



**KTO KARATAY
ÜNİVERSİTESİ**

T.C.

KTO KARATAY ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**OTONOM BİR ROBOTUN BULANIK KONTROLÖR YAKLAŞIMI İLE
KONUM KONTROLÜ**

Ali ÇETİNKAYA

KONYA

Şubat, 2017

OTONOM BİR ROBOTUN BULANIK KONTROLÖR YAKLAŞIMI İLE KONUM
KONTROLÜ

Ali ÇETİNKAYA

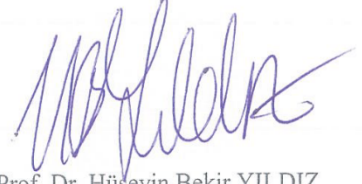
KTO Karatay Üniversitesi Fen Bilimleri Enstitüsü

Elektrik ve Bilgisayar Mühendisliği Ana Bilim Dalı Yüksek Lisans Programı

Yüksek Lisans Tezi

Şubat, 2017

Fen Bilimleri Enstitü Onayı



Prof. Dr. Hüseyin Bekir YILDIZ

Enstitüsü Müdürü

Bu tezli yüksek lisans tezinin yapılması gereken bütün gerekliliklerinin yerine getirdiğini onaylıyorum.



Yrd. Doç. Dr. H. Oktay ALTUN

Anabilim Dalı Başkanı

ALİ ÇETİNKAYA tarafından hazırlanan OTONOM BİR ROBOTUN BULANIK KONTROLÖR YAKLAŞIMI İLE KONUM KONTROLÜ başlıklı bu çalışma 28.02.2017 tarihinde yapılan savunma sınavı sonucunda başarılı bulunarak jüri tarafından tezli yüksek lisans tezi olarak kabul edilmiştir.



Prof. Dr. Novruz ALLAHVERDİ

Tez Danışmanı

Tez Jüri Üyeleri

Başkan: Prof. Dr. Novruz ALLAHVERDİ

Üye : Doç. Dr. Ali Bülent UŞAKLI

Üye : Yrd. Doç. Dr. Erdiñ KOÇER

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu bildirir aksi bir durumda aleyhime doğabilecek tüm hak ve kayıplarını kabullendiğimi beyan ederim.

28.02.2017

Ali ÇETİNKAYA



ÖZET

OTONOM BİR ROBOTUN BULANIK KONTROLÖR YAKLAŞIMI İLE KONUM KONTROLÜ

ÇETİNKAYA, ALİ

Yüksek Lisans- Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof.Dr. Novruz ALLAHVERDİ

Şubat, 2017

Bu tez çalışmasında, otonom bir robotun Bulanık kontrolör yaklaşımı ile sabit pist üzerinde konum kontrolünün uygulaması gerçekleştirilmiştir. Bu amaçla hem donanım hem de yazılım olarak gerçek zamanlı veri alışverişi yapılabilecek bir sistem tasarlanmıştır.

Bulanık kontrolör ile hedefe ulaşma ve konum kontrolü sağlamak için donanım bölümünde gömülü sistem (Atmel 2560 Entegresi) kullanılmıştır. Bu sistem üzerinden DC motor, servo motor, ultrasonik sensor, bluetooth modül ve optik sensör kontrolü yapılmıştır. Ayrıca mikrodenetleyici yazılımı ve Visual Studio 2015 platformunda c# dili kullanılarak masaüstü yazılımı tasarlanmıştır. Tasarlanan pist üzerinde gerçek zamanlı olarak geliştirilen Altu robot tarafından gönderilen koordinat (x, y) ve rota bilgileri Bulanık kontrolör ile işlenmiştir. Elde edilen veriler Altu robotun masaüstü yazılımında oluşturulan harita ile konum bilgileri ve robotun hareketleri izlenmektedir.

Bulanık kontrolör yaklaşımı kullanılarak yapılan uygulamalar ayrıca geleneksel kontrol yöntemi ile denenmiştir ve sonuçları karşılaştırılmıştır.

Altu robot pist üzerinde herhangi bir noktada çalıştırıldığında başlangıç konumunu hesaplayarak masaüstü yazılımına bluetooth modül aracılığıyla konum bilgisini göndermektedir. Bu bilgiye karşılık olarak masaüstü yazılımından Altu robota pist üzerinde gideceği hedef bilgisi aktarılmaktadır. Bulanık yaklaşımında robottan masaüstü yazılıma gelen koordinat ve mesafe verileri yeşil renkte, yazılım üzerinden robota giden rota bilgisi turuncu renkte gösterilmiştir. Geleneksel yöntemde robottan masaüstü yazılımına gelen koordinat ve mesafe verileri mavi renkte gösterilmiştir.

Sonuç olarak, Altu robot masaüstü yazılımı tarafından gönderilen hedef koordinatlarından gitmesi gereken konum bilgisini alıp, bulanık kontrolör ile gidilmesi gereken mesafeyi ve yönünü bulması sağlanmıştır.

Otonom robotların geliştirilmesi alanında yapılacak çalışmalara katkı sağlayacağı düşündüğüm bu tez çalışması, 6 bölümden oluşmaktadır.

Birinci bölümde, otonom robotlar hakkında genel bilgi verilmiştir ve tezin yapısı özetlenmiştir.

İkinci bölümde, otonom kontrol sistemleri ve robotlarda konum kontrolü hakkındaki konulara değinilmiştir.

Üçüncü bölümde, tez için tasarlanmış olan robotun gövde yapısı ve kullanılan malzemelerden bahsedilmiştir.

Dördüncü bölümde, sistem için kullanılan kontrolör ve sensörlerin kullanımı anlatılmıştır.

Beşinci bölümde, tasarlanan robotun son şeklinden bahsedilmiş ve pist içerisindeki otonom hareketleri grafik olarak verilmiştir.

Altıncı bölümde, robotun noktadan noktaya konum analizi yapılmış ve elde edilen test sonuçları ile sistemin başarısı ele alınmıştır.

Anahtar kelimeler: Bulanık Kontrolör, Konum Kontrol, Otonom Robot

ABSTRACT

FUZZY POSITION CONTROL APPROACH FOR AN AUTONOMOUS ROBOT CONTROLLER

CETINKAYA, Ali

M.Sc. - Electrical and Computer Engineering

Advisor: Prof. Dr. Novruz ALLAHVERDI

February 2017

In this thesis study, the application of the position control on the fixed track by the autonomous robot Fuzzy controller approach has been realized. For this purpose, as well as both hardware and software is designed as a system permitting real-time data exchange.

The embedded system (Atmel 2560 Integrity) is used in the hardware section to achieve the target attainment and position control with the Fuzzy controller.

DC motor, servo motor, ultrasonic sensor, bluetooth module and optical sensor are controlled through this system. In addition, microcontroller software and the Visual Studio 2015 platform using c # language designed desktop software. The coordinates (x, y) and route information sent by Altu robot on the designed runway in real time are processed with Fuzzy controller. The obtained data is displayed with Altu robot's desktop software, position information and robot movements map.

Applications using the Fuzzy approach were also tested with the conventional control method and the results were compared.

When Altu robot is run at any point on the track, it calculates the starting position and sends position information to the desktop software through the bluetooth module. In response to this information, the target information is transmitted on the Altu robot track from the desktop software. In the Fuzzy approach, the coordinate and distance data coming from the robot desktop software is shown in green color, while the route information from the software is shown in orange color. In the conventional method, coordinates and distance data coming from robot to desktop software are seen in blue color.

As a result, the Altu robot receives location information from the destination coordinates sent by the desktop software, and the Fuzzy controller is provided to find the distance and direction to go.

This thesis study which I think will contribute to the work to be done in the field of the development of autonomous robots is composed of 6 chapter.

In the first chapter, the general information about autonomous robots and thesis structure are summarized.

In the second chapter, autonomous control systems and position control in robots are discussed.

In the third chapter, the body of the robot designed for the thesis and the materials used are mentioned.

In the fourth chapter, the controllers and algorithms to be used for the system are explained.

In the fifth chapter, the autonomous movements of the designed robot and of the final shape are plotted graphically.

In the sixth chapter, the position of the robot is analyzed from the point to the point and the success of the system formed by the test results is considered.

Keywords: Fuzzy controller, Position Control, Autonomous Robot

TEŐEKKÜR

Çalıőma konusunun seimimde, tez alıőmasının planlanmasında ve deęerlendirilmesinde byk katkılarını grdęm btn aőamalarında deęerli bilgi ve deneyimleriyle bana yardımcı olan danıőman hocam sayın Prof. Dr. Novruz ALLAHVERDİ 'ye katkılarından dolayı teőekkr ederim.

Çalıőmalarım sırasında desteklerini esirgemeyen deęerli hocam Prof. Dr. Ali OKATAN'a, Öğr. Gör. Semih YUMUŐAK'a, Tuęba SARAY'a ve Aileme teőekkrlerimi sunuyorum.

ALİ ÇETİNKAYA

Őubat-2017

İÇİNDEKİLER

	Sayfa
TEZ BİLDİRİMİ	ii
ÖZET	iii
ABSTRACT	v
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
ÇİZELGELER LİSTESİ	x
ŞEKİLLER LİSTESİ	xi
KISALTMALAR	xiv
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	3
3. TASARLANAN ROBOTUN YAPISI	22
3.1. Altu Robotun Temel Parçaları	22
3.2. Robotun Gövdesi	24
3.3. Motorlar ve Tekerler	25
3.4. Kullanılan Güç Sistemi	25
3.5. Motor Sürücü Kartı	26
3.6. Mikrodenetleyici Kartı	27
3.7. Konumlandırma Sensörü	28
3.8. Kablosuz Haberleşme Birimi	28
3.9. QRD1114 Optik Sensör	31
3.10. 74HC14 Schmitt Trigger Entegresi ve Devresi	32
3.11. LM2576-5v ve 7805 Reğulatör Devreleri	33
4. ROBOTUN KONTROL ve YAZILIM YAPISI	34
4.1. Bulanık Kontrollör	34
4.1.1. Giriş ve Çıkış Değerlerinin Tanımlanması	35
4.1.2. Bulandırma	36
4.1.3. Üyelik İşlemleri	37
4.1.4. Bulanık Çıkarım	39
4.1.5. Bulanık Kuralların Belirlenmesi	39
4.1.6. Durulama	39
4.1.7. Matlab Fuzzy Toolbox ile Oluşturulan Sistemin Çıktısı	40

	Sayfa
4.2. Mikrodenetleyici Yazılımın Yapısı	41
4.2.1. Ultrasonik sensör	41
4.2.2. Kablosuz haberleşme sensörü	42
4.2.3. Hareket ünitesi	42
4.2.4. Kontrol algoritması	42
4.3. PC Yazılımın Yapısı	44
4.4. Tasarlanan Robot ve Çalışma Alanı	52
5. UYGULAMALAR	54
5.1. Birinci Uygulama	55
5.2. İkinci Uygulama	57
5.3. Üçüncü Uygulama	60
5.4. Dördüncü Uygulama	62
6. SONUÇ ve ÖNERİLER	65
EKLER	66
KAYNAKLAR	89
ÖZGEÇMİŞ	93

ÇİZELGELER LİSTESİ

Çizelge	Sayfa
Çizelge 1.1 Literatür Taraması	20
Çizelge 4.1 X Konumunun Alt Kümeleri	36
Çizelge 4.2 Y Konumunun Alt Kümeleri	36
Çizelge 4.3 Açık Verisinin Alt Kümeleri	36
Çizelge 4.4 Mesafe Verisinin Alt Kümeleri	37



ŞEKİLLER LİSTESİ

Şekil	Sayfa
Şekil 1.1 Otonom Robot Örnekleri, Tarım Robotu Akinsoft-AS-PNCR1(solda) İnsansız Kara Aracı Aselsan-Kaplan (sağda)	2
Şekil 3.1 Altu Robotu Oluşturan Gövde Parçası	23
Şekil 3.2 Altu Robotu Oluşturan Tutucu Parçalar, DC Motor Tutucu Parçası(solda) Servo Motor Kenar Tutucuları(ortada), Vidaları ve Somunları(ortada) Servo Motor Dönen Tutucu Parçası(sağda)	23
Şekil 3.3 Altu Robotun DC ve Servo Motorlarının Tutucularının Birleştirilmesi	23
Şekil 3.4 Altu Robot Üzerinde Kullanılan Kablolar(solda) 30mm Vida(sağda)	24
Şekil 3.5 Altu Robotun Alt Gövdesi	24
Şekil 3.6 Altu Robotun Alt Gövdesine Bulunan Servo ve DC Motorların Alt Gövdeye Sabitleme Aşaması(solda) Alt Gövdeye DC Motorların, Servo Motorların ve Tekerlerin Eklenmesi(sağda)	25
Şekil 3.7 Altu Robotta Kullanılan 11.1V Li-po pil(solda) 7.4V Li-po(sağda)	25
Şekil 3.8 Altu Robotun Gövdesi Üzerinde Bulunan Li-po Pillerin Konumu.	26
Şekil 3.9 Altu Robot İçin Tasarlanan L298 Entegreli Motor Sürücü Devresinin Üç Boyutlu Gösterimi	26
Şekil 3.10 Altu Robotun Alt Gövdesinde Bulunan Motor Sürücü Kartı	27
Şekil 3.11 Altu Robotun Mikrodenetleyici Kartı.	27
Şekil 3.12 HC-SR04 Ultrasonik Sensör	28
Şekil 3.13 Altu Robotun Kablosuz HC05 Bluetooth Modülü(solda) PC USB Bluetooth Adaptörü(sağda)	28
Şekil 3.14 PC USB Modülün Tanıtılması	29
Şekil 3.15 PC USB Modül Üzerinde HC05 Bluetooth Modülün Görünmesi	29
Şekil 3.16 PC USB Modül ile HC05 Modül Eşleştirilmesi	30
Şekil 3.17 HC05 Modülün Eşleşme Kodunun Girilmesi	30
Şekil 3.18 HC05 Modül ile PC USB Modül Arasında Oluşturulan Com Port	31
Şekil 3.19 HC05 Modülün Bilgisayar Üzerinde Başarılı Şekilde Eklenmesi	31
Şekil 3.20 QRD1114 Optik Sensör	31
Şekil 3.21 QRD1114 Bağlantı Şeması	32
Şekil 3.22 74HC14'ün Genel Görüntüsü(solda) 74HC14 İle QRD1114'ün Birleştirilmesiyle Oluşturulan Tasarımın Üç Boyutlu Görüntüsü(sağda)	32
Şekil 3.23 74HC14 ile Tasarlanmış Devrenin Görüntüsü	33
Şekil 3.24 Qrd1114 Optik Sensörün Çalışması.	33
Şekil 3.25 LM2576-5V Regülatör Devresi(solda) LM2576-5V Regülatör Devresinin PCB Yüzeyi(ortada) 7505 5V Regülatör Devresi (sağda)	33

