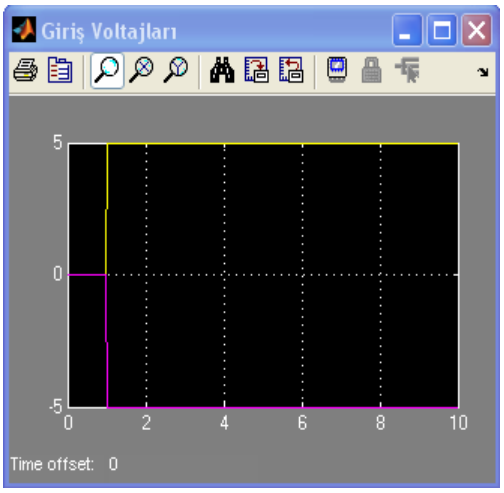
Şekil 8.13 Mobil Robot Simulink Modeli İçyapısı



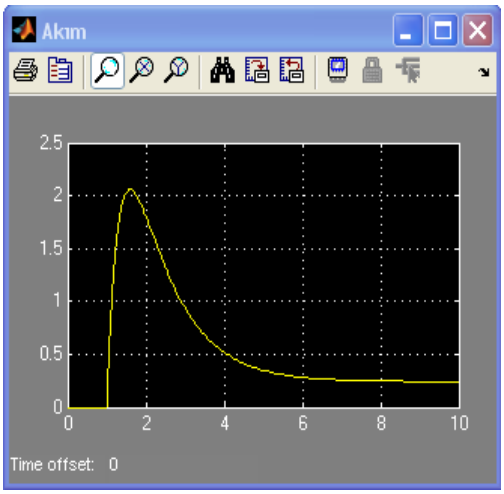
Şekil 8.14 Mobil Robot Simulink Modeli

8.2.1 Mobile Robotun Noktasal Dönüş Simülasyonları

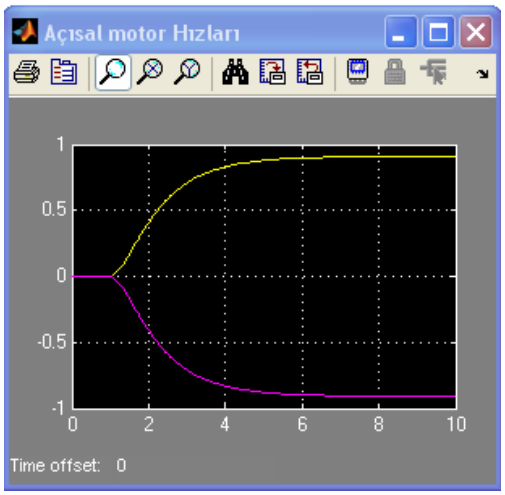
Noktasal dönüş yapabilmek için aracın iki tarafındaki motorlar aynı hızlarda fakat ters yönlerde dönmesi gerekmektedir. Bunu sağlamak için iki taraftaki motorlara genliği 5 VDC olan step giriş uygulanmıştır. Elde edilen veriler aşağıda verilmiştir.



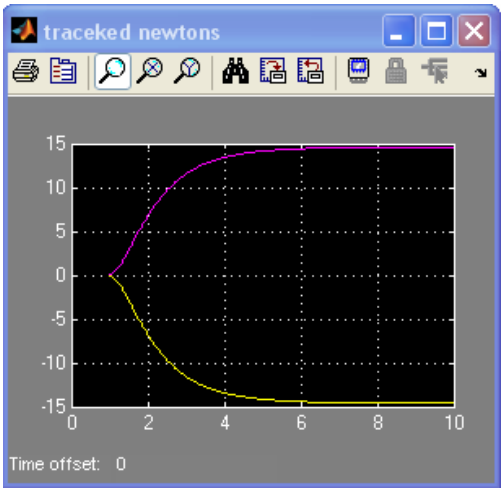
Şekil 8.15 Voltaj Girişleri



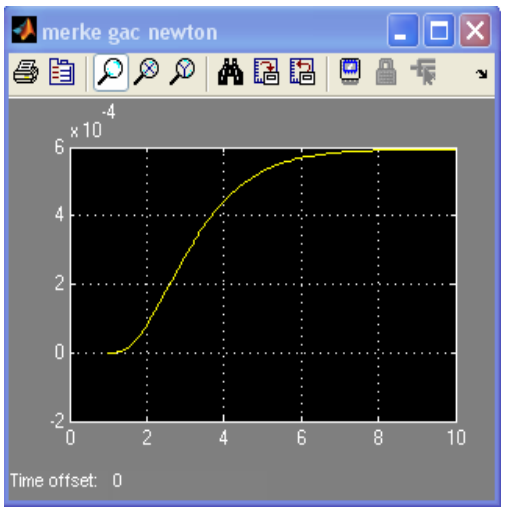
Şekil 8.16 DC Motor Akımı



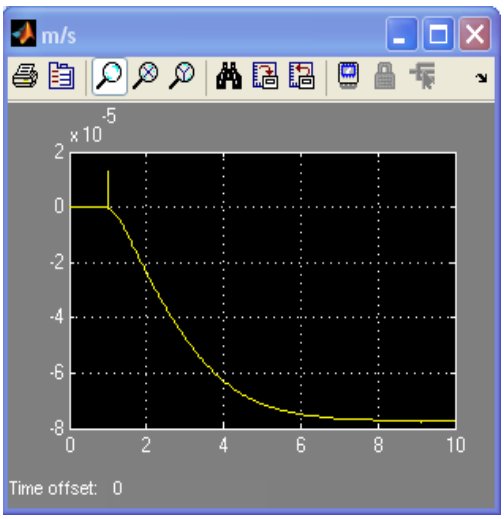
Şekil 8.17 Motor Hızları



Şekil 8.18 Tekerleklere Etkiyen Kuvvetler



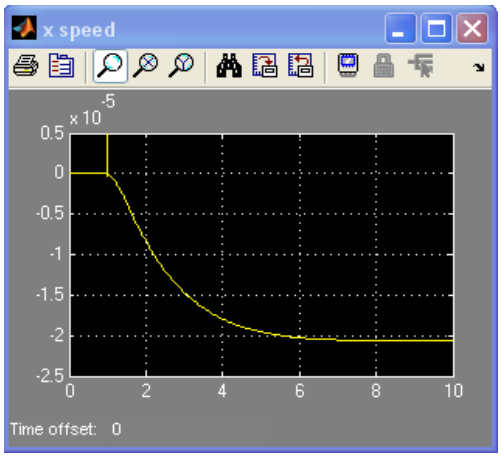
Şekil 8.19 Merkezkaç Kuvveti



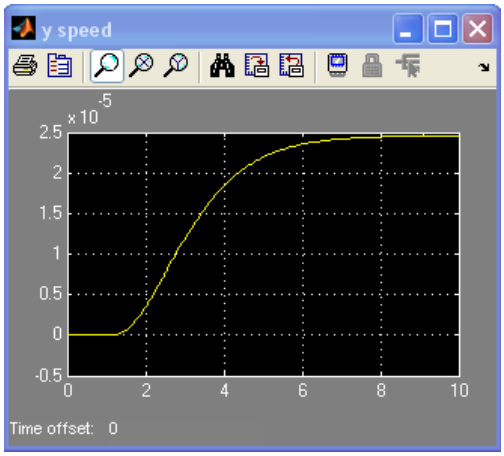
Şekil 8.20 Robotun Toplam Hızı

Yapılan simülasyondan elde edilen veriler doğrultusunda söyleyebiliriz ki araca etkiyen merkez kaç kuvveti aracın merkezinde beklenildiği gibi sıfıra yakın bir değere sahiptir. Tam sıfır olmamasının sebebi aracın tekerleklerinin 0.008 radyanlık kaymasından kaynaklanmaktadır. Bu kayma aracın x ve y yönlerindeki hızlarını dolayısı ile bileşke hızı etkilemekte ve çok düşük bir değer okumamıza yol açmaktadır.

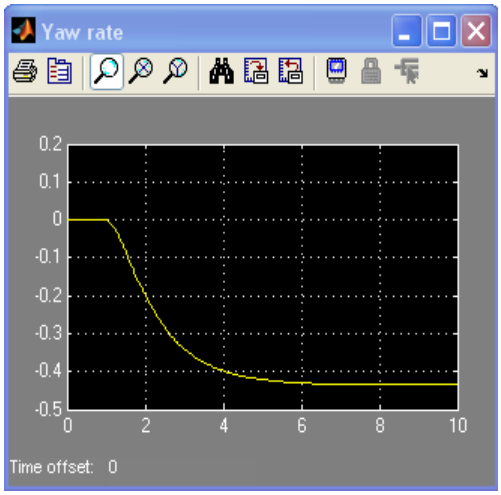
Aracın tekerleklerinin yere deydiği noktalardaki kuvvetler şekil 8.18 de gözükmektedir.



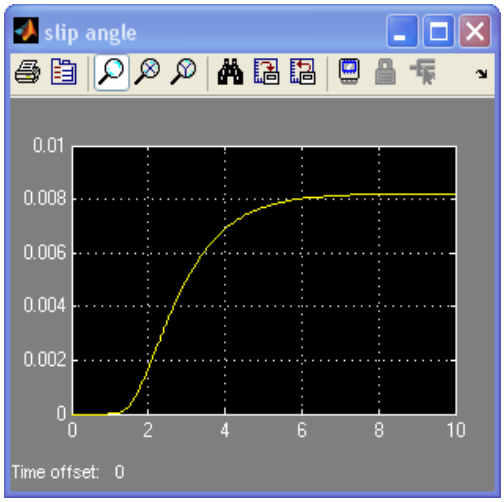
Şekil 8.21 X Yönünde ki Hız



Şekil 8.22 Y Yönündeki Hız



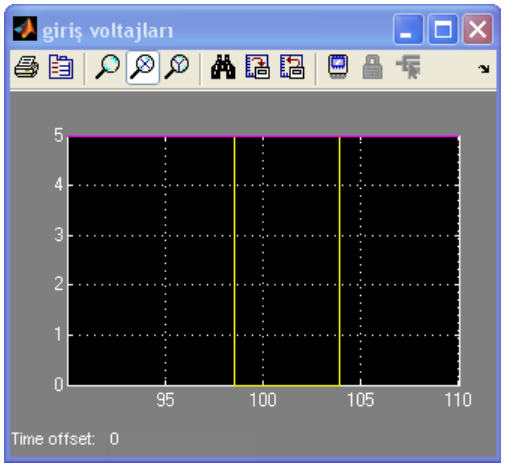
Şekil 8.23 Dönme Hızı



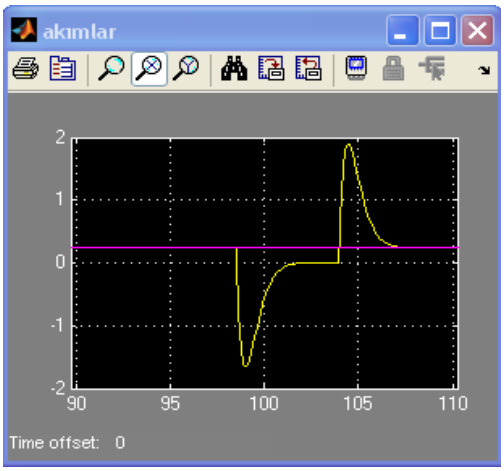
Şekil 8.24 Kayma Açısı

8.2.2 Mobile Robotun Engelden Kaçış Simülasyonu

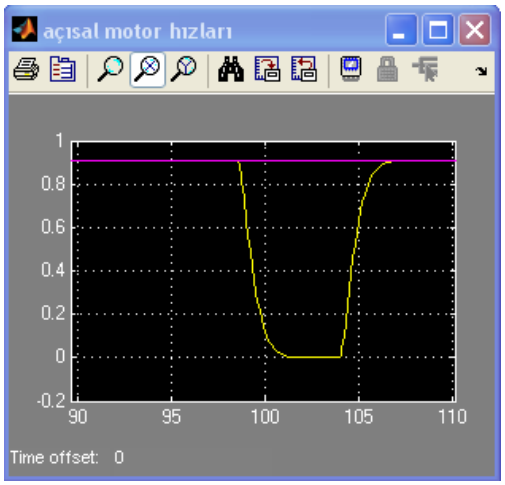
Robotun engelden kaçış simülasyonunda 2,5 metreye bir obje yerleştirilmiş ve 50 cm kala dönmesi sağlanmıştır. Dönüş esnasındaki veriler toplanmış ve yorumlanmıştır.



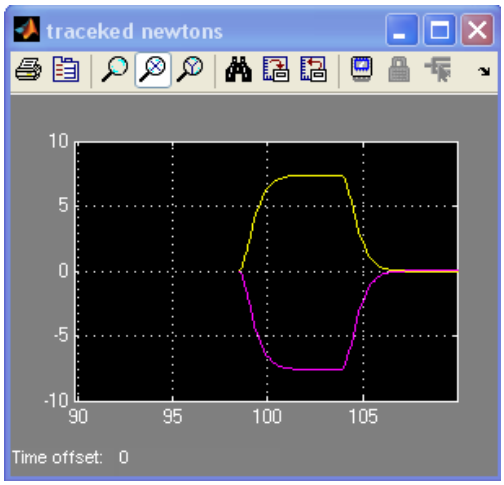
Şekil 8.25 Giriş Voltajları



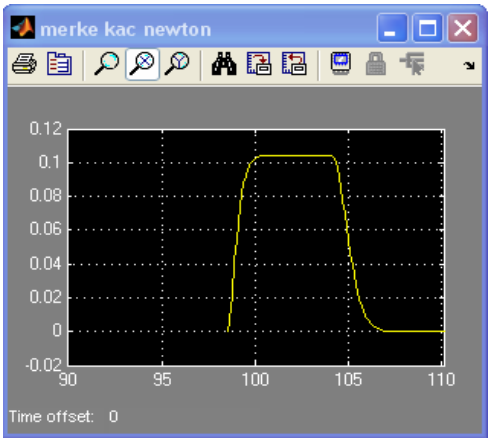
Şekil 8.26 DC Motor Akımları



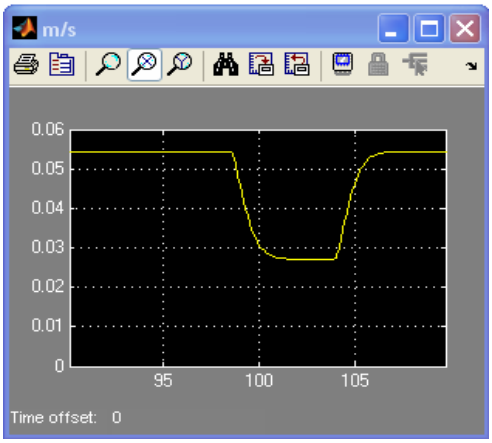
Şekil 8.27 Motor Hızlar' ı



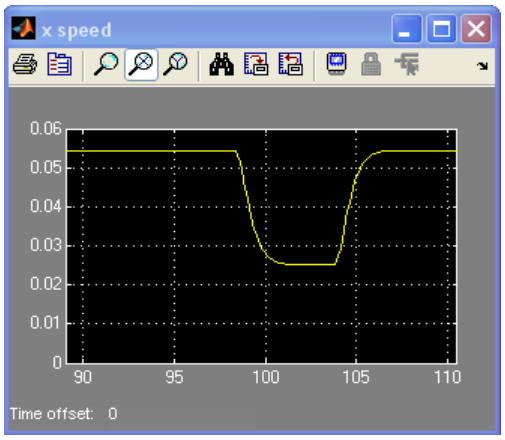
Şekil 8.28 Tekerleklere Etkiyen Kuvvetler



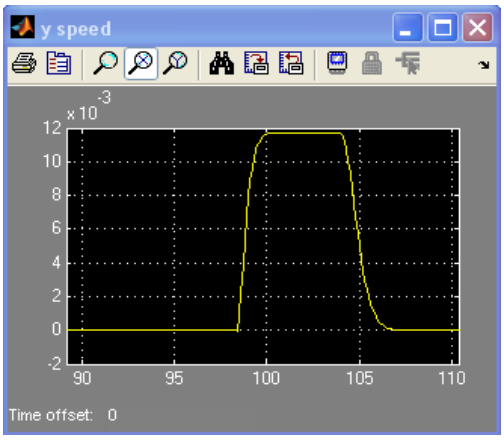
Şekil 8.29 Merkezkaç Kuvveti



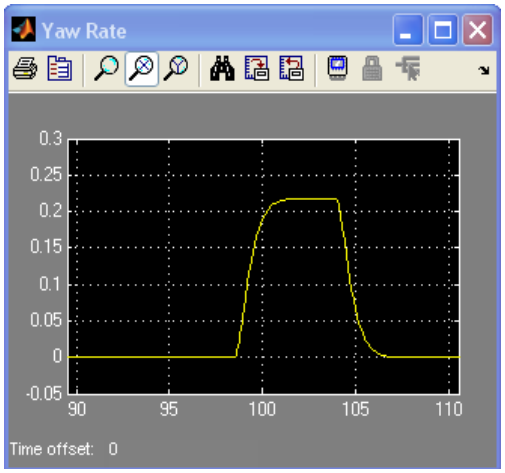
Şekil 8.30 Robotun Toplam Hızı



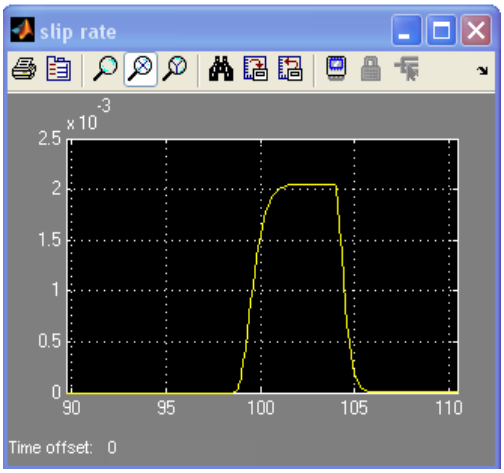
Şekil 8.31 X Yönünde ki Hız



Şekil 8.32 Y Yönünde ki Hız



Şekil 8.33 Dönme Hızı



Şekil 8.34 Kayma Açısı

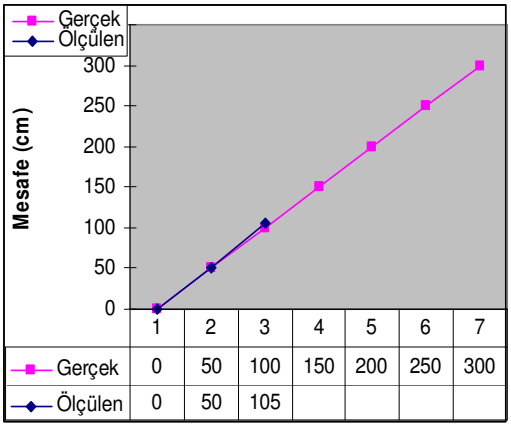
Yapılan simülasyonda motorlara 5 VDC uygulanmış 2,5 metredeki cisim 0,5 cm kala algılamıştır. Dönüş esnasında bir taraftaki tekerlekleri durdurmuş ve diğer taraftaki tekerlekleri aynı hızda döndürmeye devam etmiştir. Yapılan dönüş esnasında tekerleklere etkiyen kuvvet şekil 8.28 de görülmektedir. X yönündeki hız dönüş esnasında beklenildiği gibi azalıp y yönünde ki hız artış göstermiştir. Yapılan dönüş esnasında motorların çektiği akım şekil 8.26 de görülmektedir.

9 MOBİL ROBOT ÜZERİNDEN ALINAN ÖLÇÜMLER

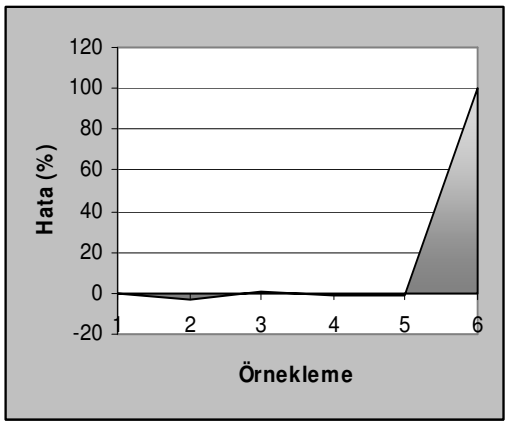
Mobil robot üzerinde bulunan dört adet ultrasonik mesafe sensörünün cisimlere duyarlılığı anlamak açısından değişik cisimler ile mesafe ölçümleri yapılmıştır. Test ortamı, çevresinde cisim bulunmayan bir ortam seçilmiş ve beş değişik obje ile farklı mesafelerden ölçümler alınmıştır. Objeler sensörü tam karşıdan görecek şekilde yerleştirilmiştir.

9.1 Mesafe Ölçümleri

Yapılan ilk ölçümde 3 cm çapına sahip yuvarlak bir obje sensörün önüne yerleştirilmiştir. Sensör, 50 ve 100 cm deki ölçümlerde objeyi tespit etmiş fakat diğer ölçümlerde başarısız olmuştur. Ölçümler ve hata oranları şekil 9.1 ve şekil 9.2 de verilmiştir.