Şekil	Sayfa
Şekil 4.1 C# Yazılımının Altu Robotun Hareket Alanı.	35
Şekil 4.2 X Konum Verisinin Bulanık Alt Kümeleri	37
Şekil 4.3 Y Konum Verisinin Bulanık Alt Kümeleri	38
Şekil 4.4 Açık Verisinin Bulanık Alt Kümeleri	38
Şekil 4.5 Mesafe Verisinin Bulanık Alt Kümeleri	38
Şekil 4.6 Ağırlık Merkezi Yöntemi ile Durulaştırma	40
Şekil 4.7 Oluşturulan Sistemin Matlab Çıktısı	41
Şekil 4.8 Bulanık Kontrol Blok Diyagramı	43
Şekil 4.9 Geleneksel Kontrol Blok Diyagramı	43
Şekil 4.10 Mesafe Hesaplama Diyagramı	44
Şekil 4.11 C# Yazılımında Oluşturulan Yazılımının Konum Alma Kısmı	44
Şekil 4.12 Oluşturulan Yazılımının Mouse ile Taranan Konumlar	45
Şekil 4.13 Yazılımda Mouse ile Oluşturulan Net Konumlar	46
Şekil 4.14 Yazılımda Oluşturulan Pistin Özellikleri ve Aktif Eksen Değerleri	46
Şekil 4.15 Kullanılan Sistemin Masaüstü Ayarları	47
Şekil 4.16 C# Yazılımında Oluşturulan Haritalama Kısmı	48
Şekil 4.17 C# Yazılımında Oluşturulan Altu Robot Bağlantı Ayarları	48
Şekil 4.19 Altu Robotun Pist Üzerindeki Konumu	49
Şekil 4.20 Altu Robotun Pist Üzerindeki Konumu ve Hedef Noktasının İşareti	50
Şekil 4.21 Geleneksel Kontrol Yöntemi	51
Şekil 4.22 Geleneksel Kontrol Yöntemi ile Altu Robotun Pist Üzerinde Konumu	51
Şekil 4.23 Altu Robotun Son Görünümü	52
Şekil 4.24 Altu Robotun Çalışma Alanı	53
Şekil 4.25 Altu Robotun pist içerisinde görünümü	53
Şekil 5.1 Birinci Uygulamanın Senaryo Haritası	55
Şekil 5.2 Açıklanan Birinci Uygulamanın Sonuç Haritası	56
Şekil 5.3 Açıklanan Birinci Uygulamanın Matlab Sonuç Haritası	56
Şekil 5.4 İkinci Uygulamanın Senaryo Haritası	57
Şekil 5.5 Açıklanan İkinci Uygulamanın Sonuç Haritası	58
Şekil 5.6 Açıklanan İkinci Uygulamanın Matlab Sonuç Haritası	59
Şekil 5.7 Üçüncü Uygulamanın Senaryo Haritası	60
Şekil 5.8 Açıklanan Üçüncü Uygulamanın Sonuç Haritası	61
Şekil 5.9 Açıklanan Üçüncü Uygulamanın Matlab Sonuç Haritası	61

Şekil	Sayfa
Şekil 5.10 Dördüncü Uygulamanın Senaryo Haritası	62
Şekil 5.11 Açıklanan Dördüncü Uygulamanın Sonuç Haritası	63
Şekil 5.12 Açıklanan Dördüncü Uygulamanın Matlab Sonuç Haritası	64



KISALTMALAR

Kısaltma	Açıklaması
ALTU	Tez İçin Tasarlanan Otonom Robotun İsmi
PC	Personal Computer (Kişisel Bilgisayar)
RPM	Dakikada ki Devir Sayısı
GUI	Kullanıcı Arayüzü
C#	Visual Studio 2015 Yazılımının GUI Arayüzü Bölümü
MATLAB	Sayısal Hesaplama Yazılımı ve Dördüncü Nesil Yazılım Dili
PID	Proportioanal Integral Derivative
PWM	Dalga Genişlik Modülasyonu
FUZZY LOGIC	Bulanık Mantık
PSEUDUCODE	Kodlamadan Önce Plan Yapılması
AND	Dijital Ve İşlemi
UNSIGNED	Değişkenlerin İşaretsiz Hali
MILLIS	Mikroişlemcinin Çalışma Zamanını Milisaniye Cinsinden Tutar
DOUBLE	64 Bitlik Değerleri Tanımlar.
ERROR	Hata
IF	Sorgu Yapısı. Eğer Anlamı İçerir.
HC04	Ultrasonik Sensörün İsmi
HC05	Bluetooth Modülün İsmi
STRAFOR	Sert Sentetik Reçinedir.
CM	Santimetre
MM	Milimetre
MSN	Milisaniye
X,Y,Z	Yeryüzü Referans Eksenleri
Θ	Robotun Doğrultu Vektörü ile Yer İmi Doğrultusu Arasındaki Açıdır.
PCB	Baskı Devre
RS232	Seri İletişim Protokolü
DC	Doğru Akım
LİPO	Lityum Polimer
UART	Universal Asynchronous Receiver and Transmitter (Evrensel Asenkron Alıcı ve Verici)
Wi-Fi	Kablosuz Ağ Bağlantısı

1. GİRİŞ

Otonom robotların kullanım alanları incelendiğinde askeri uygulamalardan, polis bomba imha robotlarına, tarım robotlarından, uzay keşif robotları ve taşıma robotlarına kadar çok geniş uygulama alanına sahip olduğu görülmektedir. Kullanım alanlarının farklılaşması robotlarında hem yazılım hem de donanım olarak birçok noktada ar-ge çalışması gerektiren sistemler haline getirmektedir.

Otonom robotlar, sensörleri sayesinde çevreden edindiği bilgilere göre hareket eden ve verilen görevleri kendi kendine yerine getiren robotlardır. Bu görevleri yerine getirirken algıladıkları çevreye göre önceden belirlenen hedefe gitmek için kendi yönlerini bulabilir ve karşısına çıkabilecek olan engelleri aşarak belirtilen konuma en az hata payı ile ilerleyebilirler.

Bu araçlar çevrelerini tanımlarken; ultrasonik sensör, GPS (Global Positioning System), radar, lazer, kamera ve bunlara benzeyen çeşitli sensörlerden yardım alırlar. Bu algılayıcılardan gelen verileri kendi içerisindeki kontrol yazılımı ile sentezleyip kendisi için en iyi rotayı belirler ve çalışmaya başlarlar. Bu süreçte gidilen rota boyunca engelleri hesaba katarak gitmesi gereken koordinat arasındaki farkı kendi içinde güncelleyip normal rotasında seyir etmektedirler.

Birçok çalışmada kullanılan bu robotlar geçmişten günümüze robot biliminin ne kadar geliştiğini, teknolojik olarak yapılabilecek işlemlerin sınırlarının olmadığını göstermektedir. Otonom araçların yeni yöntemler kullanılarak geliştirilmesi insandan kaynaklanan hata payını ortadan kaldırabilecek hassasiyete getirilebilmesini sağlayabilmektedir. Şekil 1.1’de yukarıda bahsi geçen robot örnekleri verilmiştir.



Şekil 1.1 Otonom Robot Örnekleri, Tarım Robotu Akınsoft-AS-PNCR1(solda)
İnsansız Kara Aracı Aselsan-Kaplan (sağda)

Otonom bir kontrol sistemi; sistemin güvenilirliği, kararlılığı, tepkilere göre değiştirilebilir bir yapıda olması ve sistemin sonuçlarını kendi içerisinde sentezleyip çıktılarını kolay ve anlaşılabilir şekilde kullanıcıya bildirmesi gerekmektedir. Özellikle tasarlanacak robotun teknik özellikleri istenilen kontrol yöntemini karşılaması ve döngü içerisinde görevlerinin yerine getirmesi gerekmektedir. Ayrıca uygun ve etkin bir kontrol sağlamak için sistem parametrelerinin zamanla değişmesi istenmektedir. Bu yüzden insan düşünme yeteneği ve bilgisini kontrol sisteminde örnekleyebilecek bir kontrol yöntemi olan bulanık mantık denetleyicisi iyi bir çözüm olabilmektedir. Robotun kontrol hareketleri kapalı döngü metodu ile sağlanmalı, konum kontrolü içerisinde bulunan belirsizlikler için bulanık mantık kullanılmalıdır. Otonom robot sisteminin bulanık mantık ve Atmega2560 (Arduino MEGA) ile kontrol edilmesi üzerine az sayıda çalışma yapılmıştır.

2. LİTERATÜR ARAŞTIRMASI

Otonom robotların hedeflenen konumu bulabilmeleri için en az iki farklı bilgiye ihtiyaçları vardır. Bunlar; gidilecek mesafe ve robot ile gidilecek konumun arasındaki açıdır. Robotlar önceden ortam harita bilgisine sahip ise kullandığı sensörler yardımıyla hedefine ulaşabilir. Eğer ortam bilgisine sahip değil ise kullanıcıdan aldığı veriler doğrultusunda işlemlerini gerçekleştirir.

Otonom robotlara verilen görevleri yerine getirmesi konusunda en önemli unsurlardan birisi bulunduğu konumlarını bilmeleri ve buldukları ortamın haritasını çıkarmalarıdır. Eş zamanlı konumlandırma ve harita çıkarma adı verilen bu süreçte robotların daha önceden bilmediği bir ortamda bulunduğu otonom olarak hareket edebilmesidir [1].

Otonom gezgin robotlarla ilgili literatürde pek çok çalışma yapılmıştır. Bu tezin amacı, hazırlanmış olan Altu (Tez için tasarlanan otonom robot) robot ile önceden oluşturulan pist üzerinde ultrasonik sensör kullanarak harita çıkarmaya gerek kalmadan masaüstü yazılımı üzerinden robota gelecek olan mesafe ve açı bilgilerini bulanık mantık ile işleyip hareketini gerçekleştirmektedir.

TAŞDEMİR [1] tarafından gerçekleştirilen çalışmada, çoklu mobil robot devriye sistemlerinde birden fazla algılayıcıdan elde edilen verilerdeki hata oranını nasıl ve ne kadar iyileştirdiği benzetim çalışmalarında gözlemlenmiştir. Benzetimi yapılan algılayıcı, bulunduğu ortamdaki verileri lazer, ultrasonik, kamera algılayıcıları kullanılarak ortamdaki toplamış, bu algılayıcıların değişik kombinasyonları ele alınarak veri değerlerinin nasıl değiştiği gözlemlenmiştir. Kullanılan lazer ve kamera algılayıcıları ana algılayıcı olarak belirlenmiştir. Benzetim ortamı olarak Microsoft Robotics Development Studio kullanılmış, ayrıca OpenCV (Open Source Computer Vision) kütüphanelerinden görüntü işleme yararlanılmıştır. Aynı zamanda mobil robotlar arasındaki veri iletişimi benzetim ortamında modellenerek incelenmiştir. Benzetim çalışmalarında Prowler kablosuz ağ benzetim paketi kullanılmıştır.

TUNCER [2] tarafından gerçekleştirilen çalışmada, otonom bir gezgin robot için Genetik Algoritmalar (GA) ile yol bulma problemine ilişkin yeni bir mutasyon operatörü geliştirilmiş ve bu yeni mutasyon operatörü ile daha önce literatürde sunulmuş olan farklı mutasyon operatörleri karşılaştırılmıştır. Karşılaştırmalar sonucunda yeni mutasyon operatörünün daha başarılı performans sergilediği sonucuna varılmıştır. GA ile yol bulma algoritması Pioneer 3-DX gezgin robot üzerinde denenmiştir. Robot ile haberleşmede Matlab programı kullanılmış ve görüntü almak amacıyla, laboratuvar tavanına bir kamera yerleştirilerek konumlandırması gerçekleştirilmiştir. Robotun ortamdaki engelleri ve kendi konumunu belirlemesi sağlanarak konumlandırma ve yol bulma işlemleri bütünlük bir sistem olarak gerçek laboratuvar ortamında denenmiştir.

ÖZEREN [3] tarafından gerçekleştirilen çalışmada, taşıma işleminde kullanılacak, otonom olarak hareket edebilen, verilen adrese en kısa yoldan giden, etrafındaki cisimleri algılayan, bir robot geliştirilmiştir. Robotun bulunduğu binanın haritasını çıkarması, konumunu algılayabilmesi için lazermetre, jireskop, ivmeölçer, manyetik sensör ve enkoder kullanılmıştır. Sensörden okunan bilgileri alıp haritalamayı yapan, konumu hesaplayan, en kısa yolu bulan sistemde, hareket bilgilerini oluşturmak için geliştirilen yazılım sabit bir bilgisayarda tutulmaktadır. Robot ile bilgisayarın haberleşmesi bluetooth bağlantısı ile sağlanmıştır. Robotun çalıştırılacağı ortamda, ilk olarak çevrenin öğretilmesi gerekmektedir. Bu amaçla 360 derece dönebilen lazer metre ve jireskop kullanılarak, çok sayıda noktadan tarama yapıp, ortamın haritası çıkarılmaktadır. Tarama esnasında lazer metre ile belli bir yükseklikten, duvarlar ve sabit cisimlerin mesafeleri, ölçüm noktasına bağlı olarak veri tabanına kaydedilmektedir. Ortamın öğretilmesinde sonra görev alanı içerisinde robot çalıştırıldığında, etrafındaki cisimlerin uzaklıklarını tespit ederek ortamın haritasını oluşturmada ve hafızasındaki haritalarla karşılaştırılarak konumu hesaplamaktadır. Başlangıç konumunu hesaplayan robot, verilen hedef noktaya giderken sürekli olarak etrafını taramakta ve tekerlerden aldığı enkoder bilgilerini de kullanarak hedef noktaya gitmektedir. Hedefe giderken hafızasındaki ölçüm noktalarını kullanarak karınca koloni algoritması ile en kısa yolu hesaplayarak gitmeye çalışmaktadır.

LAFCI [4] tarafından gerçekleştirilen çalışmada, sabit ve hareketli engellerin bulunduğu ortamda, bir robotun engellere çarpmadan istenilen hedef bölgeye ilerleyebilmesi alanında uygulamalar yapılmıştır. İnsanların giremediği bölgelerde, robotların rahatlıkla dolaşarak herhangi bir görevi yerine getirmesi, o bölgede engellere çarpmadan istenilen yerlere ulaşabilmesi ile mümkün olacaktır. Robotun hareket ettiği ortam sabit engelli ve sabit bir bölge olabileceği gibi hızlı değişen dinamik engelli dinamik bir bölge de olabilir. Gezgin robotlar engellerden kaçınarak hedefine giderken optimum olmayan yolları izleyebilir. Hareket planlama algoritmalarında sabit veya hareketli engellerin bulunduğu belirli bir arazide, gezgin robotların engellere ve birbirine çarpmadan otonom hareket etmeleri amaçlanmaktadır. Bu kapsamda; algoritma çalışmalarını gerçekleştirebilmek için, pratik alanda yapılacak test maliyetlerini azaltmak üzere, bir simülasyon ortamında çalışmalar yapılmıştır. “A* Arama algoritması”; en kısa yolu bulmak için kullanılan algoritmalarından birisidir. Robot bilinen bir ortamda hedef olarak gösterilen noktaya optimum yoldan giderken karşısına çıkacak dinamik engellerden kaçarak yoluna devam etmesi gerçekleştirilmiştir.

YUFKA [5] tarafından gerçekleştirilen çalışmada, holonomik olmayan (bir yere bağlı olan) bir grup robotun tek bir nesneyi koordineli olarak taşıyabilmesi için hareket planlama ve kontrol yaklaşımı önerilmiştir. Koordineli taşıma sistemi için sanal lider-takipçi formasyon kontrolü yöntemi kullanılmıştır. Taşınan nesne, koordineli taşıma sisteminin sanal lider robotu olarak düşünülmüş ve nesneyi taşıyan robotlarda takipçi robotları olarak ele alınmıştır. Kinematik kısıtlar göz önünde bulundurularak sanal robotun izleyeceği bir yörünge planlanmış ve nesnenin ağırlık merkezi noktasına yerleştirilen koordinat sisteminin merkezinin yörüngeyi takip etmesi sağlanmıştır. Her bir takipçi robotun yörüngesi formasyon yapısını koruyacak şekilde planlanarak, robotların kendi yörüngelerini izlemeleri sağlanmıştır. Takipçi robotların başlangıçta eş zamanlama yapabilmesi için bir iletim birimi kullanılmış ve yaratılan yörüngeleri izleyebilmek için asimptotik kararlı yörünge izleme denetleyicisi kullanılmıştır. Pioneer P3-DX gezgin robotlar kullanılarak gerçek dünya uygulamaları ve benzetimlerle çalışılmıştır.

DİLMEN [6] tarafından gerçekleştirilen çalışmada, gezgin robotlarda dış ortamda konumlandırma amaçlı GPS kullanımı ile ilgili çalışmalar incelenmiştir. Öncelikle, GPS tarihçesi ve genel özellikleri araştırılmıştır. Tez çalışmasında kullanılacak bir GPS alıcı devresi yapılmıştır. Alıcı devresinde U-box Neo-4S GPS modülü kullanılmış ve çeşitli ortamlarda hata karakteristiği çıkarılmıştır. Bu karakteristiğin iyileştirme yöntemleri araştırılmış ve bu kapsamda DGPS incelenmiştir. Türkiye’de bu hizmeti veren TSSAGA sisteminden faydalanılmıştır.

ÖZALP [7] tarafından gerçekleştirilen çalışmada, tekli ve çoklu İHA’lar (İnsansız Hava Araçları) için tüm koşullarda geçerli olabilecek güvenli rota planlama konusu üzerine durulmuştur. GA kullanılarak İHA’lara ait kinematik kısıtları ve arazi şartlarını göz önünde bulundurarak en uygun rotanın planlanması üzerinde çalışılmıştır. Kullanılan arazi NASA’ya ait gerçek 3 boyutlu uydu görüntülerinden oluşmaktadır. Bu sayede yükseklik verisine erişim sağlanmış ve böylece arazi üzerindeki dağların tespiti yapılmıştır. Gerçek coğrafik koordinat sistemi kullanılarak ve dünyanın geometrik şekli göz önünde bulundurularak yüksek hassasiyetli hesaplamalar yapılmıştır. GA ile global rota planlaması yapılırken noktalar arasındaki geçişlerde yerel rota planlaması üzerinde durulmuştur. Uçağın kinematik özellikleri (maksimum yükselme açısı, dönme açısı vb.) göz önünde bulundurularak aradaki engellere çarpmadan ve tespit edilen tehditli bölgelere girmeden yerel rota planlama yapılmıştır. Çoklu rota planlama GA’nın çoklu kromozom yapısı sayesinde gerçekleşmiştir. Çoklu kromozom yapısı içerisinde her bir kromozomun ilk geni pist alanını temsil etmektedir. Böylece her bir İHA için maksimum ve minimum gezmesi gereken nokta sayısı ve dolaşma mesafesi gibi kısıtlar göz önünde bulundurularak çoklu İHA’lar için rota planlaması yapılmıştır.

NİKBAY [8] tarafından gerçekleştirilen çalışmada, bir harita üzerinde otonom araca verilen başlangıç noktasından, tanımlanan hedef noktasına derinlik öncelikli arama algoritması ile bir güzergâh oluşturularak varması hedeflenmiştir. Sürücülü ve sürücüsüz araçların birlikte yer aldığı bir deneme oyun platformu geliştirilmiştir. Bu platformda izleme noktaları tek tek menü yardımıyla tanımlanarak güzergâh takibi yapılmış, ancak bunun kullanılabilirliği zor olduğundan daha soyut tanımlardan güzergâh takibinin gerçekleştirilmesi araştırılmıştır.

Yol tanımlama ve hedef güzergâh yapıları araştırılarak seçilen yapı üzerinde bir graf oluşturulmuş ve böylece graf üzerinde bilinen yol bulma algoritmaları denenmiştir. 2B bir sürüş simülasyonu gerçekleştirilmiş ve örnek senaryolar incelenmiştir. Son olarak ise uygulama sonrası elde edilen sonuçlar ve değerlendirmeler özetlenmiştir.

AKÇAKAYA [9] tarafından gerçekleştirilen çalışmada, otonom sualtı araçlarının tasarımı ve kontrolü amaçlanmıştır. Sualtı aracı tasarımı; mekanik tasarım, dinamik model çıkarımı ve kontrolör tasarımı gibi birçok alt tasarım bileşenini içermektedir. Otonom sualtı araçlarına takılacak robot kolları ile keşif, tamir, sualtından cisim toplama ve ölçüm, dalgıca yardım etme gibi birçok görevi yerine getirebilir. Sualtı aracına takılacak bir sonar ile yörünge düzeltme, deniz dibi görüntüleme, obje tanıma, mayın tanımlama, hedef takibi gibi görevleri gerçekleştirmek ve takılacak bir kamera ile düşük derinliklerde yörünge takibi, çarpışma kontrolü, hedefi takip, deniz dibi haritalama görevleri yapılabilir.

UYAR [10] tarafından gerçekleştirilen çalışmada, asenkron motorları altı farklı hata parametresine karşı koruyan elektronik koruma sistemi gömülü sistem olarak gerçekleştirilmiştir. Bu parametreler; aşırı gerilim, aşırı akım, sargı sıcaklığı, gerilim dengesizliği, akım dengesizliği ve düşük gerilimdir. Sistem hata algılandığında belirli bir süre bekleyerek bu süre içinde hata düzelmemiş ise motorun çalışması durdurulmaktadır. Bu süreyi mikrodenetleyici içine yazılmış bulanık mantık programı hesaplamaktadır. Böylece birçok hata kombinasyonu için farklı süreler üretilmiş ve kontrol sisteminin hassasiyeti arttırılmıştır. Kontrol sistemi bir hata algılandığında sesli uyarı vererek hatanın nereden kaynaklandığını grafik ekranda göstermektedir. Bu ekran aynı zamanda hata oluştuğu andaki akım, gerilim veya sıcaklık değerlerini de göstermektedir. Ayrıca okunan tüm verileri RS232 kablosu ile bilgisayara aktarılabilir.

ERENOĞLU [11] tarafından gerçekleştirilen çalışmada, klasik PID (Proportioanal Integral Derivative) ve bulanık kontrolün akıllı bir yöntemle harmanlanmasından meydana gelen akıllı melez bulanık PID kontrolör yapısı anlatılmıştır.

Temel olarak bu tasarım metodunda, sistemde oluşan hatayla ilişkili bir fonksiyona bağılı harmanlama mekanizması yardımıyla klasik PID kontrolör ve bulanık kontrolör birleştirilmektedir. Buna ek olarak, akıllı bir anahtarlama mekanizması, harmanlama mekanizmasının hangi kontrolörün (Klasik PID veya Bulanık Kontrolör) kontrol eyleminde öncelikli olarak baskın kullanılacağına kararını almaktadır. Yeni akıllı melez kontrolör yapısı ile çeşitli sistemler için simülasyon yapılmakta ve değerlendirmeler sonucu hem geçici hem de kararlı hal cevabında her iki kontrolöründen de (klasik PID ve Bulanık kontrolör) daha iyi performans sergilediği görülmektedir. Kontrolör parametreleri genetik algoritmalar yardımıyla belirlenmektedir.

CİVELEK [12] tarafından gerçekleştirilen çalışmada, bir rüzgâr türbininin kanat hatve açısı kontrolörünün PID katsayılarının bulanık mantık algoritması ile ayarlanması üzerine bir inceleme yapılmıştır. Klasik PI, bulanık kontrol ve bulanık PID kontrol olmak üzere üç ayrı kontrol yöntemi kullanılarak, yüksek rüzgâr hızlarında sistemin zarar görmemesi ve nominal çıkış gücünün korunması amaçlanmıştır. Matlab/Simulink programı ile kontrolörlerin simülasyonu yapılarak, değişik rüzgâr hızlarında rüzgâr hızlarında rüzgâr türbini kanat hatve açısının kontrolü ve çekiş gücünün ayar noktasında sabit tutulması hedeflenmiştir. Simülasyon sonuçlarından elde edilen çıkış gücünün kararlı hale gelme zamanı ve kararlı hal hataları değerlendirilerek, kontrol sistemlerinin performansları ölçülmüş ve birbirleriyle karşılaştırılmıştır. Tespit edilen sonuçlar içerisinde Bulanık PID kontrolörün, PI ve Bulanık kontrolörlere göre daha iyi performans sergilediği bulunmuştur.

GÜZAY [13] tarafından gerçekleştirilen çalışmada, akıllı kontrol sistemlerini, geleneksel kontrol yapıları ile birleştirmenin mümkün olduğu ve bu alandaki çalışmalar gerek teorik olarak gerekse uygulamalı olarak bu birleşimin iki farklı kontrol stratejisinin avantajlarını bir araya getireceği gösterilmiştir. Bu tip kontrolörlerde tasarım bir noktaya göre yapılabileceği gibi birden fazla noktaya göre yapılarak serbestlik derecesi artırılabilir. Bulanık kontrolörlerin tasarım aşaması, yapısal ve ayarlama parametreleri olmak üzere iki ana gruba ayrılabilir.

Yapısal parametreler giriş/çıkış birimlerini, dilsel değişkenlerle tanımlanmış üyelik fonksiyonlarını, kuralları, bulanık çıkarım ve durulaştırma mekanizmalarını içermektedir. Ayarlama parametreleri ise giriş/çıkış ölçekleme çarpanlarından ve üyelik fonksiyonu parametrelerinden oluşmaktadır. Tasarım, bu noktalardan biri ile yapılarak diğerleri bir reçeteye göre ayarlanabileceği gibi birden fazla noktaya göre de yapılabilir. Tasarım aşamasına benzer şekilde bulanık kontrolörlerin etkinliklerini artırılması eniyileme ya da özayarlama gibi yöntemlerle mümkündür. Özayarlama, bulanık kontrolörün serbest bırakılan bir tasarım noktasının kapalı çevrim sistemin çıkışı, hata işareti, kontrol işareti vb. ile bunların türevlerini, integrallerini girdi olarak alabilen doğrusal ya da doğrusal olmayan bir fonksiyona bağlı olmasıdır. Ayarlama parametrelerinde farklı olarak yapısal parametreler üzerinde ayarlama yapmak mümkündür. Bu tez çalışması, tamamen yenilikçi bir fikir olarak bulanık PID kontrolörleri için çıkarım mekanizmasındaki birleştirme operatörünün çevrimiçi biçimde öz-ayarlamasını içermektedir. Böylece, yapısal parametrelerin de ayarlama parametreleri gibi kullanılabilmesi gösterilerek, kontrolörün verimliliği artırılması hedeflenmiştir. Bu çalışmada birleştirme operatörü olarak bir serbest parametreye sahip Gama operatörü kullanılmaktadır.

KABASAKAL [14] tarafından gerçekleştirilen çalışmada, bulanık PID kontrolörüne ait bulanık kural ağırlıkları için ayarlama yöntemlerinin geliştirilmesi amaçlanmıştır. Bu doğrultuda iki ayrı bulanık kural ağırlıklandırma yöntemi önerilmiştir. Bu kural ağırlıklandırma yöntemlerinde sistem bilgisi kullanılmış ve yöntemler çevrimiçi olarak ele alınmıştır. Önerilen yöntemler genel bir yapıya sahip olduklarından, her türlü sistem tipine uygulanabilmektedir. Söz konusu kural ağırlıklandırma yöntemlerinin ilkinde sadece sisteme ait hata bilgisi kullanılmıştır. İkincisinde hata bilgisine ek olarak sistem hız bilgisi sağlayan birimselleştirilmiş ivme değişkeni de kullanılmıştır. Önerilen yöntemler ilk olarak Matlab ortamında bazı doğrusal olan ve olmayan test sistemlerine benzetimler yoluyla uygulanmış ve irdelenmiştir. Benzetim ortamında tasarlanan bir kontrolör yapısının, fiziksel sistemler üzerinde uygulanması neticesinde başarımda birtakım farklılıkların ortaya çıkabildiği bilinen bir gerçektir.

Bu kapsamda, benzetim uygulamalarının yanı sıra, önerilen bulanık PID kontrolör yapıları Proses Kontrol Simülatorü ve pH Kontrol Deney Seri üzerinde de gerçek zamanda uygulanmıştır. Elde edilen sistem yanıtları ve başarımları irdelenmiş ve değişik başarımları ölçütleri doğrultusunda karşılaştırılmıştır.

BULUT [15] tarafından gerçekleştirilen çalışmada, paralel olarak kullanılan Bulanık PID Kontrolör ve Klasik PID kontrolörün cevapları karşılaştırılıp özel yöntemlerle birleştirilerek oluşturulan yeni bir kontrol işareti üreten kontrol yapıları tanıtılmaktadır. İki kontrol işaretini belirli oranlarda karşılaştırarak harmanlayıp yeni bir işaret oluşturan mekanizma ‘Melez Kontrolör’ olarak adlandırılmıştır. Kısaca melez kontrolör tasarım fikri, yapısında bulunan kontrolörlerin avantajlarını birleştirmekten gelmektedir. Böylece lineer hem de lineer olmayan sistemler için uygun bir kontrol metodolojisi geliştirilecektir. Yapılan benzetimler ve gerçek zaman uygulaması sonrasında Melez kontrolör sistem cevabı ve kontrolör çıkışları karşılaştırılmış ve yapılan değerlendirmeler sonucu hem kararlı hal cevabı hem geçici hal cevabında hem de bozucular altında, yapısındaki diğer iki kontrolörden de daha iyi performans sergilediği görülmüştür.

SAĞLAM [16] tarafından gerçekleştirilen çalışmada, bulanık mantığın temel kontrolörlerinden olan bulanık PID kontrolörüne hiyerarşik özelliğın kazandırılması ele alınmıştır. İki girişli ve üç girişli PID kontrolör yapılarının yanı sıra üç girişli PID kontrolörden faydalanarak oluşturulan hiyerarşik PID kontrolör, temel üç doğrusal sistem ve bir doğrusal olmayan sistem üzerinde uygulanmıştır. Ele alınan sistemler üzerinde hangi kontrolör yapısının etkili sonuçlar verdiğine karar verilmeye çalışılmıştır.

ŞAHİN [17] tarafından gerçekleştirilen çalışmada, “Kendinden Ayarlamalı PID Kontrolör ile İki Eksenli Gimbal Uygulaması” kapsamında uygulamalar gerçekleştirilmiştir. Gimbal, füzelerde arayıcı başlığın hedefe kilitlenmesini ve takibini sağlayan ve üç ekseninde hareket kabiliyeti ile görüş açısını artıran bir sistemdir. Füzenin hatasız bir şekilde hedefi takip etmesi için, Gimbal’ın füzenin dinamik gövde hareketinden bağımsız olarak stabilize edilmesi gerekmektedir.

Hızlı ve kesin bir şekilde hedefe kilitlenme ve bozucu etkenlere karşı takibini sürdürme füzenin başarılı olması kapsamında çok kritik bir unsurdur. Endüstride, hızlı ve etkin kontrol kapsamında, uzun yıllardır geleneksel PID kontrolcüler kullanılmaktadır. Fakat bu kontrolcüler Gimbal gibi içerisinde belirsizliklerin bulunduğu lineer olmayan uzay teknolojilerinde başarılı olmamaktadır. Sonuç olarak, lineer olmayan Gimbal sistemi kapsamında, katsayıları sürekli güncellenen kendinden ayarlamalı PID kontrolcüsü geliştirilmiştir.