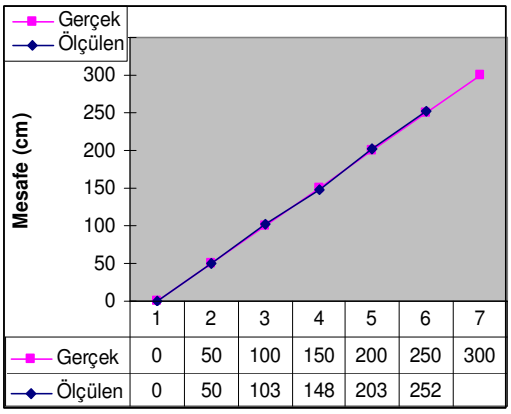


Şekil 9.1 Mesafe Ölçümü 1. Test

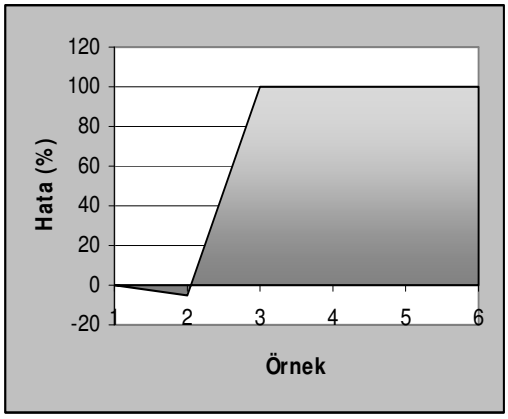


Şekil 9.2 Hata 1. Test

Bir sonraki teste objenin çapını 3 cm den 6 cm ye çıkartıp 50 şer cm ara ile ölçümler alınmıştır. Sensör 250 cm ye kadar cismi algılamış fakat 300 cm deki testte objeyi fark edememiştir. Ölçümler ve hata oranları şekil 9.3 ve şekil 9.4 de verilmiştir.

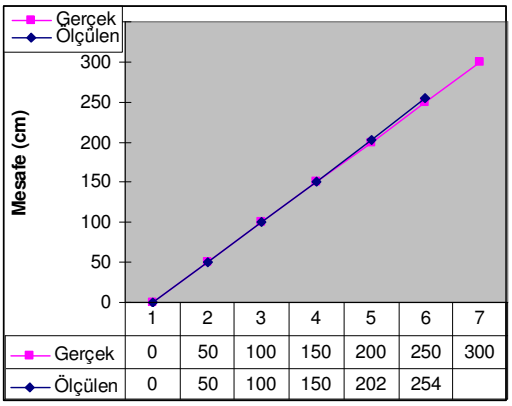


Şekil 9.3 Mesafe Ölçümü 2. Test

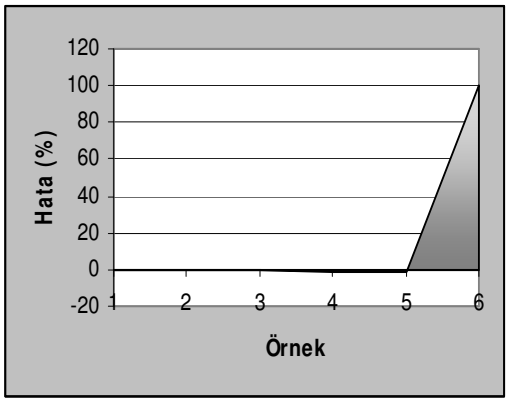


Şekil 9.4 Hata 2. Test

Objenin çapını 15 cm ye çıkarıp testi tekrarladığımızda sensör tekrardan 300 cm de cismi algılayamamıştır. Ölçümler ve hata oranları şekil 9.5 ve şekil 9.6 de verilmiştir.



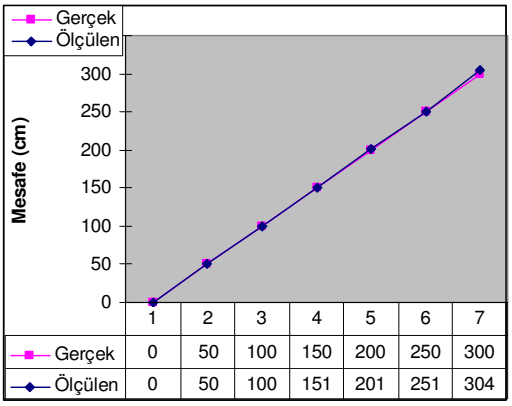
Şekil 9.5 Mesafe Ölçümü 3. Test



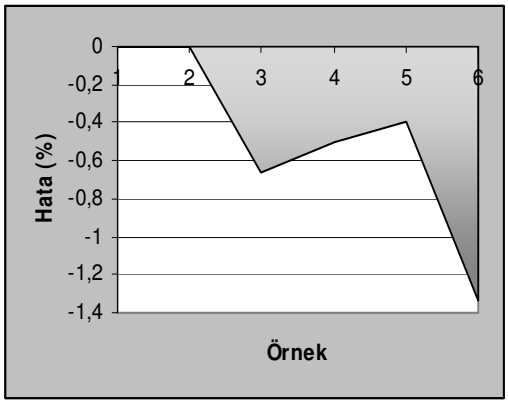
Şekil 9.6 Hata 3. Test

Bundan sonraki testlerde objenin şekli değiştirilip 5 cm enine sahip düz yüzeyli bir cisim seçilmiştir. Eninin 5 cm olmasına rağmen ultrasonik mesafe sensörü cismi 300 cm de tespit edebilmeyi başarmıştır. Ölçümler ve hata oranları şekil 9.7 ve şekil 9.8 de verilmiştir.

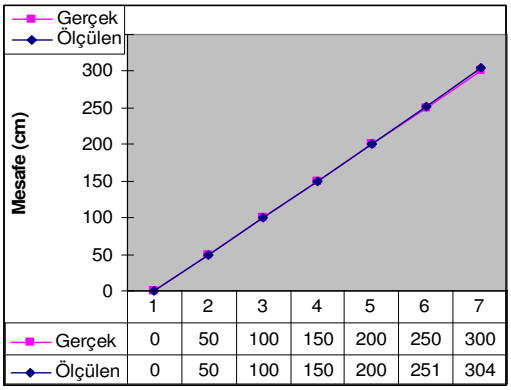
Cismin eni 5 cm den 8 cm ye çıkartılmış ve test tekrarlanmıştır. Bu test sonucunda da elde edilen veriler tatminkar bulunmuş ve daha geniş enli cisim ile test tekrarlanmamıştır. Ölçümler ve hata oranları şekil 9.9 ve şekil 9.10 da verilmiştir.



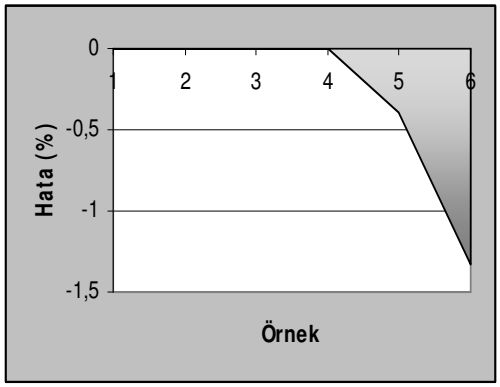
Şekil 9.7 Mesafe Ölçümü 4. Test



Şekil 9.8 Hata 4. Test



Şekil 9.9 Mesafe Ölçümü 5. Test



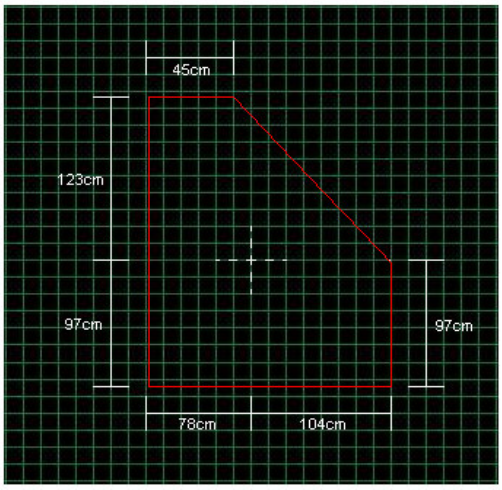
Şekil 9.10 Hata 5. Test

Yapılan ölçümlerde elde edilen sonuca göre ultrasonik sensörler yuvarlak hatlı cisimleri algılamakta düz yüzeye sahip cisimleri algılamaktan daha fazla zorluk çektiği ortaya çıkmıştır. Algılamada cismin şekli kadar eni ve çapı da önemli etken teşkil etmektedir. Elde edilen ölçümlerin hata payı 100 cm ye kadar cisimden ve şekilden bağımsız olarak %0 a yakın bir değerdedir. Hatalar 100 cm den itibaren başlamaktadır. Cismi algılamadığı durumlar haricinde hata oranı %5 ile 3 cm çapındaki cismin uzaklık ölçümünde yaşanmıştır. Bütün test sonuçlarına baktığımızda ölçüm hata oranı %1.5 in altındadır. Bu oran, ses hızının ortam sıcaklığı ile değiştiği göz önüne alındığında kabul edilebilir sınırlardadır.

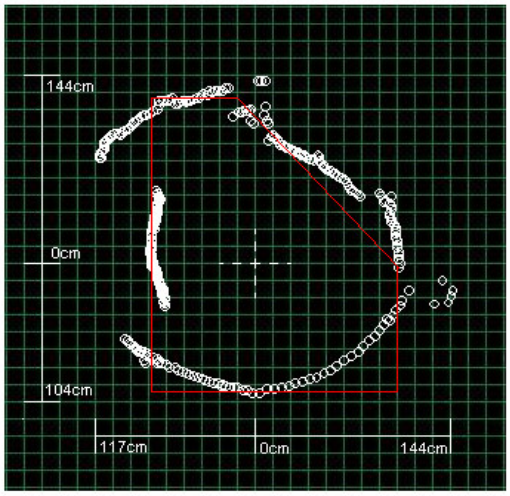
9.2 Noktasal Dönüşte Cisimleri Algılama

Ultrasonik sensörlerin açısal yüzeylerdeki ölçüm hassasiyetini anlamak için dört tarafı kapalı bir alanda ölçümler alınmıştır. Test ortamının ölçüleri şekil 9.11 de

verilmiştir. Robot, alanın işaretli yerine konulmuş ve yavaş bir hızla sağa doğru 360 derece döndürülmüştür. Alınan veriler Visual Basic programında yazılmış olan programda toplanıp işlenmiştir.



Şekil 9.11 Test Alanı

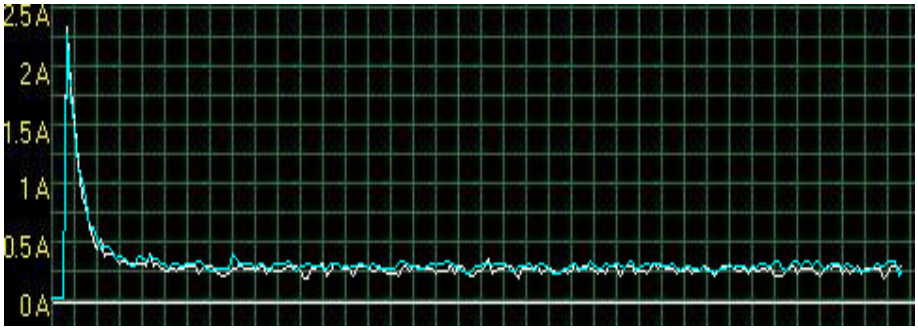


Şekil 9.12 Mesafe Ölçümleri

Yapılan testler sonucunda elde edilen verilerin yapılmış olan literatür araştırmalarındaki yüksek hata oranları ile uyuştuğu gözlenmiştir. Burada oluşan hata oranlarının sebebi olarak sensörün yolladığı ses sinyalinin cismin yüzeyine çarpıp yön değiştirmesi ve alıcıya gelene kadar daha uzun bir yol almasıdır. Bu testten çıkan diğer bir sonuç ise sensörü direkt gören düz yüzeyli cisimlerin algılanması ve mesafe ölçümü doğru yapılmasına karşın cismin yüzeyi sensörü 90 derece yerine başka bir açı ile gördüğünde mesafesinin doğru ölçülmesi yuvarlak hatlara sahip bir cismin mesafe ölçümünden daha zor ve hatalı gerçekleşmesidir.