GÜLLÜ [18] tarafından gerçekleştirilen çalışmada, en çok kullanılan kontrol algoritmalarından olan PID kavramı anlatılmakta ve hazırlanan arayüz ile PID uygulamalarının simülasyonu yapılmaktadır. PID kontrol yapısındaki P, PI, PD ve PID gibi ayrı ayrı sisteme uygulanabilmekte ve bu uygulamaların sonucu, sistemin kapalı, açık çevrim, zaman ve frekans cevap eğrilerinde izlenebilmektedir. PID kavramının yanı sıra kontrol, sistem ve kararlılık kavramlarının önemi, teorik ve uygulamalı olarak sunulmuştur.

DURMUŞ [19] tarafından gerçekleştirilen çalışmada, çeşitli alanlarda kullanılmak üzere (öncelikli olarak tarım alanında) otonom robot ve kontrol ünitesi tasarlanmış ve gerçekleştirilmiştir. Proje tasarımı yapılırken modülerlik, düşük maliyet, yerli üretime katkı, çevre şartlarına uygunluk ve özgünlük özelliklerine sahip bir robot tasarlanmaya çalışılmıştır. Robotun kontrol ünitesinin robotun taşıyıcı platformundan bağımsız olmasına dikkat edilmiştir. Çünkü projede kullanılan platformun amaca ve kullanılacağı araziye uygun olarak değiştirilebilmesi istenmektedir. Sonuç olarak, çeşitli amaçlara uygun olan bir otonom mobil robot ve kontrol ünitesi tasarlanmış ve gerçekleştirilmiştir.

AKSOY ve ark. [20] tarafından gerçekleştirilen çalışmada, robot ve robotik teknolojilerinin gelişimi süreci, terminolojisi ve kategorileri incelenerek insansız kara araçlarına ait muhabere gereksinimleri ortaya koyulmuştur. Geleceğin muhabere sistemlerinde önemli bir yeri olan insansız kara aracı teknolojileri için askeri konsept ve doktrinlerin öngördüğü mühendislik geliştirme ihtiyaçları genel bir değerlendirmeye tabi tutulmuş ve çok kritik olan bu sistemlerin günümüzün asimetrik muharebe koşullarındaki önemi ve avantajları tespit edilmeye çalışılmıştır.

Yapılan deęerlendirmede aę merkezli konseptlerin önemli bir bileşeni olan otonom sistemler modüler yapıda, müşterek mimaride ve işlevsel beklentiler paralelinde güçlü bir mühendislik çalışmasıyla projelendirilip üretildięi takdirde hem ülke ekonomisine önemli katkı sağlayacak hem de muharebede üstünlüğün elde bulundurulmasında kuvvet çarpanı olacağı sonucuna varılmıştır.

YÜKSEL ve ark. [21] tarafından yapılan çalışmada, bir mobil robotun hedef noktaya erişimi ve toplanan verilerin RF (Radyo Frekansı) ile transferi işleminde, çalışan bölgenin şartlarına baęlı olarak, insan saęlığı açısından tehlike arz edebilecek durumlarda mobil robotların kullanımı artmıştır. Bu amaçta mobil robotu hedefe ulaştıracak algoritma büyük önem taşımaktadır. Bu kapsamda hedefe ulaşmak için duvar takip ve böcek algoritması kullanılmıştır. Bu araştırmanın neticesinde istenen bir konumdan insansız olarak her türlü tehlikeden uzak durularak bir ölçümün yapılabilmesi gerçekleştirilmiştir.

ÖZTÜRK ve ark. [22] tarafından yapılan çalışmada, havaalanlarındaki uçuş güvenliğini tehlikeye atan yabancı nesnelerin temizliği için çoklu-robot koordinasyonu yaklaşımın üzerinde durmuşlardır. Havaalanı pistlerinde ve çevresinde yer alan yabancı nesneler uçuş güvenliğini riske atmakta ve kazalara sebep olmaktadır. Bu nesnelerin tespiti için birkaç milimetre boyutundaki nesnelere kilometrelerce öteden tespit eden radarlar ve yüksek çözünürlüğe ve yüksek büyütme oranına sahip kameralar gibi ileri teknolojiler kullanılmaktadır. Ancak tespit edilen nesnelerin toplanmasında insan faktörü devreye girmektedir. Nesnelerin insan gücü ile toplanması ise hem güvenlik açısından hem de zaman alıcı bir yöntem olması açısından problem teşkil etmektedir. Bu çalışmada, havaalanı pistlerindeki yabancı nesnelerin toplanması amacıyla otonom çoklu robotların koordinasyonu çözüm olarak önerilmektedir. Koordinasyonu sağlamak amacıyla, son yıllarda popüler olan çoklu-robot pazar tabanlı görev alışverişinden esinlenmiş bir hibrid alışveriş mekanizması kullanılmıştır. Pist üzerinde robotun hareket edebileceęi alanlar haritalanmış ve havaalanı pistleri için özgün bir yol planlama stratejisi geliştirilmiştir. Deney sonuçları, kazalara sebep veren yabancı nesnelerin temizliğinde robotların kullanımı hızlı, ucuz ve güvenli bir tercih olacağını göstermiştir.

WOLFER ve ark. [23] tarafından yapılan çalışmada, Indiana Üniversitesi içinde öğrenci merkezli bir program geliştirilmiştir. Bu çalışmada öğrencilere assembly dilini öğretmek için özerk robotların programlanması yapılmıştır. Robot üzerinde bulanık tabanlı bir motor ve kontrolünü sağlayan bir yazılım üzerinde durulmuştur. Bir sonraki çalışma c# üzerinde bir yazılım ile robotun kontrolünün sağlanmasıdır olmuştur. Bu projeler birlikte, assembly dil programlanmasında robot geliştirilmiş olup bir ortamda uygulanabilirliğinin göstermeye yardımcı olmuştur.

THILMANY [24] tarafından yapılan Robotic Brooms araştırmasında, Amerika Birleşik Devletleri Deniz Kuvvetlerinin teknelerin altına yerleştirilen sualtı mayınlarını aramak amacıyla eğittiği yunus ve denizaslanlarının üzerinde durmuştur. Bu sistemler kısa süre içerisinde geniş bir alanı tarayabilseler de, eğitimleri ve bakımları oldukça maliyetlidir ve her defasında beklenen performansı da gösterememektedir. MIT'de dolaşarak bulanık sualtı çevrelerinin haritasını çıkarabilen ve iPod kadar küçük boyutlardaki mayınları tespit edebilen tümüyle otonom özelliğe sahip robotların üretilmesi için çalışılmaktadır.

BOUFERA ve ark. [25] tarafından yapılan çalışmada, bilinmeyen ortamdaki bulunan mobil robotların engel aşma sorunu üzerinde bulanık mantıklı hibrit bir yaklaşım önerilmektedir. Güvenli ve esnek navigasyon elde etmek için temel limit çevrim yöntemi geliştirilmiştir. Bu amaçla kaçınma sırasında yön değiştirme sayısını azaltarak kaçınmayı daha esnek hale getirmektir. Önerilen algoritma, simülasyon üzerinde farklı konfigürasyonlarda başarıyla test edilmiştir.

OKATAN ve ark. [26] tarafından yapılan çalışmada, robotlara yönelik uygulamalarda en önemli engel algılama yöntemlerinden biri olan ses dalgalarıyla engel algılama alanında uygulamalar yapılmıştır. Üç tekerlekli mobil bir aracın Motorola 16HC11 mikro işlemci kontrolünde ultrasonik sensör kullanarak yolu üzerindeki engelleri algılayıp, bu engellere çarpmadan yol alması sağlanmıştır. Projede ekipman olarak iki adet DC motor, bir adet MIT Handyboard ve bir adet Devantech SFR04 ultrasonik sensör kullanılmıştır. Projede programlama dili olarak C dillerinin bir türevi olan Interactive C ve Assembly kullanılmıştır.

Ayrıca proje kapsamına giren diğer konular ise; DC motor kontrolü, mikroişlemci ile sensör iletişimi, fuzzy kontrolü, ultrasonik hesaplamalar (cm hassasiyetinde mesafe hesaplaması) olmuştur. Projede çalışılan mobil robot program içerisinde kendime verilen süre ve yönde harekete başlamaktadır. Bu hareket esnasında yolu üzerindeki engelleri her 0,1 saniyede yolladığı 40KHz frekansa sahip ses dalgalarıyla algıladıktan sonra etrafından dolaşarak gideceği mesafeye ulaşmaktadır.

OKATAN ve ark. [27] tarafından yapılan çalışmada, holonomik olmayan bir robotik elektrikli süpürge için basit bir bulanık mantık navigasyon ve kontrol tasarımı sunulmuştur. Gömülü kontrol, Motorola mikro denetleyici kullanılarak tasarlanmıştır. Temizleme algoritması yürütülmekte olan alanı, ikili rastgele dizilerin temel felsefesine ve engel oluşumuna göre yönlendirilen alanı temizlemek için kullanılmaktadır. Herhangi bir engel, çarpışma gerçekleşmeden önce kızıl ötesi detektörler tarafından algılanmaktadır. Buna karşılık, navigasyon ve kontrol algoritması, muhtemel bir çarpışmayı önlerken, robot temizleyiciyi yakın çevresindeki başka bir bölgeye yönlendirmiştir.

KAYA ve ark. [28] tarafından yapılan çalışmada, statik ortamda nesne etiketleme için eş zamanlı çalışan otonom bir robotla, içinde bulunduğu ortamdaki nesnelere tanıma, ortamın haritasını çıkarma ve bir hedefe giden en kısa yolu bulma sistemleri geliştirilmiştir. Projenin amacı, robotun belli başlangıç ve bitiş noktaları arasındaki mesafeyi nesnelere ve duvar ile temas etmeden hareket ederek, etrafında gördüğü nesnelere şekil özelliklerini tanıyıp konum bilgileri ile beraber etiketlenmesini sağlamak olmuştur. Robotun önünde bulunan kameradan aldığı görüntüdeki nesnelere tanıyabildiği ve hareketlerini otonom değiştirebildiği bir sistem oluşturulmuştur.

HÜLAKO ve ark. [29] tarafından yapılan çalışmada, otonom bir İKA (İnsansız Kara Aracı) varış noktasına gönderilmesi sağlanmıştır. Çalışmada GPS, pusula ve mikrodenetleyiciden faydalanılmıştır. Başlangıçta varış noktaları (waypoints) veya coğrafi koordinatlar mikro denetleyicinin eepromuna yüklenmektedir. GPS modül ise kuzey çizgisi ve anlık konum-varış noktası arasında oluşan açıyı hesaplayabilmek için her 200ms'de bir kendi konumunu hesaplamaktadır.

Pusula sensörü de aracın kuzey ile yaptığı açığı hesaplamaktadır. Bu bilgilere göre mikrodenetleyici sonuçları karşılaştırılıp İKA'nın yön kontrolünü yapan servo motorunun yapacağı doğru açığı seçmektedir. Bu düzenek güzergâh yolu veya yörüngesi izleme yeteneklerinin geliştirilmesi için test düzeneği olarak kullanılmıştır.

ARDA ve ark. [30] tarafından yapılan çalışmada, gömülü sistem (ATMEGA328 Entegresi) üzerinden kontrol edilen bir DC Servo Motorun gerilimi, Bulanık Mantık yöntemli PID Denetleyici ile Ziegler ve Nichols'un önerdiği optimizasyon parametrelerinin uygulandığı klasik PID denetleyiciye olan etkisinin belirlenmesi amaçlanmıştır. DC servo motorun gerilim kontrolü, önce sadece PID denetleyici ile daha sonra ise bulanık mantık yönetimindeki PID denetleyici ile yapılmıştır. Her iki durum için PID denetleyicinin performans verileri çıkarılarak bulanık mantık yönetiminin PID denetleyici performansına etkisi gözlemlenmiştir. DC servo motor MATLAB/Simulink programı üzerinden gerçek zamanlı kontrol edilmiş ve kontrol algoritması bu program üzerinde oluşturulmuştur.

SARIEL ve ark. [31] tarafından yapılan çalışmada, tamamıyla özerk olan etkin bir alan tarama planlama yöntemi geliştirilmiştir. Bu planlama stratejisi uygulama alanına göre değişen görevlere sahip, kendi başına ortamda dolaşarak ilgili görevini yerine getiren robot sistemleri ve bu robotların hava yoluyla uygun yerlere dağıtımını sağlayan bir genel dağıtım birimi için tasarlanmıştır. Özellikle geniş alan taramasını sağlayacak olan yöntemin mayın tarama gibi insanlar için oldukça tehlikeli olan uygulamalarda kullanımı ile bu tür uygulamalarda zararı sadece maddi boyutlara indirgemek mümkün olacaktır.

YİNG [32] tarafından yapılan çalışmada, Bulanık PID kontrolü ile gerçekleştirilen ve yeni geliştirilen basitleştirilmiş Takagi-Sugeno (TS) kural şemasını kullanan yeni bir genel amaçlı doğrusal olmayan PID kontrolörü sunulmuştur. Bulanık PID denetleyicisinin analitik yapısı türetilmiş ve yapısı doğrusal PID denetleyicisine göre analiz edilmiştir. Bulanık denetleyicinin birinci özelliği, oransal, integral ve türev kazançlar, kontrol altındaki sistemin çıktısıyla sürekli olarak değişmesidir. Kazanç değişimi, karşılaştırılabilir bir lineer PID denetleyicisine kıyasla daha kısa bir artış zamanı, daha az bir aşma ve daha kısa bir yerleşim süresi sağlar.

İkincisi, kazanım varyasyonlarının bulanık kurallar tarafından belirlendiği ve sezgisel olarak tasarlanabileceği düşünülmektedir. Bulanık PID kontrol sistemlerinin yerel istikrarını da araştırılmıştır. Bir uygulama sunumu olarak, cerrahi sonrası hastalarda bilgisayarlı simülasyonda kan basıncını düzenlemek için bulanık bir PID kontrol sistemi geliştirilmiştir. Çünkü literatürdeki çalışmalar tatmin edici kontrol sonuçlarını elde etmek için değişken kazanımlara sahip doğrusal olmayan bir kontrolörün kullanılması gerektiğini tespit etmiştir. Simülasyon sonuçları, bulanık PID denetleyicisinin doğrusal karşılığı daha iyi performans gösterdiğini ve çok çeşitli hasta koşullarında daha güvenli ve daha sağlam olduğunu göstermektedir.

DOĞAN ve ark. [33] tarafından yapılan çalışmada, Matlab/Simulink yazılımı kullanılarak çeyrek araç modeli tasarlanmıştır. Tasarlanan bu modelin kontrolü, sırasıyla, geleneksel bir kontrol yöntemi olan PID kontrol yöntemi ve modern kontrol yöntemlerinden birisi olan Bulanık Mantık kontrol yöntemi ile gerçekleştirilmiştir. PID kontrol ve bulanık mantık kontrol yöntemlerinde, kontrol parametreleri kontrol performansının iyileştirilmesi için oldukça önemlidir. Parametrelerin en iyi olarak belirlenmesi süspansiyon sisteminden daha iyi sonuçlar elde edilmesine imkân vermektedir. Bu çalışmalar ışığında PID kontrol yöntemini kullanılan çeyrek araç modelindeki etkinliği bulanık kontrol yöntemine göre daha iyi olduğu saptanmıştır.

KAPLAN ve ark. [34] tarafından yapılan çalışma, BOUN (Boğaziçi Üniversitesi) Robot takımının oluşturduğu ve ekibe yeni katılan üyelerin robotlar hakkında bilgi sahibi olmaları için oluşturulmuş; otonom araçların tanımı, tarihsel gelişimi ve türleri, otonom araçların algılama, planlama, hareket ettirme ve kontrol sistemlerinin geliştirilmesi, otonom araçların duyuları ve algılayıcıları, biyolojik tabanlı oluşturulmuş robotlar hakkında ve bir robotun nasıl kontrol edileceği hakkında oluşturulmuş kaynaklardır.

BODRUMLU ve ark. [35] tarafından yapılan çalışmada, matematiksel modeli elde edilen Qball X4 dört pervaneli rotor sisteminin, verilen referansı nasıl takip ettiği ve bu referans takibinin iyileştirilmesi için kullanılan yöntemler anlatılmıştır. Başlangıçta referans takibi için klasik PID yöntemi kullanılmış, daha sonra PID tipli bulanık kontrolör tasarlanmıştır.

Bu tasarlanan kontrolörün performansını arttırmak adına bir sonraki aşamada ise ölçekleme katsayılarını kendiliğinden ayarlayan PID tipli bir bulanık kontrolör tasarlanmıştır. Bu kontrolörlerin nasıl tasarlandığı detaylı olarak anlatılmıştır. Bu çalışmada elde edilen tüm grafikler MATLAB™ simülasyon ortamında oluşturulmuştur.

TETİK ve ark. [36] tarafından yapılan çalışmada, PID ve bulanık mantık kontrol teorilerin bir top ve kiriş problemine uygulaması verilmiştir. Top ve kiriş sisteminin matematiksel modeli MATLAB/Simulink'de oluşturulmuştur. PID ve bulanık mantık kontrol teorilerinin top ve kiriş problemine uygulanması kıyaslamak için MATLAB/GUI'de arayüz hazırlanarak, kontrolör etkileri karşılaştırılmıştır. Sistem üzerine uygulanan kontrol etkisinin görselleştirilmesi için MATLAB/Virtual Reality 'de basit olarak sistem oluşturularak, kontrol etkisi izlenmiştir.

YAVAS ve ark. [37] tarafından yapılan çalışmada, dizel motorun hava hattındaki egzoz gaz çevrimi olan EGR (Exhaust Gas Recirculation) ve aşırı doldurma sistemi olan VGT (Variable Geometry Turbocharger) için temel kontrol çevrimleri açıklanarak, yapının kontrolör parametrelerinin belirlenmesinde daha önce AVL Türkiye tarafından geliştirilen yazılım yardımı ile otomasyonu yapılmıştır. Geliştirilen yazılıma yapılan eklemeler ile süreç geliştirilmiştir. Otomotiv sektörünün kısıtlı motor geliştirmeye süresi ve kalibrasyonu yapılması gereken PID denetleyici sayısı düşünüldüğünde, önerilen süreç sayesinde hem denetleyici performansının kabul görmüş metotlarla elde edilmesi hem de sürecin kısaltılması hedeflenmiştir.

TURHAN [38] tarafından yapılan çalışmada, kısıtlı hareket yeteneğine sahip (non-holonomic) tekerlekli gezgin robot (TGR) Pioneer 3-DX için, sensör tabanlı rastgele ağaç yapısı (Sensor Based Random Tree - SRT) algoritması kullanılarak ortam haritasının çıkartılması uygulaması gerçekleştirilmiştir. Algoritma, ortam haritasını ziyaret edilen pozisyonlar ve bu pozisyonlarla ilişkilendirilmiş engel bulunmayan lokal bölgeler olarak tutmaktadır. Gerçekleştirilen uygulamada engel bulunmayan lokal bölgeleri tespit etmek için lazer mesafe algılayıcı sensör kullanılmıştır.

Sensör tabanlı rastgele ağaç yapısı algoritmasının türevleri olan SRT-Ball ve SRT-Star algoritmaları MobileSim Robot simülatörü kullanılarak gerçekleştirilmiş ve karşılaştırılmıştır.

ÖKSÜZ ve ark. [39] tarafından yapılan çalışmada, sabit mıknatıslı senkron motor (SMSM) tahrikli servo presin konum denetimi bulanık mantık yöntemli PID ile yapılmıştır. Servo presin konum denetimi geliştirebilmek için kaskad denetim sistemi ve bulanık mantık algoritması kullanılmıştır. Bulanık mantık yaklaşımı ile PID parametreleri bulunmuştur. Benzetim olarak çalışan servo pres sistemi, deneylerle denenmiş, sonuçlarda kaç hareketinde hassasiyetin büyük oranla arttığı gözlemlenmiştir.

RYU ve ark. [40] tarafından yapılan çalışmada, bir tür ters sarkaç tipi sistem olan Segway mobil robotun kontrolü için iki bulanık mantık kontrol sisteminin tasarımı önerilmektedir. Önce Segway mobil robotunun dinamik bir modelini uyandırıp sistemi analiz edilmiştir. Daha sonra herhangi bir doğrusal olmayan sistemin kontrolü için iyi performans gösteren bulanık mantık kontrol ile sisteminin tasarımı kullanılmıştır. Simülasyon ve örnekler aracılığıyla faydaları gösterilmiştir.

TASKIRAN ve ark. [41] tarafından yapılan çalışmada, QBot 2 mobil robot, manevra kabiliyeti açısından değerlendirilmiştir. Bu sayede QBot 2 mobil robotun gerçek zamanlı olarak kontrol edilmesi planlanırken bir komut tanıma sistemi gelişmiştir. QBot 2 ile bağlantı kablosuz bir ortamda gerçekleştirilmektedir. MATLAB ortamında bir Simulink modeli geliştirilmiştir. Oluşturulan model Quarc kontrol yazılımı ile oluşturulmuştur. Derlenmiş model TCP / IP bağlantısı ile QBot 2'ye indirilir ve uygulama bir bilgisayarda gerçekleştirilmiştir. QBot 2 mobil platformu, ortak bir eksene monte edilmiş iki merkezi tahrik tekerleğinden oluşmuştur. Bu sürücü yapılandırması diferansiyel sürücü olarak bilinmektedir. İki tahrik tekeri, robotu harekete geçirmek için bağımsız olarak ileri ve geriye doğru sürülmektedir. Tekerleklerin hareketi, yüksek performanslı DC motorlarla gerçekleştirilmiştir. Sonuçlar analiz edildiğinde toplamda 13 farklı hareket türü gözlemlenmektedir.

Gözlemlenen hareket tipleri, insan sesi ile QBot 2 mobil robotun uzaktan kumandası gibi pek çok pratik uygulama farklı hareket türlerinin bulunmasını gerektirdiğinden gelecek çalışmalarında referans olarak kullanılabilir.

HACIMURTAZAOĞLU ve ark. [42] tarafından yapılan çalışmada, bulanık uzman sistem helikopter bulanık mantık ile take-off ve iniş sağlanması için tasarlanmıştır. Bu amaçla, model helikopter ve test platformu helikopter hareketleri test edilebilir oluşturulmuştur. Bulanık mantık kontrol sistemi tarafından tanımlanan eksen dikey hareket kontrolü dört derece ile oluşturulan test platformu ile helikopter hareketi sağlamaktadır. Bulanık mantık hareket kontrolü helikopter modeli ve analizi kontrol etmek için Arduino 2560 Denetim Kurulu ve Visual Studio 2010 c# programı ile tasarlanmıştır. Test platformunda oluşturulan helikopter gerçek zamanlı kontrol ampirik çalışma ile incelenmiştir. Helikopterin kontrolü sırasında, helikopter imu sensör değerleri eksen ve iki helikopter modeli üzerinde yerleştirilmiş olan Ultrasonik mesafe sensörü ile mesafe değerleri Arduino ile okunup ve bilgisayara gönderilmektedir. İşlenmiş anlamlı veri giriş değerleri oluşturmak için denetleyici yazılımı ile işlenmiştir. Sonuç olarak dikey eksen helikopter hareketi kontrol deney ve simülasyon olarak incelenmiştir.

Sonuç olarak literatür araştırmalarında da görüldüğü gibi PID ve Bulanık mantık denetimli çalışmalarda kullanılan sistemlerin verimliliği karşılaştırılmış; basit yapıları ve sistemlerde sergiledikleri değişken koşullara çabuk ve kolay uyum sağlayan ve doğrusal olmayan sistemlerdeki başarısıyla bulanık mantık denetimi bu tezin en önemli unsuru olmuştur. Bu kontrolör göz önüne alınarak bir otonom robot ve sabit pist ortamı oluşturulmuştur. C# arayüzü üzerinden robotun pist üzerindeki hareketleri izlenerek hata payı c# üzerinde ölçülmüştür. Alınan sonuçlar Matlab ortamında denetlenerek doğruluğu ölçülmüştür. Çizelge 1.1'de yapılan literatür araştırmasında yer alan çalışmalarda kullanılan teknolojiler ve yazılımlar göz önüne alınarak özeti verilmiştir.

Çizelge 1.1 Literatür Taraması

Kaynak Numarası	Kullanılan Teknoloji	Kullanılan Yazılım
1	Lazer, Ultrasonik, Kamera algılayıcıları	OpenCV, Microsoft Robotics Development Studio
2	Pioneer 3-DX	Genetik Algoritma
3	Lazer, Jireskop, İvme ölçer, Manyetik Sensör ve Enkoder	Karınca koloni algoritması
4	Otonom Robot	A* arama algoritması
5	Pioneer 3-DX	Hareket Planlama ve Kontrol Yaklaşımı
6	U-box Neo-4S GPS Modül ve TSSAGA sistemi	C++
7	İHA	Genetik Algoritma
8	Otonom Robot	OpenGL
9	Otonom Su Altı Aracı, Openpilot GCS	Kalman Algoritması, Matlab
10	Gömülü sistem	Bulanık Mantık
11	-	PID ve Bulanık Kontrolör
12	Rüzgar Türbini	PID ve Bulanık Kontrolör, Matlab Simulink
13	-	PID ve Bulanık Kontrolör
14	Ph Kontrol Deney Seti ve Proses Kontrol Simülatörü	Bulanık Kontrolör ve Matlab
15	-	Bulanık PID Kontrolör ve Klasik PID Kontrolör
16	-	Bulanık PID Kontrolör
17	Gimbal	Kendinden Ayarlamalı PID Kontrolör
18	-	P, PI, PD ve PID Kontrolör
19	Otonom Mobil Robot	C++ ve Matlab
20	-	-
21	RF Teknolojisi	Duvar Takip ve Böcek Algoritması
22	Radar, Kamera	-
23	-	Visual Studio c#, Bulanık Mantık
24	Otonom Robot	-
25	Mobil Robot	-
26	Motorola 16HC11, Ultrasonik Sensör	Interactive c ve Assembly ile Fuzzy Kontrolü
27	Temizlik Robotu	Bulanık Mantık
28	Otonom Robot	En Kısa Yol Algoritması
29	Otonom Robot(İKA), GPS, Pusula, PIC18F452	PID Kontrolör
30	Gömülü Sistem	Klasik PID ve Matlab Simulink
31	Mayın tarama Robotu	C++ ve OpenGL

(Çizelge 1.1 devam ediyor.)		
32	-	Bulanık PID Kontrolör, Takagi-Sugeno Kural Şeması
33	-	Matlab Simulink, PID Kontrol, Bulanık Kontrol
34	-	Robotlar Hakkında Genel Bilgi Kaynağı
35	Qball X4	Klasik PID ve Matlab
36	-	PID, Bulanık Mantık Kontrol ve Matlab Simulink
37	EGR ve VGT	PID Denetleyici
38	Pioneer 3-DX	Sensör tabanlı Rastgele Ağaç Yapısı Algoritması
39	Servo Pres	Bulanık Mantık Yönetimli PID
40	Segway Mobil Robot	Bulanık Mantık Kontrol
41	QBot 2 Mobil Robot	Quarc Kontrol, Matlab
42	Arduino 2560	Bulanık Mantık, Visual Studio c#

3. TASARLANAN ROBOTUN YAPISI

3.1. Altu Robotun Temel Parçaları

Altu robot olarak isimlendirilen ve çalışmaların ana hatlarını oluşturan robotun gövdesi Şekil 3.1’de verilmiştir. Tasarlanan robotun gövdesi alt ve üst kısım olmak üzere iki bölüm halinde oluşturulmuştur.

Gövde genel hatlarıyla 140 mm x 225 mm uzunluklara sahip ve 5 mm kalınlıkta pleksiglas bir yapıdır.

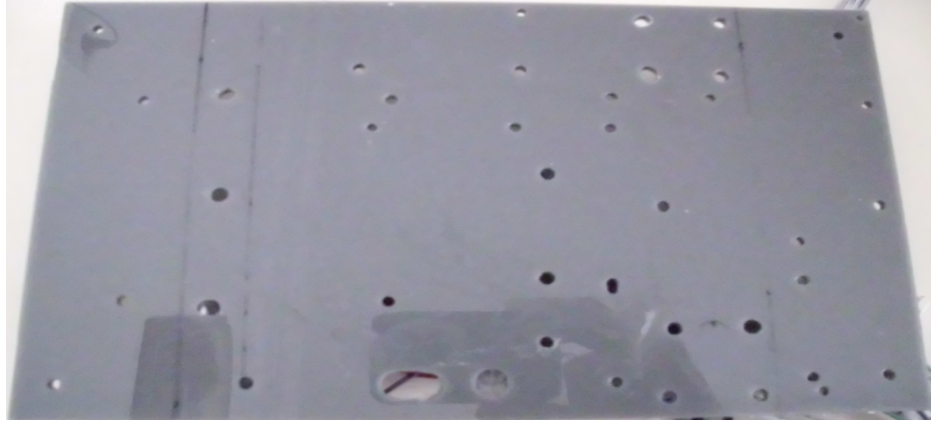
Gövdenin üst bölümünde:

- Dört adet HC04 ultrasonik sensör,
- Bir adet kablosuz bağlantı noktası (HC05),
- Bir adet 74HC14 Schmitt Trigger devresi,
- Bir adet Mikrodenetleyici kartı (ATMEGA_2560),
- Bir adet LM2576-5v Regüle devresi kullanılmıştır.

Gövdenin alt bölümünde:

- Bir adet 11.1V 800 mA Li-po pil,
- Bir adet 7.4V 1050 mA Li-po pil,
- Bir adet DC motor sürücü kartı,
- 4 adet On-OFF buton ve şarj ünitesinin bulunduğu bölüm,
- Dört adet 12V 500Rpm DC Motor
- Dört adet standart Servo Motor
- Dört adet silikon teker
- Bir adet 7805 5V Regüle Devresi,
- İki Adet QRD1114 Kıızıl Ötesi Sensör bulunmaktadır.

Tasarlanan robot üzerinde kullanılan malzemeler literatür araştırmasında karşılaşılan çalışmalarda ve genel olarak yangın söndüren, labirent çözen, triatlon robot, çizgi izleyen ve engellerin olduğu aşamalı robot kategorilerinde kullanılan malzemelerdir.



Şekil 3.1 Altu Robotu Oluşturan Gövde Parçası

Robotun alt gövdesine yerleştirilmiş DC Motor tutucuları Şekil 3.2 (solda)'da, robotun alt bölümünde kullanılan servo motorların kenar tutucuları Şekil 3.2 (ortada)'da, servo motorun üst plastik tutucu parçası Şekil 3.2 (sağda)'da verilmiştir.



Şekil 3.2 Altu Robotu Oluşturan Tutucu Parçalar, DC Motor Tutucu Parçası(solda)
Servo Motor Kenar Tutucuları (ortada), Vidaları ve Somunları (ortada) Servo Motor
Dönen Tutucu Parçası (sağda)

Altu robotun alt bölümünde kullanılan DC motor ve servo motor tutucularının birleştirilmesiyle oluşturulan parça Şekil 3.3'de verilmiştir.