9.3 Noktasal Dönüşte Motor Akımları

Matlab Simulink' de yapılmış olan noktasal dönüş simülasyonun da elde edilen akım değerlerini karşılaştırmak için bir test düzeneği kuruldu. İlk simülasyonda aracın tekerlekleri yerden kaldırıldı ve noktasal dönüş yapabilmesi için iki taraftaki motorlar aynı hızda fakat ters yönlere döndürüldüler. Sensörlerden gelen veriler akım yönünü bildirmediğinden elde edilen sonuçlarda akımlar pozitif yönde gözükmektedir. Şekilde gözüken her iki kare 1 saniyeyi belirtmektedir



Şekil 9.13 Akım Ölçümü

Beklendiği gibi motorlara voltaj verildiğinde anlık bir tepe değer gözlendi. Bu anlık tepe değer simülasyonda (Şekil 8.4) 4 A gibi hesaplanmışken ölçümlerde 2,5 amper olarak gözlemlenmiştir. Bunun sebebi oluşturulmuş olan DC motor modelinde gerekli parametrelerin motor sağlayıcısından temin edilememiş olması ve farklı değerler kullanılmış olmasıdır.

Motorun nominal hızına ulaşmasından sonra motorlar üzerindeki akım 300 ile 350 mA arasında ölçülmüştür. Bu değerler yapılan simülasyon ile uyuşmaktadır.

10 SONUÇ

Projede otonom çalışan mobil bir robot gerçekleştirilmiştir. Mobil robot otonom çalışabileceği gibi bilgisayar başındaki bir operatörün de robota müdahale edebileceği şekilde tasarlanmıştır.

Gerekli olan devreler tasarlanmış ve gerçekleştirilmiştir. Mobil platform üzerine sensörler yerleştirilmiş, gerekli yazılım ve kontrol algoritmaları mikrokontrolöre yazılmıştır. PID hız kontrolü gerçekleştirilmiş ve IF-THEN-ELSE mantığı ile çalışan mobil robotun kontrol algoritmasına uyarlanmıştır. PID algoritmasına alternatif olarak Fuzzy-Crisp algoritması ile hız kontrolü yapılmıştır. İki kontrol algoritması birbirleri ile karşılaştırılmış ve PID algoritmasının Fuzzy-Crisp algoritmasına oranla daha iyi bir kontrol gerçekleştidiği ortaya çıkmıştır. Bilgisayar ile verileri gözlemleyebilmek için yazılım geliştirilmiş ve başarı ile tamamlanmıştır.

Mobil robotta kullanılan DC motorların modelleri çıkartılmış ve Matlab Simulink' de simülasyonları yapılmıştır. PID algoritması test edilmiştir.

Mobil robotun yapısı paletli araç modeline uyduğundan, paletli araç modeli baz alınarak matematiksel modeli oluşturulmuş ve noktasal dönüş, engelden kaçış gibi Matlab Simulink simülasyonları yapılmıştır.

Yapılan simülasyonlardan elde edilen veriler gözlemlenip, yorumlanmış ve robot üzerinden alınan ölçümler ile karşılaştırılmıştır. Karşılaştırmalar sonucunda alınan verilerin yapılan simülasyonlar ile uyuştuğu ortaya çıkmıştır.

Mesafe ölçüm testlerinde elde edilen sonuç ise sensörü doğrudan gören düz yüzeyli cisimlerin algılanması ve mesafe ölçümü doğru yapılmasına karşın cismin yüzeyi sensörü doğrudan görmek yerine bir açı ile gördüğünde mesafenin doğru ölçülmesi yuvarlak hatlara sahip bir cismin mesafe ölçümünden daha zor ve hatalı gerçekleşmiştir.

Geliştirmeye açık bir tasarıma sahip olduğundan, mobil robota ileride enkoder yerleştirilerek konum kontrolü ve mapping algoritmaları gerçekleştirilebilir. Yine eklenecek bir dijital kamera ile görüntü işleme, obje takibi, belirlenen hedeflere ulaşma, en kısa yol (Shortest Path) gibi konular uygulanabilir.

Şu an kullanılan geleneksel PID algoritması yerine yapay zeka teknikleri olan bulanık mantık, yapay sinir ağları, genetik algoritmalar gibi kontrol algoritmaları gerçeklenebilir.

KAYNAKLAR

[1] **Momo, M.M. and Kime, R.C.**, 2001, Logic and Computer Design Fundamentals, Prentice Hall, New Jersey.

[2] **Proakis, G.J. and Salehi, M.**, 2002, Communication Systems Engineering, Prentice Hall, New Jersey.

[3] **Nilsson, W.J. and Riedel, A.S.**, 2001, Electric Circuits, Prentice Hall, New Jersey.

[4] **Sedra, S.A. and Smith, C.K.**, 1998, Microelectronic Circuits, Oxford University Press, New York.

[5] **Kuo, C.B.**, 1999, Automatic Control Systems, John Wiley & Sons, Inc, New York

[6] **Çölkesen, R.**, 1993, İşte C, Sistem Yayıncılık ve Matbaacılık, İstanbul

[7] **Le, T.A.**, 1999, Modelling and Control of Tracked Vehicles, *PhD Thesis*, The University of Sydney, Sydney

[8] **Borenstein, J. and Koren, Y.**, 1995, Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance, *IEEE Transaction on Robotics and Automation,* **11,** 132-138.

[9] **Ohyo, A., Ohno, T and Yuta, S.**,1996, X. Obstacle Detectability of Ultrasonic Ranging System and Sonar Map Understanding, University of Tsukuba, Tsukuba

[10] **Wijk, O.**, 2000, Triangulation-Based Fusion of Sonar Data with Application in Robot Pose Tracking, *IEEE Transaction on Robotics and Automation,* **16,** 740-752.

EKLER

Ek A (RoboTzan V1.38 Software)

```
Option Explicit
Dim Le As Integer
Const PI = 3.14159
Dim lWord As Long
Dim Data1 As Long
Dim Data2 As Long
Dim Data3 As Long
Dim Data4 As Long
Dim Data5 As Long
Dim Data6 As Long
Dim Data7 As Long
Dim Data8 As Long
Dim Data9 As Long
Dim angle1 As Long
Dim angle2 As Long
Dim angle3 As Long
Dim robotcon As Long
Dim curcount As Long
Dim curnow1 As Long
Dim curpre1 As Long
Dim curnow2 As Long
Dim curpre2 As Long
Dim mapcou As Long
Dim notman As Byte

Private Sub Combo1_Click()
   MSComm1.CommPort = Combo1.ListIndex + 1
End Sub
```

```
Private Sub Combo2_Click()
  If Combo2.ListIndex = 0 Then
    MSComm1.Settings = "9600,n,8,1"
  End If
  If Combo2.ListIndex = 1 Then
    MSComm1.Settings = "19200,n,8,1"
  End If
  If Combo2.ListIndex = 2 Then
    MSComm1.Settings = "38400,n,8,1"
  End If
End Sub

Private Sub Command1_Click()
  If (Combo1.ListIndex <> -1) And (Combo2.ListIndex <> -1) Then
    If MSComm1.PortOpen = False Then
      MSComm1.PortOpen = True
      Command1.Caption = "Disconnect"
      Frame5.Visible = True
      Frame7.Visible = True
    Else
      MSComm1.PortOpen = False
      Command1.Caption = "Connect"
      Frame5.Visible = False
      Frame7.Visible = False
    End If
  End If
  If Command1.Caption = "Disconnect" Then
    Combo1.Locked = True
    Combo2.Locked = True
  Else
    Combo1.Locked = False
    Combo2.Locked = False
  End If
End Sub

Private Sub Command10_Click()
Dim first As Byte
Dim second As Byte
```

```
Dim full As Double
full = HScroll1.Value
first = full \ 256
second = full Mod 256
MSComm1.Output = Chr$(40)
MSComm1.Output = Chr$(first)
MSComm1.Output = Chr$(second)
Text16.Text = Text15.Text
End Sub

Private Sub Command11_Click()
MSComm1.Output = Chr$(35)
End Sub

Private Sub Command12_Click()
Dim first As Byte
first = HScroll2.Value
MSComm1.Output = Chr$(45)
MSComm1.Output = Chr$(first)
Text17.Text = Text14.Text
End Sub

Private Sub Command2_Click()
MSComm1.Output = Chr$(5)
End Sub

Private Sub Command3_Click()
MSComm1.Output = Chr$(10)
End Sub

Private Sub Command4_Click()
MSComm1.Output = Chr$(15)
End Sub

Private Sub Command5_Click()
MSComm1.Output = Chr$(20)
End Sub
```

```
Private Sub Command6_Click()
MSComm1.Output = Chr$(25)
End Sub

Private Sub Command7_Click()
MSComm1.Output = Chr$(30)
End Sub

Private Sub Command8_Click()
Picture3.Picture = LoadPicture("")
Picture3.Picture = LoadPicture("back.jpg")
Shape7.Left = 2520 / (5175 / Picture3.ScaleWidth)
Shape7.Top = 3480 / (7155 / Picture3.ScaleHeight)
Command7.Caption = "Map On"
'Form1.Width = 8250
'Form1.Caption = "RoboTzan V1.34                                    Designed By Alper
TOZAN"
'Command8.Visible = False
Text12.Text = "Manual Mode"
Command7.Enabled = True
mapcou = 0
End Sub

Private Sub Data1_Validate(Action As Integer, Save As Integer)

End Sub

Private Sub Command9_Click()
SavePicture Picture4.Image, "Currents.BMP"
SavePicture Picture3.Image, "Map.BMP"
End Sub

Private Sub Form_Load()
   Frame5.Visible = False
   Frame7.Visible = False
   With MSComm1
     .DTREnable = False
     .EOFEnable = False
```

```
        .Handshaking = 0
        .RThreshold = 4
    End With
MSComm1.InputLen = 4
Command7.Caption = "Map On"
'Text13.Text = "PID"
Text14.Text = "30"
Text15.Text = "0"
Text16.Text = "400"
Text17.Text = "40"
Form1.Width = 14010
Form1.Caption="RoboTzan                                                    V1.38
Designed By Alper TOZAN"
curcount = 375
Picture4.Width = 13695
End Sub

Private Sub HScroll1_Change()
Text15.Text = CStr(HScroll1.Value)
End Sub

Private Sub HScroll2_Change()
Text14.Text = CStr(HScroll2.Value)
End Sub

Private Sub MSComm1_OnComm()
Dim data As Variant
Dim datab As Byte
'Dim Data1 As Integer
'Dim Data2 As Integer
'Dim Data3 As Integer
'Dim Data4 As Integer
'Dim Data5 As Integer
'Dim Data6 As Integer
'Dim Data7 As Integer
'Dim Data8 As Integer
'Dim data9 As Integer
Dim Data10 As Integer
```

```
Dim Count As Integer
Dim InBuff As String * 25

Dim sData As String

Dim DataHigh As Long
Dim DataLow As Long

Dim Unit As Long
Dim Check As Long
Dim tam As Long
Dim elde As Long
Dim robcon As Integer

    Select Case MSComm1.CommEvent
      ' Errors
        Case comEventBreak
        Case comEventCDTO
        Case comEventCTSTO
        Case comEventDSRTO
        Case comEventFrame
        Case comEventOverrun
        Case comEventRxOver
        Case comEventRxParity
        Case comEventTxFull
        Case comEventDCB

      ' Events
        Case comEvCD
        Case comEvCTS
        Case comEvDSR
        Case comEvRing
        Case comEvReceive
            sData = MSComm1.Input
            Unit = Asc(Mid$(sData, 1, 1))
            DataHigh = Asc(Mid$(sData, 2, 1))
            DataLow = Asc(Mid$(sData, 3, 1))
            Check = Asc(Mid$(sData, 4, 1))
```

```
lWord = (DataHigh * &H100) Or DataLow
'Text1.Text = CStr(lWord)
Select Case Unit
  Case 1
    Text3.Text = CStr(lWord)
    Data1 = lWord
  Case 2
    Text4.Text = CStr(lWord)
    Data2 = lWord
  Case 3
    Text5.Text = CStr(lWord)
    Data3 = lWord
  Case 4
    Text6.Text = CStr(lWord)
    Data4 = lWord
  Case 5
    Text1.Text = CStr(lWord)
    curpre1 = curnow1
    curnow1 = lWord
  Case 6
    Text2.Text = CStr(lWord)
    curpre2 = curnow2
    curnow2 = lWord
  Case 7
    tam = lWord / 10
    elde = lWord Mod 10
    Text11.Text = CStr(tam) + "," + CStr(elde)
    Data7 = lWord
    angle3 = lWord

    If (Command7.Caption <> "Map Off") Then

      angle1 = (lWord + 450) Mod 3600
      tam = angle1 / 10
      elde = lWord Mod 10
      Text7.Text = CStr(tam) + "," + CStr(elde)
      Data5 = angle1
```

```
            angle2 = (lWord + 3150) Mod 3600
            tam = angle2 / 10
            elde = angle2 Mod 10
            Text8.Text = CStr(tam) + "," + CStr(elde)
            Data6 = angle2

            Data8 = (lWord + 1800) Mod 3600
            tam = Data8 / 10
            elde = Data8 Mod 10
            Text10.Text = CStr(tam) + "," + CStr(elde)
          End If
          tam = lWord / 10
          elde = lWord Mod 10
          Data7 = lWord
          Text9.Text = CStr(tam) + "," + CStr(elde)

          Line1.X2 = 1320 * Cos(lWord * PI / 1800) + Line1.X1
          Line1.y2 = 1320 * Sin(lWord * PI / 1800) + Line1.y1
          Line2.X1 = 1320 * Cos((lWord - 1800) * PI / 1800) + Line2.X2
          Line2.y1 = 1320 * Sin((lWord - 1800) * PI / 1800) + Line2.y2
          If (Command7.Caption = "Map Off") Then ' And (Command8.Visible =
False) Then
            mapping
            Command7.Enabled = False
          End If
        Case 8
          robotcon = lWord
          If notman Then
            Select Case lWord
              Case 1
                Text12.Text = "Forward"
              Case 2
                Text12.Text = "Reverse"
              Case 4
                Text12.Text = "Right"
              Case 8
                Text12.Text = "Left"
            End Select
```

```
            End If
          Case 9
            If (lWord And &H8) Then
              Text12.Text = "Mapping"
              Command7.Caption = "Map Off"
              Form1.Width = 14010
              Form1.Caption="RoboTzan                                    V1.34
Designed By Alper TOZAN"
            End If
            If (lWord And &H4) Then
                If (Text12.Text <> "Mapping") Then
                   Text12.Text = "Manual Mode"
                End If
                Command2.Caption = "Disable"
                Command3.Enabled = True
                Command4.Enabled = True
                Command5.Enabled = True
                Command6.Enabled = True
                Command8.Enabled = True
                Command10.Enabled = True
                Command11.Enabled = True
                Command12.Enabled = True
                notman = 0
            Else
                Command2.Caption = "Enable"
                notman = 1
                Command3.Enabled = False
                Command4.Enabled = False
                Command5.Enabled = False
                Command6.Enabled = False
                Command7.Enabled = False
                Command8.Enabled = False
                Command10.Enabled = False
                Command11.Enabled = False
                Command12.Enabled = False
            End If
            If (lWord And &H2) Then
              Text13.Text = "Fuzzy"
```

```
                End If
                If (lWord And &H1) Then
                  Text13.Text = "PID"
                End If
              Case 45

              Case 46

            End Select

          Case comEvSend

          Case comEvEOF

        End Select
      placing
      Timer2.Enabled = True
End Sub

    Sub HandleInput(InBuff As String)

      Text1.SelText = InBuff
    End Sub
Function placing()
  Shape2.Left = (Data1 + 60) * 3.75 * Cos(Data5 * PI / 1800) + Shape6.Left
  Shape2.Top = (Data1 + 60) * 3.75 * Sin(Data5 * PI / 1800) + Shape6.Top + 60
  Shape3.Left = (Data2 + 60) * 3.75 * Cos(Data6 * PI / 1800) + Shape6.Left
  Shape3.Top = (Data2 + 60) * 3.75 * Sin(Data6 * PI / 1800) + Shape6.Top + 60
  Shape4.Left = (Data3 + 60) * 3.75 * Cos(Data7 * PI / 1800) + Shape6.Left
  Shape4.Top = (Data3 + 60) * 3.75 * Sin(Data7 * PI / 1800) + Shape6.Top + 60
  Shape5.Left = (Data4 + 60) * 3.75 * Cos(Data8 * PI / 1800) + Shape6.Left
  Shape5.Top = (Data4 + 60) * 3.75 * Sin(Data8 * PI / 1800) + Shape6.Top + 60
End Function

Function mapping()
Dim x3 As Long
Dim y3 As Long
```

```
  If (Data3 > 0) And (Data3 < 300) Then
    x3 = (Data3) * Cos(Data7 * PI / 1800) + Shape7.Left
    y3 = (Data3) * Sin(Data7 * PI / 1800) + Shape7.Top
    Picture3.Circle (x3, y3), (3)
  End If
  'mapcou = mapcou + 1
  'Text13.Text = CStr(mapcou)
End Function

Private Sub Timer2_Timer()
    Dim y1 As Long '1. motor
    Dim y2 As Long ' 1. motor
    Dim y3 As Long '2. motor
    Dim y4 As Long ' 2. motor
    curcount = curcount + 30
    y1 = 2500 - curpre1
    y2 = 2500 - curnow1
    Picture4.Line ((curcount - 30), y1)-(curcount, y2), RGB(255, 255, 255)
    curpre1 = curnow1
    y1 = 2500 - curpre2
    y2 = 2500 - curnow2
    Picture4.Line ((curcount - 30), y1)-(curcount, y2)
    curpre2 = curnow2
    If curcount > (Picture4.Width) Then
      curcount = 375
      Picture4.Picture = LoadPicture("back1.jpg")
    End If
    Timer2.Enabled = False
End Sub
```