Şekil 3.3 Altu Robotun DC ve Servo Motorlarının Tutucularının Birleştirilmesi

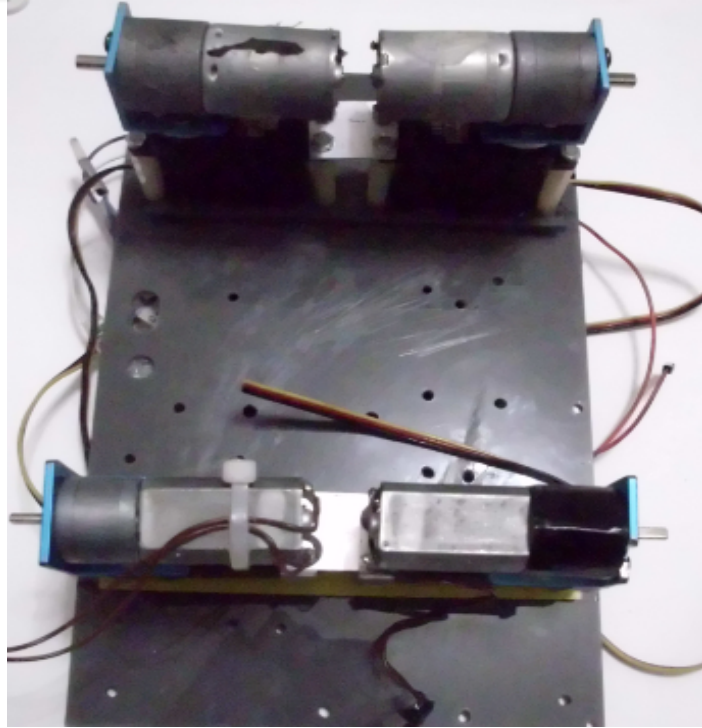
Robot üzerinde bulunan sensörler ve haberleşme cihazında kullanılan aktarma kabloları Şekil 3.4 (solda)'da, robotun üst bölümü ile alt bölümü arasında bulunan malzemelerin gövde ile birleştirilmesinde kullanılan destek vidalar Şekil 3.4 (sağda)'da verilmiştir.



Şekil 3.4 Altu Robot Üzerinde Kullanılan Kablolar (solda) 30mm Vida (sağda)

3.2. Robotun Gövdesi

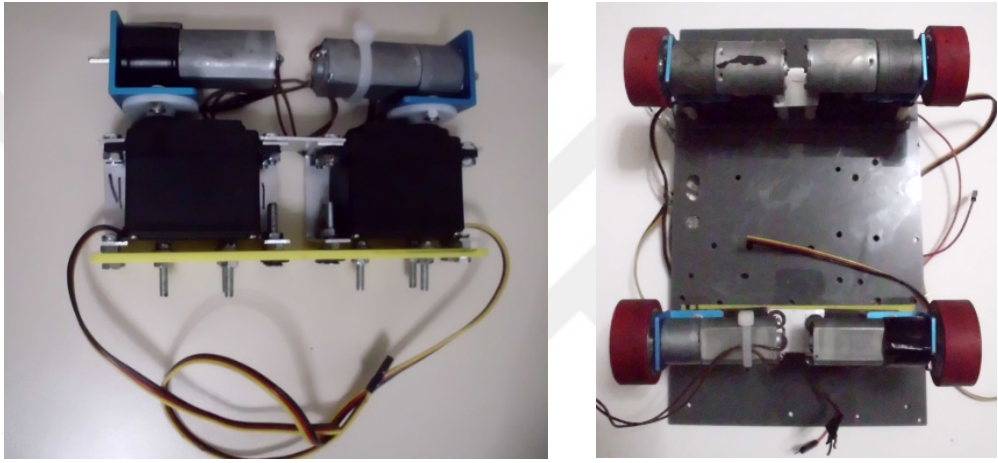
Altu robotun gövde parçası Şekil 3.1'de verilmiştir. Servo motorların ve DC motor tutucu parçaların bir araya getirilmesiyle ortaya çıkan alt gövde parçası Şekil 3.5'de verilmiştir



Şekil 3.5 Altu Robotun Alt Gövdesi

3.3. Motorlar ve Tekerler

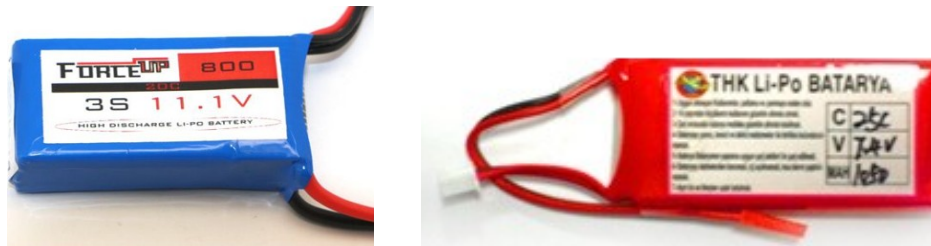
Altu robotun hareket mekanizmasının bel kemiği olan DC motorlar, servo motorlar ve tekerlerin birleştirilmiş hali Şekil 3.6'da verilmiştir. DC motorlar 12V 500 RPM gücündedir. DC motorlarda silikon yapılı, kırmızı renkte ve yüzey yapışması arttırılmış tekerler kullanılmıştır. Servo motorlar standart özelliklerde olup 0-180 derece arası çalışmaktadır. Robot üzerinde dengenin sağlanması ve dönüş işlemlerinin daha kolay olması için dört adet servo motor ile dört adet DC motor kullanılmıştır.



Şekil 3.6 Altu Robotun Alt Gövdesine Bulunan Servo ve DC Motorların Alt Gövdeye Sabitleme Aşaması (solda) Alt Gövdeye DC Motorların, Servo Motorların ve Tekerlerin Eklenmesi (sağda)

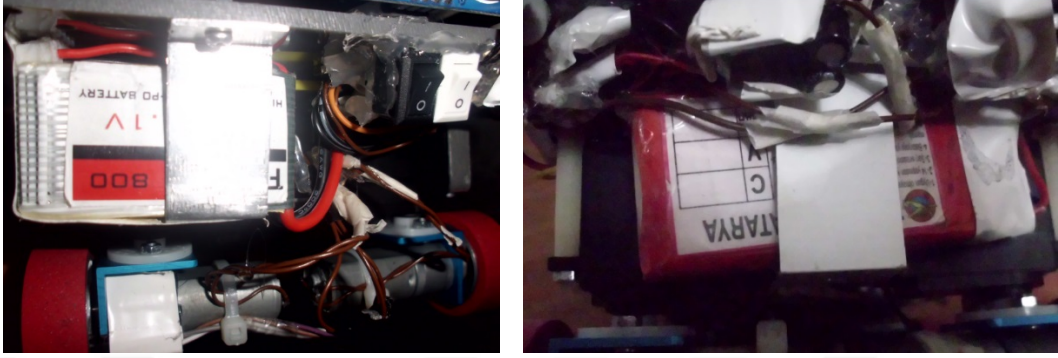
3.4. Kullanılan Güç Sistemi

Altu robotun üzerinde bulunan mikrodenetleyici kartına, motor sürücüyeye, sensörlere, DC motorlara gerekli olan beslemeyi sağlayan 11.1V Li-po pil Şekil 3.7 (solda)'da verilmiştir. Alt gövde üzerinde bulunan servo motorlara enerji veren harici 7.4V Li-po pil Şekil 3.7 (sağda)'da gösterilmektedir.



Şekil 3.7 Altu Robotta Kullanılan 11.1V Li-po pil (solda) 7.4V Li-po (sağda)

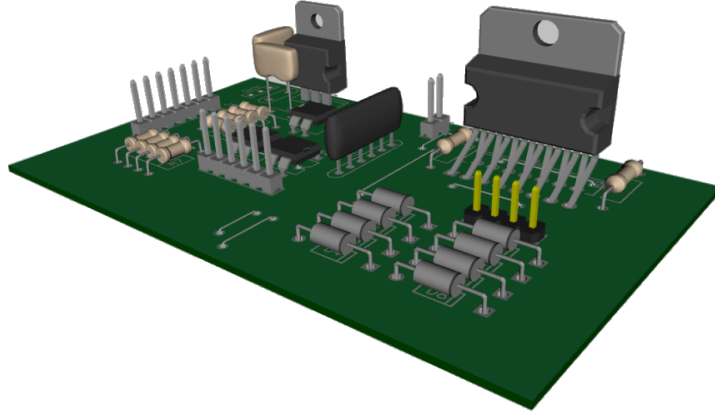
Pillerin robot üzerindeki konumları Şekil 3.8’de verilmiştir.



Şekil 3.8 Altu Robotun Gövdesi Üzerinde Bulunan Li-po Pillerin Konumu.

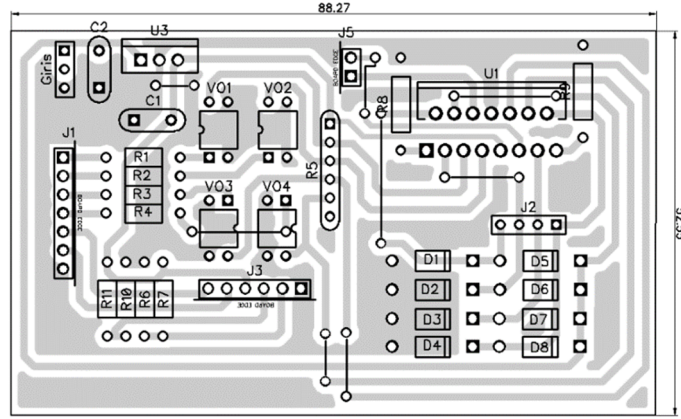
3.5. Motor Sürücü Kartı

Tasarlanan motor sürücü kart üzerinde dahili regülatör, dört adet optokuplör, hat koruma dirençleri ve soğutucu vardır. Sumo, Mini Sumo, Labirent, Triatlon, Çizgi İzleyen ve birçok robot uygulamasında kullanılmaktadır. Şekil 3.9’da verilen L298 Motor sürücü kartı, iki kanallı olup kanal başına 2 A akım verebilmektedir.



Şekil 3.9 Altu Robot İçin Tasarlanan L298 Entegreli Motor Sürücü Devresinin Üç Boyutlu Gösterimi

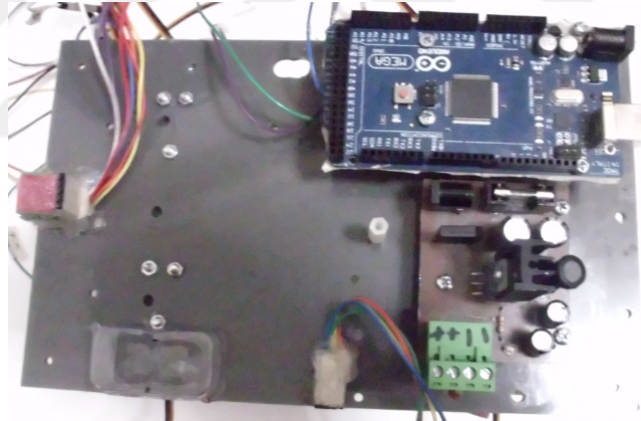
Şekil 3.10’da Altu robot için tasarlanan L298 motor sürücü devresinin malzeme yerleşim yüzeyleri verilmiştir. EK-4’de L298 Motor sürücü entegresinin standart pin şeması, EK-5’de L298 entegresi için kullanılan standart devre şeması ve EK-6’da L298 pin bağlantı şeması verilmiştir.



Şekil 3.10 Altu Robotun Alt Gövdesinde Bulunan Motor Sürücü Kartı

3.6. Mikrodenetleyici Kartı

Altu robot üzerinde kullanılan Arduino Mega 2560 mikrodenetleyici kartının gövde üzerinde bulunduğu yer Şekil 3.11’de verilmiştir.

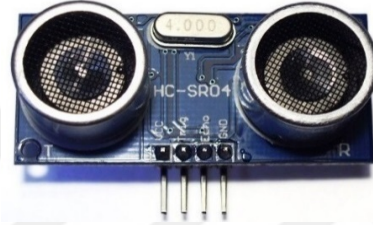


Şekil 3.11 Altu Robotun Mikrodenetleyici Kartı.

Mikrodenetleyici kartı üzerinde 54 adet giriş/çıkış pini bulunurken bu pinlerin 16 adeti analog giriş pinidir. 15 Adet PWM çıkışı 4 adet UART donanımı ve kart üzerinde 3.3V Regüle sistemi vardır. Kart USB üzerinden veya ICSP pinleri üzerinden programlanabilmektedir. Kullanılan mikrodenetleyici kartının devre şeması EK-1’de, kartın genel görünümü EK-2’de ve pin bağlantı şeması EK-3’de verilmiştir.

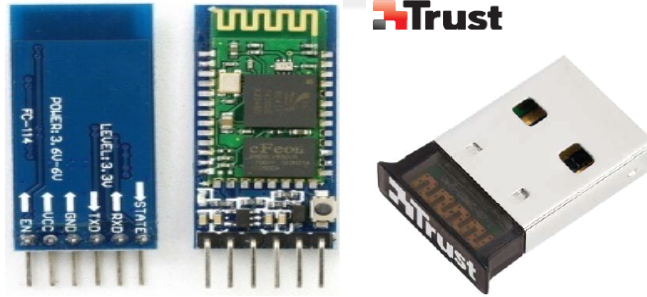
3.7. Konumlandırma Sensörü

Bu çalışmada konumlandırma için kullanılan HC-SR04 ultrasonik sensör Şekil 3.12’de verilmiştir. HC-SR04 40 Khz frekansında ultrasonik ses üretmektedir. Ses dalgaları sınıflandırılmasında 20 Khz – 1 Ghz aralığındaki ses sinyalleri ultrasonik ses olarak tanımlanmıştır. Burada önemli olan sesin yüksekliğinde belirleyici olan etken frekanstır. Ses yüksekse frekansta yüksektir. Sensörden çıkan sesin gidiş geliş zamanı mesafeye göre orantılıdır. HC-SR04 ultrasonik sensörün çalışma şeması EK-7’de verilmiştir.



Şekil 3.12 HC-SR04 Ultrasonik Sensör

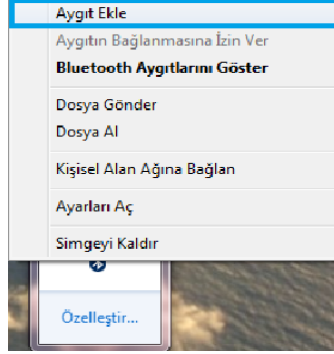
3.8. Kablosuz Haberleşme Birimi



Şekil 3.13 Altu Robotun Kablosuz HC05 Bluetooth Modülü (solda) PC USB Bluetooth Adaptörü (sağda)

Şekil 3.13 (solda)’da gösterilen kablosuz haberleşme ünitesi HC05, bluetooth 2.0.’ı destekleyen 2.4 Ghz frekansında kablosuz haberleşme uygulamaları için tasarlanmıştır. 10 metrelik haberleşme mesafesine sahiptir. HC05 modül düşük güç ile çalışabilmekte ve 3 V ile 5 V arasında kullanılabilir. Bu modülün Altu robot üzerinde temel görevi, seri portun yerini alabilmektedir. Modül RX ve TX pinleri üzerinden PC veya ikinci bir mikrodenetleyici ile kablosuz haberleşme protokolünü uyguladığında yöneten bölüm “Master” yönetilen bölüm ise “Slave” olarak iletişim sağlanır.