Ek B (RoboTzan V1.24 FIRMWARE)

```
#device PIC18F452
#nolist
#define PIN_A0  31744
#define PIN_A1  31745
#define PIN_A2  31746
#define PIN_A3  31747
#define PIN_A4  31748
#define PIN_A5  31749
#define PIN_A6  31750
#define PIN_B0  31752
#define PIN_B1  31753
#define PIN_B2  31754
#define PIN_B3  31755
#define PIN_B4  31756
#define PIN_B5  31757
#define PIN_B6  31758
#define PIN_B7  31759
#define PIN_C0  31760
#define PIN_C1  31761
#define PIN_C2  31762
#define PIN_C3  31763
#define PIN_C4  31764
#define PIN_C5  31765
#define PIN_C6  31766
#define PIN_C7  31767
#define PIN_D0  31768
#define PIN_D1  31769
#define PIN_D2  31770
#define PIN_D3  31771
#define PIN_D4  31772
#define PIN_D5  31773
#define PIN_D6  31774
```

```
#define PIN_D7  31775
#define PIN_E0  31776
#define PIN_E1  31777
#define PIN_E2  31778
#define FALSE 0
#define TRUE 1
#define BYTE int
#define BOOLEAN short int
#define getc getch
#define fgetc getch
#define getchar getch
#define putc putchar
#define fputc putchar
#define fgets gets
#define fputs puts
#define WDT_TIMEOUT      4
#define MCLR_FROM_SLEEP  8
#define NORMAL_POWER_UP  12
#define BROWNOUT_RESTART 14
#define RTCC_INTERNAL   0
#define RTCC_EXT_L_TO_H 32
#define RTCC_EXT_H_TO_L 48
#define RTCC_DIV_1      8
#define RTCC_DIV_2      0
#define RTCC_DIV_4      1
#define RTCC_DIV_8      2
#define RTCC_DIV_16     3
#define RTCC_DIV_32     4
#define RTCC_DIV_64     5
#define RTCC_DIV_128    6
#define RTCC_DIV_256    7
#define RTCC_OFF        0x80
#define RTCC_8_BIT      0x40
#define WDT_ON      0x100
```

```
#define WDT_OFF     0
#define T1_DISABLED         0
#define T1_INTERNAL         0x85
#define T1_EXTERNAL         0x87
#define T1_EXTERNAL_SYNC    0x83
#define T1_CLK_OUT          8
#define T1_DIV_BY_1         0
#define T1_DIV_BY_2         0x10
#define T1_DIV_BY_4         0x20
#define T1_DIV_BY_8         0x30
#define T2_DISABLED         0
#define T2_DIV_BY_1         4
#define T2_DIV_BY_4         5
#define T2_DIV_BY_16        6
#define T3_DISABLED         0
#define T3_INTERNAL         0x85
#define T3_EXTERNAL         0x87
#define T3_EXTERNAL_SYNC    0x83
#define T3_DIV_BY_1         0
#define T3_DIV_BY_2         0x10
#define T3_DIV_BY_4         0x20
#define T3_DIV_BY_8         0x30
#define CCP_OFF                     0
#define CCP_CAPTURE_FE              4
#define CCP_CAPTURE_RE              5
#define CCP_CAPTURE_DIV_4           6
#define CCP_CAPTURE_DIV_16          7
#define CCP_COMPARE_SET_ON_MATCH    8
#define CCP_COMPARE_CLR_ON_MATCH    9
#define CCP_COMPARE_INT             0xA
#define CCP_COMPARE_INT_AND_TOGGLE  0x2
#define CCP_COMPARE_RESET_TIMER     0xB
#define CCP_PWM                     0xC
#define CCP_PWM_PLUS_1              0x1c
```

```
#define CCP_PWM_PLUS_2                0x2c
#define CCP_PWM_PLUS_3                0x3c
#define CCP_USE_TIMER3                0x100
long CCP_1;
#byte   CCP_1=0xfbe
#byte   CCP_1_LOW=0xfbe
#byte   CCP_1_HIGH=0xfbf
long CCP_2;
#byte   CCP_2=0xfbb
#byte   CCP_2_LOW=0xfbb
#byte   CCP_2_HIGH=0xfbc
#define PSP_ENABLED                   0x10
#define PSP_DISABLED                  0
#byte  PSP_DATA=0xF83
#define SPI_MASTER       0x20
#define SPI_SLAVE        0x24
#define SPI_L_TO_H       0
#define SPI_H_TO_L       0x10
#define SPI_CLK_DIV_4    0
#define SPI_CLK_DIV_16   1
#define SPI_CLK_DIV_64   2
#define SPI_CLK_T2       3
#define SPI_SS_DISABLED  1
#define SPI_SAMPLE_AT_END 0x8000
#define SPI_XMIT_L_TO_H  0x4000
#define UART_ADDRESS          2
#define UART_DATA             4
#define OSC_TIMER1  1
#define OSC_NORMAL  0
#define ADC_OFF               0
#define ADC_CLOCK_DIV_2   0x10000
#define ADC_CLOCK_DIV_4    0x4000
#define ADC_CLOCK_DIV_8    0x0040
#define ADC_CLOCK_DIV_16   0x4040
```

```
#define ADC_CLOCK_DIV_32   0x0080
#define ADC_CLOCK_DIV_64   0x4080
#define ADC_CLOCK_INTERNAL 0x00c0
#define NO_ANALOGS                      7
#define ALL_ANALOG                      0
#define AN0_AN1_AN2_AN4_AN5_AN6_AN7_VSS_VREF 1
#define AN0_AN1_AN2_AN3_AN4             2
#define AN0_AN1_AN2_AN4_VSS_VREF        3
#define AN0_AN1_AN3                     4
#define AN0_AN1_VSS_VREF                5
#define AN0_AN1_AN4_AN5_AN6_AN7_VREF_VREF 0x08
#define AN0_AN1_AN2_AN3_AN4_AN5         0x09
#define AN0_AN1_AN2_AN4_AN5_VSS_VREF    0x0A
#define AN0_AN1_AN4_AN5_VREF_VREF       0x0B
#define AN0_AN1_AN4_VREF_VREF           0x0C
#define AN0_AN1_VREF_VREF               0x0D
#define AN0                             0x0E
#define AN0_VREF_VREF                   0x0F
#define ANALOG_RA3_REF      0x1
#define A_ANALOG            0x2
#define A_ANALOG_RA3_REF    0x3
#define RA0_RA1_RA3_ANALOG  0x4
#define RA0_RA1_ANALOG_RA3_REF 0x5
#define ANALOG_RA3_RA2_REF          0x8
#define ANALOG_NOT_RE1_RE2          0x9
#define ANALOG_NOT_RE1_RE2_REF_RA3  0xA
#define ANALOG_NOT_RE1_RE2_REF_RA3_RA2  0xB
#define A_ANALOG_RA3_RA2_REF        0xC
#define RA0_RA1_ANALOG_RA3_RA2_REF  0xD
#define RA0_ANALOG                  0xE
#define RA0_ANALOG_RA3_RA2_REF      0xF
#define ADC_START_AND_READ  7
#define ADC_START_ONLY      1
#define ADC_READ_ONLY       6
```

```
#define L_TO_H            0x40
#define H_TO_L              0
#define GLOBAL              0xF2C0
#define INT_RTCC            0xF220
#define INT_TIMER0          0xF220
#define INT_TIMER1          0x9D01
#define INT_TIMER2          0x9D02
#define INT_TIMER3          0xA002
#define INT_EXT             0xF210
#define INT_EXT1            0xF008
#define INT_EXT2            0xF010
#define INT_RB              0xF208
#define INT_PSP             0x9D80
#define INT_AD              0x9D40
#define INT_RDA             0x9D20
#define INT_TBE             0x9D10
#define INT_SSP             0x9D08
#define INT_CCP1            0x9D04
#define INT_CCP2            0xA001
#define INT_BUSCOL          0xA008
#define INT_LOWVOLT         0xA004
#define INT_EEPROM          0xA010
#list
#use delay(clock=10000000,restart_wdt)
#use
rs232(baud=19200,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,RESTART_WDT)
#use i2c(Master,Fast,sda=PIN_C4,scl=PIN_C3)
int den1;
int den2;
#define  cSftVer 124
#use standard_io(A)
#use standard_io(B)
#use standard_io(C)
#use standard_io(D)
```

```
#use standard_io(E)
typedef  unsigned int16 word;
#define  pRAN1LS PIN_D0
#define  pRAN1TR PIN_D1
#define  pRAN2LS PIN_D2
#define  pRAN2TR PIN_D3
#define  pRAN3LS PIN_D4
#define  pRAN3TR PIN_D5
#define  pRAN4LS PIN_D6
#define  pRAN4TR PIN_D7
#define  pMOTBR1 PIN_A2
#define  pMOTBR2 PIN_A4
#define  pCOMFRE PIN_B4
#define  pCOMREN PIN_C0
#define  pMOTPW1 PIN_C1
#define  pMOTPW2 PIN_C2
#define  pMOTDI1 PIN_E1
#define  pMOTDI2 PIN_E2
#define RAN1
#define RAN2
#define RAN3
#define RAN4
#define COMP
#define  cRa1Com 450
#define  cRa2Com 3150
#define  cRa3Com 0
#define  cRa4Com 1800
#define cAdcDly 10

#define  paMORCUR 0
#define  paMOLCUR 1
word  sMorCur;
word  sMolCur;
word  sRa1Com;
```

```
word  sRa2Com;
word  sRa3Com;
word  sRa4Com;
word  sRanLs1;
word  sRanLs2;
word  sRanLs3;
word  sRanLs4;
word  sRanFi1;
word  sRanFi2;
word  sRanFi3;
word  sRanFi4;
int   sFilCou;
int   sTurCou;
word  sComDat;
int1  sRobSta;
int1  sRobSt1;
byte  sComRei;
#define  cMotFor 0x01
#define  cMotRev 0x02
#define  cMotLef 0x04
#define  cMotRig 0x08
#define  cMotStp 0x10
#define  cRobMan 0x01
#define  cRobFor 0x02
#define  cRobBac 0x04
#define  cRobLef 0x08
#define  cRobRig 0x10
#define  cRobMap 0x20
int      sMotCon;
int1     sMotCfg;
int1     sMotRvr;
int      sMotMem;
word     sMotPwm1;
word     sMotPwm2;
```

```
word     sPidPwm;
word     sRfrSpc;

#define  cDisThr 10
#define  cDisMin 100
#define  cDisMax 250
byte     sDisCos;
#define  cCurMin 1000
#define  cCurmax 3500
#define  cPdiInc 12
#define  cPcuInc 5
#define  cPdiPrc 7
#define  cPcuPrc 2
int32 pidd;
int32 pidc;
word  sRfdCou;
int   sRfrDat;
int1  sRecAve;
int1  sManInf;
int1  sMapInf;
int   sManDis;
int16   sManTur;
int   sManTrs;
int   sOtoTrs;
word  sComTar;
int1  sComTa1;
int1  sComTa2;
#define cManDis 100
#define cManTur 400
#define cManTrs 30
#define cOtoTrs 2
int16 right;
int16 left;
int16 center;
```

```
int16 sRobCon;
#define cRobPid 0x01
#define cRobFuz 0x02
#define cManMod 0x04
#define cMapMod 0x08
#define  cSlvTxc 9
word     sRegArr[cSlvTxc+1];
byte     sSlvTxb[5];
word     SUMM;

byte     sSlvTxc;
int      sCmdSta;
int1 alt;
int1 alt2;
int1 a;
#include "map.c"
#include "measure.c"
#include "compass.c"
#include "calcul.c"
#include "ai.c"
#include "pid.c"
#include "motor.c"
#include "arm.c"
#zero_ram

void main()
{
  setup_adc_ports(AN0_AN1_AN3);
  setup_adc(ADC_CLOCK_DIV_32);
  setup_psp(PSP_DISABLED);
  setup_wdt(WDT_OFF);
  setup_timer_0(RTCC_INTERNAL);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DIV_BY_1,255,1);
```

```
  setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
  setup_ccp1(CCP_PWM);
  setup_ccp2(CCP_PWM);
  enable_interrupts(INT_TBE);
  enable_interrupts(GLOBAL);
  output_high(pCOMREN);
  output_low(pin_b5);
  output_low(pin_b6);
  output_low(pin_b7);
  sFilCou=0;
  output_high(pMOTBR1);
  output_high(pMOTBR2);
  set_pwm1_duty(1);
  set_pwm2_duty(1);
  delay_ms(500);
  sRobCon|=cRobPid;
  sRfrSpc=400;
  sDisCos=40;
  while(1)
    {
#ifndef MOTTES
    if (!alt2)
      {
      if (sCmdSta&cRobMan)
        {
        if (sCmdSta&cRobMap)
          {
          sManDis=0;
          restart_wdt();
          measure();
          restart_wdt();
          compass();
          restart_wdt();
          sFilCou=3;
```

```
            }
          }
        else
          {
          measure();
          }
        }
      else
        {
        sFilCou=3;
        sRanFi1=0;
        sRanFi2=0;
        sRanFi3=0;
        sRanFi4=0;
        }
#else
sFilCou=3;
#endif
      restart_wdt();
      if (!(sCmdSta&cRobMan))
        compass();
      restart_wdt();
      calcul();
      restart_wdt();
      if (sFilCou>1)
        {
          sFilCou=0;
#ifndef MOTTES
        if (!(sCmdSta&cRobMan))
          {
          ai();
          restart_wdt();
          if (sRobCon==1)
            {
```

```
            pid();
            }
          if (sRobCon==2)
            {
            fuzzy();
            }
          restart_wdt();
          }
#endif
        if (sCmdSta&cRobMap)
          ai();
        motor();
        restart_wdt();
        arm();
        restart_wdt();
        map();
        }
    }
}

map()
{
if (!sRecAve)
  sRecAve=kbhit();
if (sRecAve)
  {
  sRfrDat=getc();
  sRecAve=0;
  if (sRfrDat==5)
    {
    if (sCmdSta&cRobMan)
      {
      sRobCon&=~cManMod;
      sCmdSta&=~cRobMan;
```

```
      sMotCon=cMotFor;
      }
    else
      {
      sRobCon|=cManMod;
      sCmdSta|=cRobMan;
      }
    sManInf=1;
    }
  if (sRfrDat==10)
    {
    sMotCon=cMotFor;
    sManTur=0;
    }
  if (sRfrDat==15)
    {
    sMotCon=cMotRev;
    sManTur=0;
    }
  if (sRfrDat==20)
    {
    sMotCon=cMotLef;
    sManTur=0;
    }
  if (sRfrDat==25)
    {
    sMotCon=cMotRig;
    sManTur=0;
    }
  if (sRfrDat==30)
    {
    sCmdSta|=cRobMap;
    sRobCon|=cMapMod;
    sMotCon=cMotRig;
```

```
    sComTar=sComDat;
    sComTa1=0;
    sComTa2=0;
    sMapInf=1;
    }
  if (sRfrDat==35)
    {
    if (sRobCon&cRobPid)
      {
      sRobCon&=~cRobPid;
      sRobCon|=cRobFuz;
      }
    else
      {
      sRobCon&=~cRobFuz;
      sRobCon|=cRobPid;
      }
    sManInf=1;
    }
  if (sRfrDat==40)
    {
    delay_ms(50);
    den1=getc();
    den2=getc();
    sRfrSpc=make16(den1,den2);
    }
  if (sRfrDat==45)
    {
    delay_ms(50);
    sDisCos=getc();
    }
  sRegArr[9]=sRobCon;
  }
a=input(pCOMFRE);
```

```
if ((sCmdSta&cRobMan)&(!(sCmdSta&cRobMap)))
  {
  if (sManTrs<cManTrs)
    {
    sManTrs=sManTrs+1;
    }
  else
    {
    if (a)
      {
      disable_interrupts(INT_TBE);
      }
    else
      {
      if (sManInf==1)
        {
        enable_interrupts(INT_TBE);
        sManTrs=0;
        }
      }
    }
  }
else
  {
  if(!a)
    {
    if (sCmdSta&cRobMap)
      {
      enable_interrupts(INT_TBE);
      }
    else
      {
      if(sOtoTrs<cOtoTrs)
        {
```

```
          sOtoTrs=sOtoTrs+1;
          }
        else
          {
          sOtoTrs=0;
          enable_interrupts(INT_TBE);
          }
        }
      }
   else
      {
      disable_interrupts(INT_TBE);
      }
   }
}

measure()
{
int1  ok;
int   count;
      sRanLs1=0;
      sRanLs2=0;
      sRanLs3=0;
      sRanLs4=0;
      ok=0;
#ifdef RAN1
   if (!(sCmdSta&cRobMap))
      {
      output_high(pRAN1TR);
      delay_us(40);
      output_low(pRAN1TR);
      while (!ok)
        {
        ok=input(pRAN1LS);
```

```
        }
      while (ok)
        {
        sRanLs1=sRanLs1+1;
        ok=input(pRAN1LS);
        }
      sRanLs1=sRanLs1/12;
      sRanFi1=sRanFi1+sRanLs1;
      }
#endif
      ok=0;
      if (!sRecAve)
        sRecAve=kbhit();
#ifdef RAN2
   if (!(sCmdSta&cRobMap))
      {
      output_high(pRAN2TR);
      delay_us(40);
      output_low(pRAN2TR);
      while (!ok)
        {
        ok=input(pRAN2LS);
        }
      while (ok)
        {
        sRanLs2=sRanLs2+1;
        ok=input(pRAN2LS);
        }
      sRanLs2=sRanLs2/12;
      sRanFi2=sRanFi2+sRanLs2;
      }
#endif
      ok=0;
      if (!sRecAve)
```

```
        sRecAve=kbhit();
#ifdef RAN3
      output_high(pRAN3TR);
      delay_us(40);
      output_low(pRAN3TR);
      while (!ok)
        {
        ok=input(pRAN3LS);
        }
      while (ok)
        {
        sRanLs3=sRanLs3+1;
        ok=input(pRAN3LS);
        }
      sRanLs3=sRanLs3/12;
      sRanFi3=sRanFi3+sRanLs3;
#endif
      ok=0;
      if (!sRecAve)
        sRecAve=kbhit();
#ifdef RAN4
   if (!(sCmdSta&cRobMap))
      {
      output_high(pRAN4TR);
      delay_us(100);
      output_low(pRAN4TR);
      while (!ok)
        {
        ok=input(pRAN4LS);
        }
      while (ok)
        {
        sRanLs4=sRanLs4+1;
        ok=input(pRAN4LS);
```

```
      }
    sRanLs4=sRanLs4/12;
    sRanFi4=sRanFi4+sRanLs4;
    }
#endif
    ok=0;
    if (!sRecAve)
      sRecAve=kbhit();
    sFilCou=sFilCou+1;
    if(sFilCou>1)
      {
      if (!(sCmdSta&cRobMan))
        {
        sRanLs1=sRanFi1>>1;
        sRanLs2=sRanFi2>>1;
        sRanLs3=sRanFi3>>1;
        sRanLs4=sRanFi4>>1;
        }
      else
        {
        sRanLs1=sRanFi1;
        sRanLs2=sRanFi2;
        sRanLs3=sRanFi3;
        sRanLs4=sRanFi4;
        }
      sRanFi1=0;
      sRanFi2=0;
      sRanFi3=0;
      sRanFi4=0;
      }
    set_adc_channel(paMORCUR);
    delay_us(cAdcDly);
    sMorCur=(word)read_adc();
    sMorCur=sMorCur*6
```

```
    set_adc_channel(paMOLCUR);
    delay_us(cAdcDly);
    sMolCur=(word)read_adc();
    sMolCur=sMolCur*6
    if (!sRecAve)
      sRecAve=kbhit();
}

compass()
{
#ifdef COMP
int   datah;
int   datal;
i2c_start();
i2c_write(0xC0);
i2c_write(0x02);
i2c_start();
i2c_write(0xC1);
datah=i2c_read();
datal=i2c_read(0);
i2c_stop();
sComDat=make16(datah,datal);
if (sComDat>4000)
  sComDat=0;
if (!sRecAve)
  sRecAve=kbhit();
#endif
}
```

```
calcul()
{
if (sComDat>3149)
  {
  sRa1Com=sComDat+cRa1Com;
  sRa1Com=sRa1Com-3600;
  }
else
  {
  sRa1Com=sComDat+cRa1Com;
  }
if (sComDat>549)
  {
  sRa2Com=sComDat+cRa2Com;
  sRa2Com=sRa2Com-3600;
  }
else
  {
  sRa2Com=sComDat+cRa2Com;
  }
if (sComDat>3599)
  {
  sRa3Com=sComDat+cRa3Com;
  sRa3Com=sRa3Com-3600;
  }
else
  {
  sRa3Com=sComDat+cRa3Com;
  }
if (sComDat>1799)
  {
  sRa4Com=sComDat+cRa4Com;
  sRa4Com=sRa4Com-3600;
  }
```

```
else
  {
  sRa4Com=sComDat+cRa4Com;
  }
}

ai()
{
if (!sRecAve)
  sRecAve=kbhit();
if (sMotCon==0)
  {
  if ((sRanLs1>15)&&((sRanLs2>15))&&((sRanLs3>15)))
    {
    sMotCon=cMotFor;
    sMotPwm1=sRfrSpc-sPidPwm;
    sMotPwm2=sRfrSpc-sPidPwm;
    }
  }
if(sMotCon&cMotFor)
  {
  if((sRanLs1<100)||(sRanLs2<100))
    {
    if ((sRanLs1)>(sRanLs2+40))
      {
      if (sRfrSpc>right)
        sMotPwm1=sRfrSpc-right;
      else
        sMotPwm1=0;
      if (800>(left+sRfrSpc))
        sMotPwm2=sRfrSpc+left;
      else
        sMotPwm2=800;
      }
```

```
    if (((sRanLs1+40)<(sRanLs2)))
      {
      if (800>(right+sRfrSpc))
        sMotPwm1=sRfrSpc+right;
      else
        sMotPwm1=800;
      if (sRfrSpc>left)
        sMotPwm2=sRfrSpc-left;
      else
        sMotPwm2=0;
      }
    }
  else
    {
    if (800>(center+sRfrSpc))
      {
      sMotPwm1=sRfrSpc+center;
      sMotPwm2=sRfrSpc+center;
      }
    else
      {
      sMotPwm1=800;
      sMotPwm2=800;
      }
    }
  }
if(sMotCon&cMotLef)
  {
  if (800>(right+sRfrSpc))
    sMotPwm1=sRfrSpc+right;
  else
    sMotPwm1=800;
  if (800>(left+sRfrSpc))
    sMotPwm2=sRfrSpc+left;
```

```
  else
    sMotPwm2=800;
  }
if(sMotCon&cMotRig)
  {
  if (800>(right+sRfrSpc))
    sMotPwm1=sRfrSpc+right;
  else
    sMotPwm1=800;
  if (800>(left+sRfrSpc))
    sMotPwm2=sRfrSpc+left;
  else
    sMotPwm2=800;
  }
if (sCmdSta&cRobMap)
  {
  if ((sComTar+50)<sComDat)
    {
    sComTa1=1;
    }
  if ((sComTa1)&&(sComTar>sComDat))
    {
    sComTa2=1;
    sMotCon=cMotStp;
    sCmdSta=sCmdSta&~cRobMap;
    sRobCon&=~cMapMod;
    sRegArr[9]=sRobCon;
    sMapInf=1;
    }
  sMotPwm1=sRfrSpc;
  sMotPwm2=sRfrSpc;
  }
else
  {
```

```
  if (sMotCon&cMotFor)
    {
    if(!sMotRvr)
      {
      if
((sRanLs1>sDisCos)&&(sRanLs2>sDisCos)&&(sRanLs3>(sDisCos+cDisThr)))
        {
        sMotCon=cMotFor;
        }
      }
    if (sRanLs3<sDisCos)
      {
      if (sRanLs1>=sRanLs2)
        {
        sMotCon=cMotLef;
        }
      else
        {
        sMotCon=cMotRig;
        }
      }
    if(sMotRvr&&(sMotCon&cMotLef))