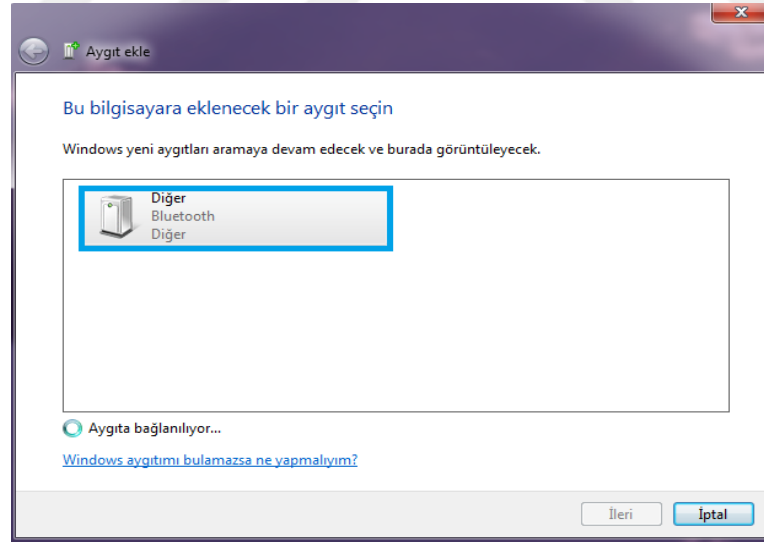
HC05 modülün kullanılan pin şeması EK-8’de ve HC05 modül devre şeması EK-9’da verilmiştir. HC05 Modül maksimum 10 metre çekim gücü sağlamaktadır.



Şekil 3.14 PC USB Modülün Tanıtılması

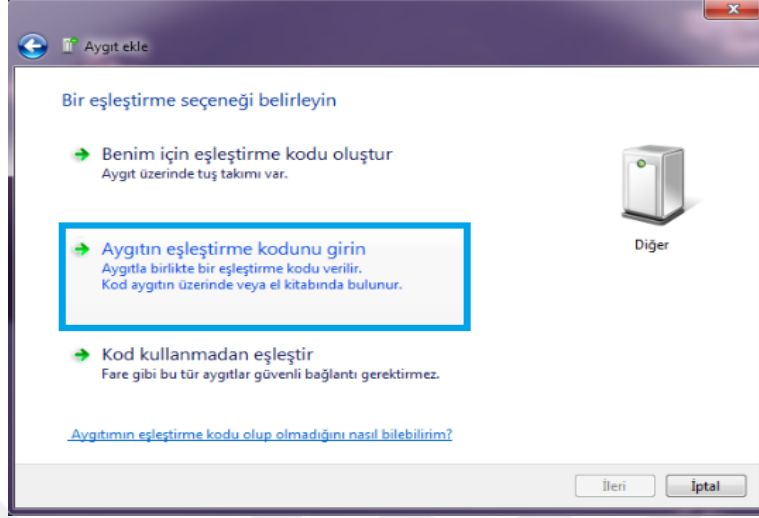
HC05 ile PC USB modüller arasında kurulacak olan kablosuz iletişimin kurulum aşamaları sırasıyla Şekil 3.14’den Şekil 3.19 arasında anlatılmıştır.

Şekil 3.14’de yeni PC USB modülün bilgisayar ortamına tanıtılması işlemi anlatılmıştır. Masaüstü sağ alt bölümde saat simgesinin yanında bulunan bluetooth simgesine sol tıklayarak “Aygıt Ekle” Şekil 3.14’de verilmiştir.



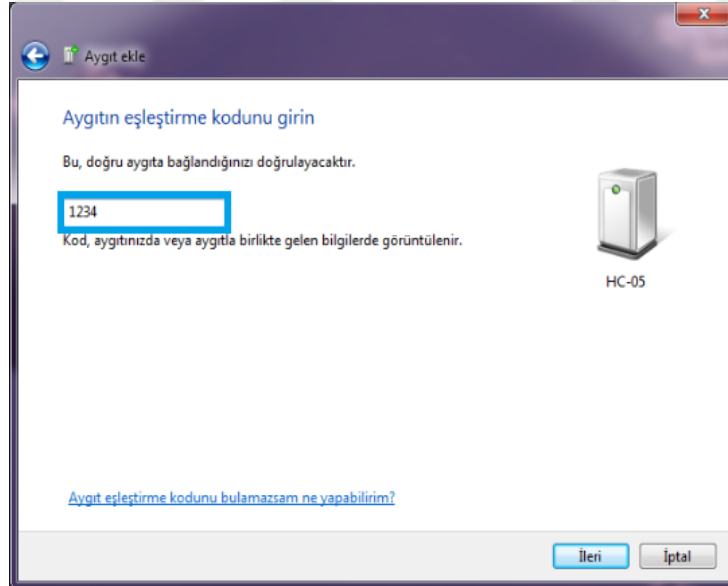
Şekil 3.15 PC USB Modül Üzerinde HC05 Bluetooth Modülün Görünmesi

Aygıt ekle işleminden sonra var olan Bluetooth donanımlarının aranıp, seçilip bağlantı kurma işlemleri gerçekleştirilecektir. Altu robot üzerinde bulunan HC05 modülünün aygıt ekleme ekranı Şekil 3.15’de verilmiştir.



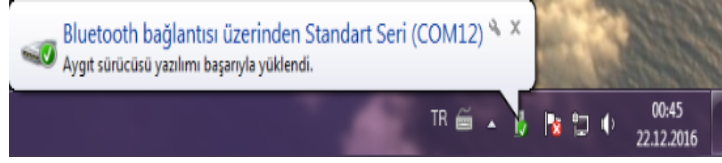
Şekil 3.16 PC USB Modül ile HC05 Modül Eşleştirilmesi

Aygıtın taranıp seçilmesi ile donanım üzerinden nasıl haberleşme yapılacağı Şekil 3.16’da verilmiştir. Haberleşmenin kurulabilmesi için HC05 Bluetooth modül ile Şekil 3.16’da verilen “Aygıtın eşleşme kodunu gir” bölümünü seçerek üretici firma tarafından oluşturulan eşleşme kodu “1234” girilir. Bu aşama Şekil 3.17’de verilmiştir.



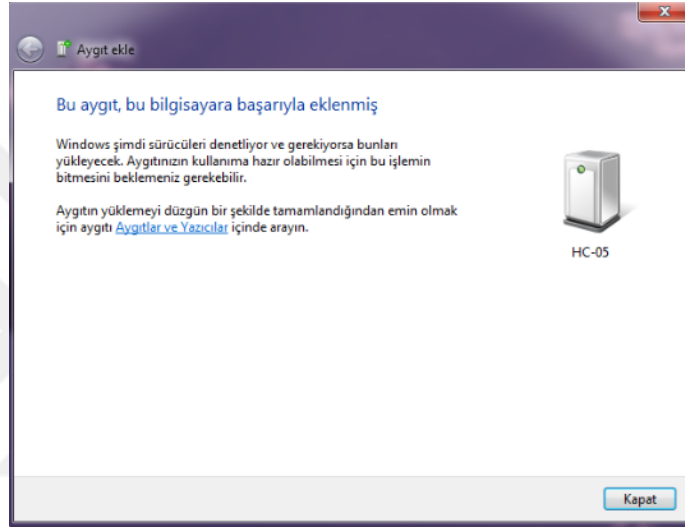
Şekil 3.17 HC05 Modülün Eşleşme Kodunun Girilmesi

İki modül arasında eşleştirilme kodu girilerek HC05 üzerinde com port açılır. Bu port PC Bilgisayarın masaüstünde sağ alt kısımda bir ileti şeklinde kullanıcıya iletilmektedir (Şekil 3.18).



Şekil 3.18 HC05 Modül ile PC USB Modül Arasında Oluşturulan Com Port

Bağlantı başarılı şekilde kurulduktan sonra Şekil 3.19’da gösterildiği gibi “Aygıtlar ve Yazıcılar” sekmesinde HC-05 Modülün port detayları görülebilir.



Şekil 3.19 HC05 Modülün Bilgisayar Üzerinde Başarılı Şekilde Eklenmesi

3.9. QRD1114 Optik Sensör

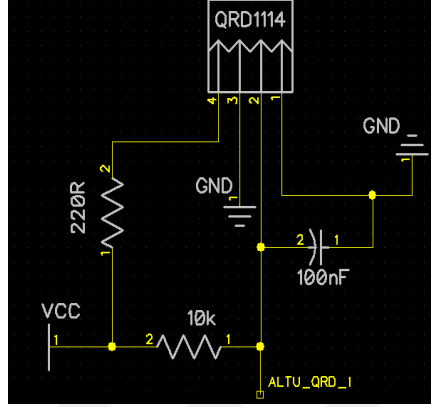
Şekil 3.20’de görülen QRD1114 bir IR LED “yayıcı diyot” ve bir adet NPN fototransistörden oluşmaktadır. Diyotun yaydığı ışık zemin üzerinde yansyarak fototransistörü tetiklemektedir.



Şekil 3.20 QRD1114 Optik Sensör

Beyaz zeminde yansıyan ışık siyah zeminden daha fazla olacağından renklerin ayırımı sağlanır ve analog sinyal alınır.

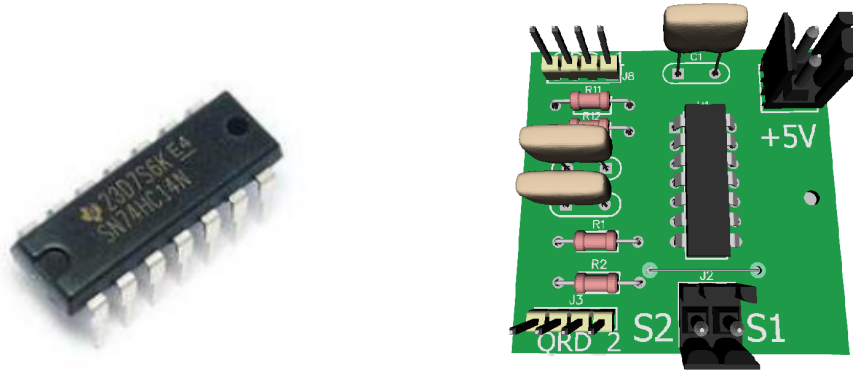
Fototransistörün iletken olmasıyla sensörün çıkışı Lojik “0” olur. Şekil 3.21’de QRD1114’ün kullanımına örnek bir devre şeması verilmiştir.



Şekil 3.21 QRD1114 Bağlantı Şeması

3.10. 74HC14 Schmitt Trigger Entegresi ve Devresi

74HC14 schmitt triger çevirici ile QRD1114 optik sensörün Lojik “1” ve “Lojik “0” seviyesinde çalışır. Fototransistör aktif ise “1” değilse “0” bilgisi mikrodenetleyiciye aktarılır.



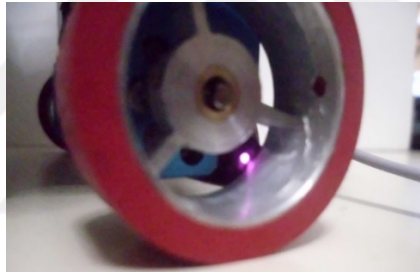
Şekil 3.22 74HC14’ün Genel Görüntüsü (solda) 74HC14 İle QRD1114’ün Birleştirilmesiyle Oluşturulan Tasarımın Üç Boyutlu Görüntüsü (sağda)

Altu robot üzerinde kullanılan schmitt triger entegresi Şekil 3.22 (solda)’da verilmiştir. Şekil 3.22 (sağda)’da 74HC14 İle QRD1114’ün birleştirilmesiyle oluşturulan tasarımın üç boyutlu görüntüsü verilmiştir.



Şekil 3.23 74HC14 ile Tasarlanmış Devrenin Görüntüsü

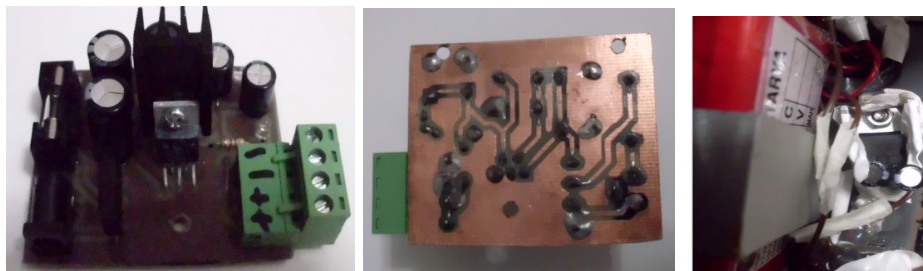
Altu robot üzerinde iki adet QRD1114 kullanılmıştır ve Şekil 3.23’de gösterilen elektronik devre oluşturulmuştur. Şekil 3.24’de QRD1114 optik sensörün çalışma durumuna ilişkin resim verilmiştir.



Şekil 3.24 Qrd1114 Optik Sensörün Çalışması.

3.11. LM2576-5v ve 7805 Regülatör Devreleri

Servo motorları ve Ultrasonik sensörleri sürmek için Şekil 3.25 (solda) ve Şekil 3.25 (ortada)’da gösterilen 5V 3 Amper regülatör devresi yapılmıştır. Harici besleme durumunda Servo motorları sürmek için kullanılan 5V 1.5 Amper regülatör ise devresi Şekil 3.25 (sağda)’da görülmektedir.



Şekil 3.25 LM2576-5V Regülatör Devresi (solda) LM2576-5V Regülatör Devresinin PCB Yüzeyi (ortada) 7805 5V Regülatör Devresi (sağda)

4. ROBOTUN KONTROL ve YAZILIM YAPISI

İnsansız araçlar insan unsuru bulundurmayan, uzaktan ya da otonom olarak yönetilebilen ve önceden belirlenmiş görevleri icra eden kritik alanlarda kullanılan teknolojilerdir. İnsansız araçlara olan talepte belki de en temel faktör bu araçların insanların emniyetle gidip dönemeyecekleri her yere gönderilebiliyor olmasında yatmaktadır. Otonom robotun hareketlerini yönlendirecek veya müdahalede bulunacak harici bir operatör yoktur. Sisteme bir defa başla komutu verildiğinde, beklenen tüm sıralı hareketler robot akıl mekanizması tarafından yönetilir [20].

Gezgin bir robot üç durumda çalışmaktadır. Bu durumlar; tam otonom, yarı otonom ve otonom olmayan robotlardır. Tam otonom robotlar, kendi kendine karar verebilme ve verdiği kararı uygulama yeteneğine sahip olan sistemlerdir. Yarı otonom robotlar, belirli durumlarda kararı kendisi verirken diğer durumlarda ise kullanıcı kontrollü olan sistemlerdir. Otonom olmayan robotlar, tüm kontrolü kullanıcı tarafından yapılan sistemlerdir.

Bu çalışma içerisinde Altu robota gönderilmiş olan verilerin planlı seyrüsefer işleminden geçerek robot içerisine bir harita bilgisi yüklenmeden alınan hedef bilgisi doğrultusunda çalışması sağlanmıştır. Yapılan planlı seyrüseferde robot ultrasonik sensörleri ve kızılötesi sensörleri yardımıyla sürekli çevreden bilgi toplayıp, alınan veriler doğrultusunda bir plan oluşturulup ve bu plana göre hareket edebilmesidir. MATLAB ortamında oluşturulan yazılım ile pist üzerinde benzetim çalışması yapılmış ve deney sonuçları matlab sonuçları ile karşılaştırılmıştır.

4.1. Bulanık Kontrollör

Altu robotun bulanık kontrol mantığında c# yazılımı üzerinden gelen konum ve açı verileri robot üzerinde bulanık mantıkta işlenip, gidilecek olan mesafeyi hesaplamak için kullanılmıştır. Açı ve hatalı konum verisi bulanık mantıktan sonra kontrol edilip c# üzerinde bilgisi görülebilmektedir.