      if ((sRanLs1>sDisCos)&&(sRanLs2>(sDisCos+cDisThr)))
        {
        sMotCon=cMotFor;
        sMotRvr=0;
        }
    if(sMotRvr&&(sMotCon&cMotRig))
      if ((sRanLs1>(sDisCos+cDisThr))&&(sRanLs2>sDisCos))
        {
        sMotCon=cMotFor;
        sMotRvr=0;
        }
    }
```

```
if (sMotCon&cMotFor)
  {
  if ((sRanLs1<sDisCos)&&(sRanLs2>sDisCos))
    {
    sMotCon=cMotRig;
    }
  }
if (sMotCon&cMotFor)
  {
  if ((sRanLs1>sDisCos)&&(sRanLs2<sDisCos))
    {
    sMotCon=cMotLef;
    }
  }
if (sMotCon&(cMotLef|cMotRig))
  {
  if (!sMotRvr)
    if ((sRanLs1>sDisCos)&&(sRanLs2>sDisCos)&&(sRanLs3>(sDisCos+20)))
      {
      sMotCon=cMotFor;
      }
  if(sMotRvr&&(sMotCon&cMotLef))
    if ((sRanLs1>sDisCos)&&(sRanLs2>(sDisCos+cDisThr)))
      {
      sMotCon=cMotFor;
      sMotRvr=0;
      }
  if(sMotRvr&&(sMotCon&cMotRig))
    if ((sRanLs1>(sDisCos+cDisThr))&&(sRanLs2>sDisCos))
      {
      sMotCon=cMotFor;
      sMotRvr=0;
      }
  }
```

```
#ifndef REV
  if (sMotCon&(cMotFor))
    {
    if ((sRanLs1<sDisCos)&&(sRanLs2<sDisCos))
      {
      if (sRanLs1>sRanLs2)
        {
        sMotCon=cMotLef;
        }
      else
        {
        sMotCon=cMotRig;
        }
      sMotRvr=1;
      }
    }
#endif
  }
if (!sRecAve)
  sRecAve=kbhit();
}

pid()
{
int32   d1,d2,c1,c2,d,c,iacc,dacc,pacc;
d=0;
c=0;
d1=0;
d2=0;
c1=0;
c2=0;
pacc=0;
if (!sRecAve)
  sRecAve=kbhit();
```

```
if (sRanLs1<cDisMin)
  {
  d1=cDisMin-sRanLs1;
  d1=d1*cPdiInc;
  }
if (sRanLs2<cDisMin)
  {
  d2=cDisMin-sRanLs2;
  d2=d2*cPdiInc;
  }
if (d1<d2)  {d=d2;}
else        {d=d1;}
if (d)
  {
  dacc+=d;
  if (dacc>500) dacc=500;
  }
if (sRanLs1>cDisMin)
  {
  d1=cDisMax-sRanLs1;
  d1=d1;//*cPdiInc;
  }
else
  d1=0;
if (sRanLs2>cDisMin)
  {
  d2=cDisMax-sRanLs2;
  d2=d2;//*cPdiInc;
  }
else
  d2=0;
if (d1<d2)  {d=d1;}
else        {d=d2;}
if (d)
```

```
        {
        if (d>dacc) {dacc=0;}
        else        {dacc-=d;}
        }
d1=0;
d2=0;
      if (sRanLs1<cDisMin)
        {
        d1=cDisMin-sRanLs1;
        d1=d1*cPdiPrc;
        }
      if (sRanLs2<cDisMin)
        {
        d2=cDisMin-sRanLs2;
        d2=d2*cPdiPrc;
        }
      if (d1<d2)  {d=d2;}
      else        {d=d1;}
      if (d)
        {
        pacc=d;
        }
      if (dacc)
        {
        if (sRanLs1>cDisMin)
          {
          d1=sRanLs1-cDisMin;
          d1=d1*cPdiPrc;
          }
        else
          d1=0;
        if (sRanLs2>cDisMin)
          {
          d2=sRanLs2-cDisMin;
```

```
          d2=d2*cPdiPrc;
          }
        else
          d2=0;
        if (d1<d2)  {d=d1;}
        else        {d=d2;}
        }
if (d)   {pacc=d;}
if (d)
  {
  if (pacc>dacc) {pidd=0;}
  else           {pidd=dacc-pacc;}
  }
else
  {pidd=pacc+dacc;}
pacc=0;
c1=0;
c2=0;
     if (sMorCur>cCurMin)
       {
       c1=sMorCur-cCurMin;
       c1=c1*cPcuInc;
       }
     if (sMolCur>cCurMin)
       {
       c2=sMolCur-cCurMin;
       c2=c2*cPcuInc;
       }
     if (c1<c2)  {c=c2;}
     else        {c=c1;}
     if (c)
       {
       if (c>iacc) {iacc=0;}
       else        {iacc-=c;}
```

```
        }
      if (sMorCur<cCurMin)
        {
        c1=cCurMin-sMorCur;
        c1=c1*cPcuInc;
        }
      else
        {
        c1=0;
        }
      if (sMolCur<cCurMin)
        {
        c2=cCurMin-sMolCur;
        c2=c2*cPcuInc;
        }
      else
        {
        c2=0;
        }
      if (c1<c2)  {c=c1;}
      else        {c=c2;}
      iacc+=c;
      if (iacc>100000) iacc=100000;
c1=0;
c2=0;
pacc=0;
      if (sMorCur>cCurMin)
        {
        c1=sMorCur-cCurMin;
        c1=c1*cPcuPrc;
        }
      if (sMolCur>cCurMin)
        {
        c2=sMolCur-cCurMin;
```

```
  c2=c2*cPcuPrc;
  }
if (c1<c2)  {c=c2;}
else        {c=c1;}
if (c)      {pacc=c;}
if (sMorCur<cCurMin)
  {
  c1=cCurMin-sMorCur;
  c1=c1*cPcuPrc;
  }
else
  {
  c1=0;
  }
if (sMolCur<cCurMin)
  {
  c2=cCurMin-sMolCur;
  c2=c2*cPcuPrc;
  }
else
  {
  c2=0;
  }
if (c1<c2)  {c=c1;}
else        {c=c2;}
if (c)   {pacc=c;}
pidc=pacc+iacc;
c=pidd>>1;
if (c>200) c=200;
sPidPwm=c;
if (!sRecAve) sRecAve=kbhit();
right=sPidPwm;
left=sPidPwm;
if (sRanLs3>cDisMin)
```

```
        {
        center=sRanLs3-cDisMin;
        center=center>>1;
        if (center>400)
          center=400;
        }
      else
        {
        center=cDisMin-sRanLs3;
        center=center>>1;
        if (center>300)
          center=300;
        }
}

fuzzy()
{
if (sRanLs1>cDisMin)
  {
  right=sRanLs1-cDisMin;
  right=right<<2;
  if (right>300)
    right=300;
  }
else
  {
  right=cDisMin-sRanLs1;
  right=right<<2;
  if (right>300)
    right=300;
  }
if (sRanLs2>cDisMin)
  {
  left=sRanLs2-cDisMin;
```

```
  left=left<<2;
  if (left>300)
    left=300;
  }
else
  {
  left=cDisMin-sRanLs2;
  left=left<<2;
  if (left>300)
    left=300;
  }
if (sRanLs3>cDisMin)
  {
  center=sRanLs3-cDisMin;
  center=center>>1;
  if (center>300)
    center=300;
  }
else
  {
  center=cDisMin-sRanLs3;
  center=center>>1;
  if (center>300)
    center=300;
  }
if (right<left)
  {
  right=left;
  }
else
  {
  left=right;
  }
}
```

```
motor()
{
if (!sRecAve)
  sRecAve=kbhit();
if (sCmdSta&cRobMan)
  {
  if (!(sCmdSta&cRobMap))
    {
    if (sManTur<cManTur)
      {
      sManTur=sManTur+1;
      sMotPwm1=sRfrSpc;
      sMotPwm2=sRfrSpc;
      }
    else
      {
      sMotCon=cMotStp;
      sMotPwm1=0;
      sMotPwm2=0;
      }
    }
  else
    {
    output_high(pMOTDI1);
    output_high(pMOTDI2);
    output_low(pMOTBR1);
    output_low(pMOTBR2);
    if(sMotCon!=sMotMem)
      sMotCfg=1;
    }
  }
  if(sMotCon&cMotFor)
    {
    output_low(pMOTDI1);
```

```
    output_low(pMOTDI2);
    output_low(pMOTBR1);
    output_low(pMOTBR2);
    if(sMotCon!=sMotMem)
      sMotCfg=1;
    }
  if(sMotCon&cMotLef)
    {
    output_high(pMOTDI1);
    output_low(pMOTDI2);
    output_low(pMOTBR1);
    output_low(pMOTBR2);
    if(sMotCon!=sMotMem)
      sMotCfg=1;
    }
  if(sMotCon&cMotRig)
    {
    output_low(pMOTDI1);
    output_high(pMOTDI2);
    output_low(pMOTBR1);
    output_low(pMOTBR2);
    if(sMotCon!=sMotMem)
      sMotCfg=1;
    }
  if(sMotCon&cMotRev)
    {
    output_high(pMOTDI1);
    output_high(pMOTDI2);
    output_low(pMOTBR1);
    output_low(pMOTBR2);
    if(sMotCon!=sMotMem)
      sMotCfg=1;
    }
  if(sMotCon&cMotStp)
```

```
    {
    output_low(pMOTDI1);
    output_low(pMOTDI2);
    }
  if(sMotCfg)
    {
  int16 l;
    if (sMotCon&cMotLef)
      {
      output_high(pMOTBR1);
      delay_ms(10);
      output_low(pMOTBR1);
      }
    if (sMotCon&cMotRig)
      {
      output_high(pMOTBR2);
      delay_ms(10);
      output_low(pMOTBR2);
      }
    sMotMem=sMotCon;
    sMotCfg=0;
    }
  else
  {
  set_pwm1_duty(sMotPwm1);
  set_pwm2_duty(sMotPwm2);
  }
#ifdef MOTTES
a=!a;
if (a)
  {
  sMotCon=sMotLef;
  }
else
```

```
  {
  sMotCon=sMotRig;
  }
  set_pwm1_duty(500);
set_pwm2_duty(500);
delay_ms(2000);
#endif
if (!sRecAve)
  sRecAve=kbhit();
}

#int_TBE
TBE_isr()
{
int a,b;
word c;
a=0;
if(sManInf||sMapInf)
  {
  sManInf=0;
  sMapInf=0;
  c=sRegArr[9];
  sSlvTxb[0]=9;
  sSlvTxb[1]=make8(c,1);
  sSlvTxb[2]=make8(c,0);
  sSlvTxb[3]=sSlvTxb[0]+sSlvTxb[1]+sSlvTxb[2];
  while (a<4)
    {
    putc(sSlvTxb[a]);
    a=a+1;
    }
  a=0;
  }
else
```

```
{
if (sCmdSta&cRobMap)
  {
  c=sRegArr[3];
  sSlvTxb[0]=3;
  sSlvTxb[1]=make8(c,1);
  sSlvTxb[2]=make8(c,0);
  sSlvTxb[3]=sSlvTxb[0]+sSlvTxb[1]+sSlvTxb[2];
  while (a<4)
   {
   putc(sSlvTxb[a]);
   a=a+1;
   }
  a=0;
  c=sRegArr[7];
  sSlvTxb[0]=7;
  sSlvTxb[1]=make8(c,1);
  sSlvTxb[2]=make8(c,0);
  sSlvTxb[3]=sSlvTxb[0]+sSlvTxb[1]+sSlvTxb[2];
  while (a<4)
   {
   putc(sSlvTxb[a]);
   a=a+1;
   }
  a=0;
  }
else
  {
  b=0;
  while (b<3)
    {
    b=b+1;
    if (cSlvTxc<=sSlvTxc)
      sSlvTxc=0;
```

```
        sSlvTxc=sSlvTxc+1;
        c=sRegArr[sSlvTxc];
        sSlvTxb[0]=sSlvTxc;
        sSlvTxb[1]=make8(c,1);
        sSlvTxb[2]=make8(c,0);
        sSlvTxb[3]=sSlvTxb[0]+sSlvTxb[1]+sSlvTxb[2];
        while (a<4)
          {
          putc(sSlvTxb[a]);
          a=a+1;
          }
        a=0;
        }
      }
  }
  disable_interrupts(INT_TBE);
}

arm()
{
if (!sRecAve)
  sRecAve=kbhit();
sRegArr[1]=sRanLs1;
sRegArr[2]=sRanLs2;
sRegArr[3]=sRanLs3;
sRegArr[4]=sRanLs4;
sRegArr[5]=sMorCur;
sRegArr[6]=sMolCur;
sRegArr[7]=sComDat;
sRegArr[8]=sMotCon;
sRegArr[9]=sRobCon;
restart_wdt();
}
```

ÖZGEÇMİŞ

Alper TOZAN, 1 Kasım 1979 tarihinde Kütahya' da doğdu. İlk, orta ve lise eğitimini Bursa' da bitirdi. 2004 yılında Yeditepe Üniversitesi, Mühendislik Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği bölümünden 3.5 senede mezun oldu. Aynı yıl İTU Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Programı' nda yüksek lisansa başladı.