Bulanık mantığın tez içerisindeki temel yapısı; konum verilerindeki belirsizlik durumlarını modellemek ve gelecekteki tahmin yapısındaki belirsizlikleri analiz etmektir. Altu bulanık birimi; giriş ve çıkış değişkenleri, bulandırma, bulanık çıkarım ve durulama bölümlerinden meydana gelmektedir.

4.1.1. Giriş ve Çıkış Değerlerinin Tanımlanması

Tasarlanan bulanık mantık birimin;

Giriş değişkenleri:

$X \text{ konum}(cm) = 0 - 100$ cm arası x eksenindeki robotun konum verisi.

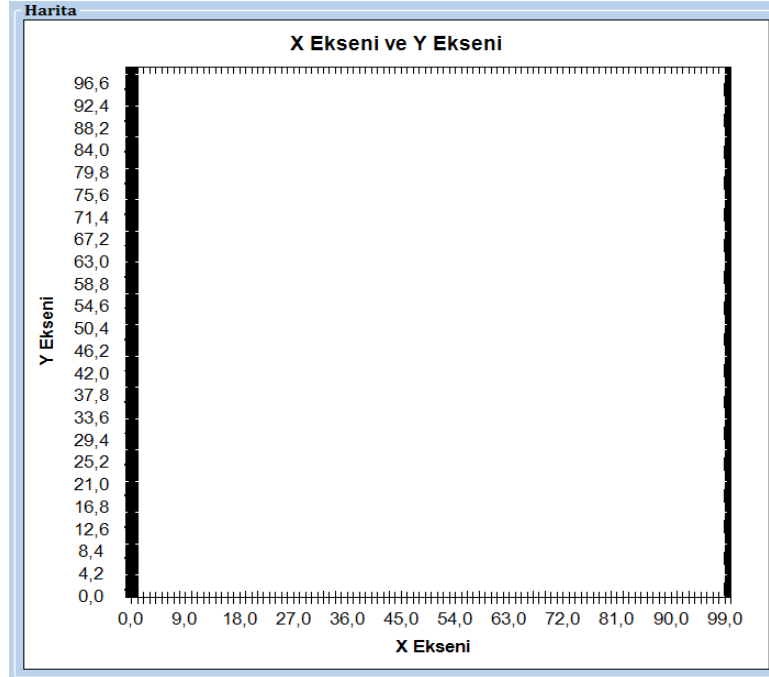
$Y \text{ Konum}(cm) = 0 - 100$ cm arası y eksenindeki robotun konum verisi.

$Açı(^{\circ}) =$ Altu Robotun eksenlere göre konum açısı verisi.

Çıkış değişkeni:

$Mesafe (cm) =$ X ve Y konum verileri ile Açısı verisine bağlı olan sistem çıktısı bilgilerinden oluşmaktadır.

Tasarlanan c# yazılımının bulanık mantık girdi ve çıkış değişkenlerine göre konum sisteminin çalıştırılacağı alan Şekil 4.1’de verilmiştir.



Şekil 4.1 C# Yazılımının Altu Robotun Hareket Alanı.

4.1.2. Bulandırma

Bulandırma işleminde, sayısal giriş ve çıkış değişkenleri sembolik değerlere dönüştürülmektedir. X Konum, Y Konum, Açık ve Mesafe bulanık kümelerinin alt kümelere ayrıldığı bölümdür.

Giriş ve Çıkış değişkenlerinin etiket ve değişim aralıkları X konumu için Çizelge 4.1, Y konumu için Çizelge 4.2’de, açı verisi için Çizelge 4.3’de ve mesafe verisi için Çizelge 4.4’de verilmiştir.

Çizelge 4.1 X Konumunun Alt Kümeleri

Alt Küme ismi	Detaylı İsmi	Aralığı
XCK	X Çok Küçük	[0, 20]
XKO	X Küçük Orta	[10, 40]
XK	X Küçük	[30, 50]
XO	X Orta	[40, 60]
XB	X Büyük	[50, 70]
XBO	X Büyük Orta	[60, 90]
XCB	X Çok Büyük	[80, 100]

Çizelge 4.2 Y Konumunun Alt Kümeleri

Alt Küme ismi	Detaylı İsmi	Aralığı
YCK	Y Çok Küçük	[0, 20]
YKO	Y Küçük Orta	[10, 40]
YK	Y Küçük	[30, 50]
YO	Y Orta	[40, 60]
YB	Y Büyük	[50, 70]
YBO	Y Büyük Orta	[60, 90]
YCB	Y Çok Büyük	[80, 100]

Çizelge 4.3 Açık Verisinin Alt Kümeleri

Alt Küme ismi	Detaylı İsmi	Aralığı
AK	Açık Küçük	[0, 30]
AKO	Açık Küçük Orta	[20, 40]
AO	Açık Küçük	[35, 55]
ABO	Açık Büyük Orta	[50, 70]
AB	Açık Büyük	[60, 90]

Çizelge 4.4 Mesafe Verisinin Alt Kümeleri

Alt Küme ismi	Detaylı İsmi	Aralığı
MKS	Mesafe Küçük Sınır	[0, 20]
MKC	Mesafe Küçük Çok	[10, 30]
MKB	Mesafe Küçük Büyük	[20, 40]
MKO	Mesafe Küçük Orta	[30, 50]
MOK	Mesafe Orta Küçük	[40, 60]
MK	Mesafe Küçük	[50, 70]
M	Mesafe	[60, 80]
MB	Mesafe Büyük	[70, 90]
MOB	Mesafe Orta Büyük	[80, 100]
MBO	Mesafe Büyük Orta	[90, 110]
MBK	Mesafe Büyük Küçük	[100, 120]
MBC	Mesafe Büyük Çok	[110, 130]
MBS	Mesafe Büyük Sınır	[120, 140]

4.1.3. Üyelik İşlemleri

Altı robotun hem giriş hem de çıkış değişkenleri için bulanık alt kümeler, üçgen ve yamuk tipi üyelik işlevleri olarak tanımlanmıştır.

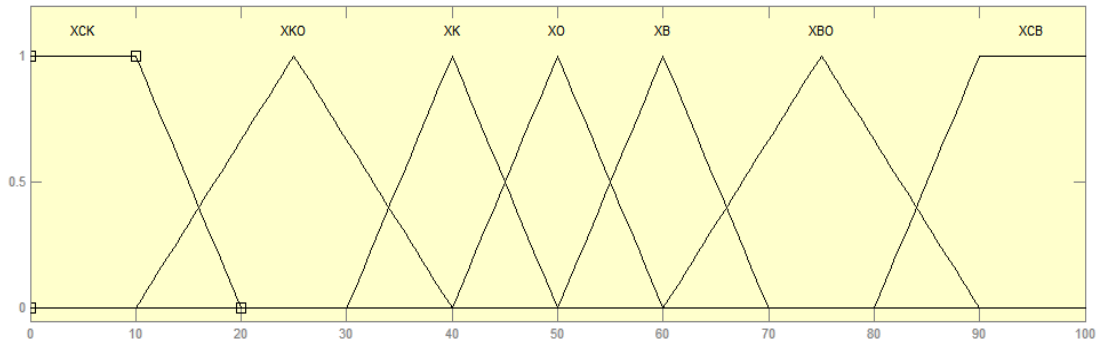
X Konum, Y konum ve Açılı değişkenlerinin değişim aralıkları;

$$0 < X \text{ konum} < 100$$

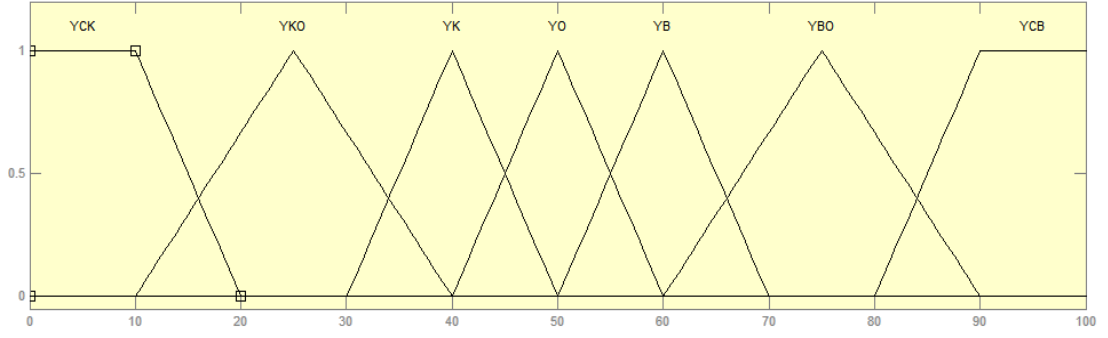
$$0 < Y \text{ konum} < 100$$

$$0 < \text{Yatay Açılı} < 90^\circ \text{ dır.}$$

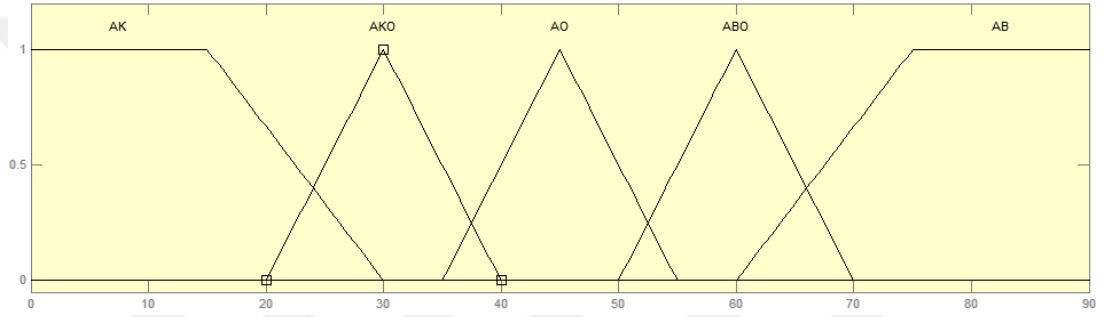
Bu aralılara göre üyelik işlevleri X konum için Şekil 4.2’de, Y konum için Şekil 4.3’de ve açılı verisi için Şekil 4.4’de verilmiştir.



Şekil 4.2 X Konum Verisinin Bulanık Alt Kümeleri



Şekil 4.3 Y Konum Verisinin Bulanık Alt Kümeleri

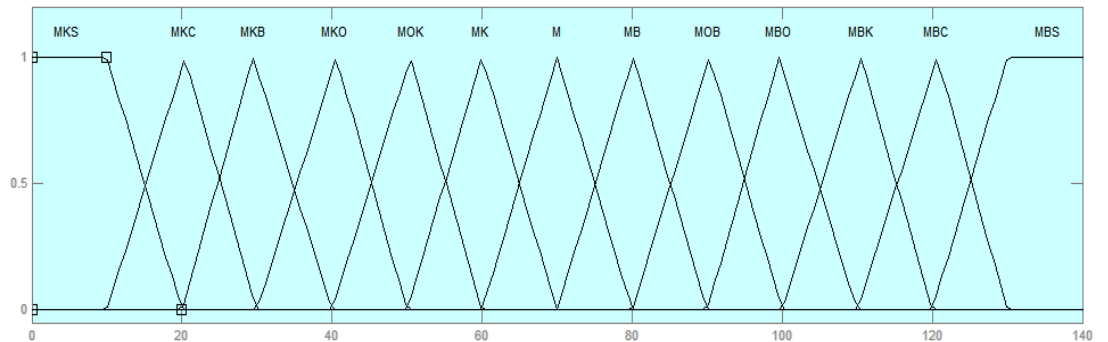


Şekil 4.4 Açık Verisinin Bulanık Alt Kümeleri

Çıkış değişkeni olan mesafe verisinin değişim aralıkları;

$0 < \text{Mesafe} < 140$ 'dır.

Bu aralılara göre üyelik işlevleri Şekil 4.5'de verilmiştir.



Şekil 4.5 Mesafe Verisinin Bulanık Alt Kümeleri

4.1.4. Bulanık Çıkarım

Bulanık kurallar bulanık mantık denetleyici biriminin en önemli bölümüdür. Çünkü bu birimde Altu robotun bilgi tabanı ve karar verme yeteneği oluşturulmaktadır. Bulanık muhakemeyi oluşturmak için birçok teknik mevcuttur.

Herhangi bir giriş için, her kural belli bir ağırlık derecesi ile kullanılmaktadır. İşi gerçekleştirmek için Altu robot üzerinde Max-Min (Mamdani) bulanık çıkarımı seçilmiştir. Bu yöntemde çıkıştaki bulanık küme girişteki kümelerin mantıksal ve işlemine tabi tutulması sonucunda ortaya çıkar. Çıkış sayısal değeri ağırlık ortalaması metoduyla belirlenmektedir.

4.1.5. Bulanık Kuralların Belirlenmesi

Bu tez çalışmasında her bir koordinat (x,y), açı (°) ve mesafe (m) değişkeninin sözel ifadesi ve bulanık ifadelerinin ilişkilerinin gösteren bulanık kurallar 141 tane belirlenmiştir. Üç giriş ve tek çıkış bulunan sistemimizde 2 giriş değişkeni için 7 adet üyelik fonksiyonu, bir giriş değişkeni için 5 adet üyelik fonksiyonu ve çıkış değişkeni için 13 adet üyelik fonksiyonu tanımlanmış bulanık sistemde her koşul olası olduğunda 245 adet kural tanımlanmaktadır. Ancak bu kurallar içerisinde gerçek hayatta oluşması mümkün olmayan durumlar mevcuttur. Bu durumlar kural tablosundan çıkartılabilir.

Mesafe için;

Kural 1. Eğer XCK ve YCK ve AK ise mesafe değeri MKS

.

.

Kural 141. Eğer XCB ve YCB ve AB ise mesafe değeri MBS

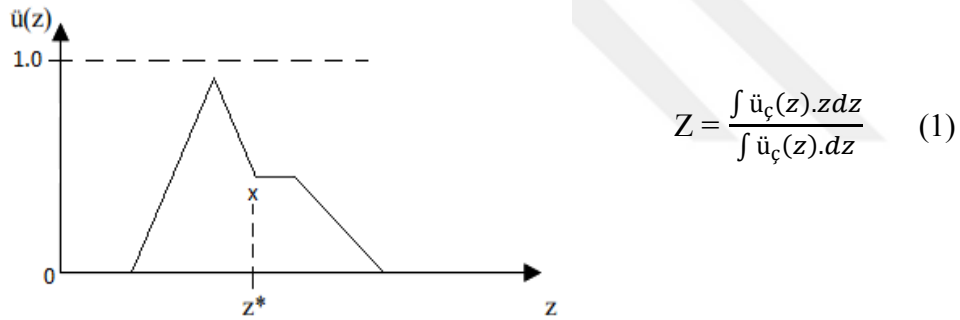
4.1.6. Durulama

Altu robotun bulanık mantık denetleyici biriminin çıkışından alınan bilgi bulanık bir bilgidir. Bu bulanık değerın kesin değere dönüştürmek için durulama işleminden geçirilmesi gerekmektedir.

Önce kullanılan her kural için üyelik değerlerinden oluşan her bir bulanık çıkış kümesi, çıkış evrensel kümesinde tespit edilmektedir. Daha sonra bu kümeler tarafından oluşturulan mantıksal birleşim kümesi üzerinde durulama yöntemlerinden birisi kullanılır ve tek çıkış değeri bulunarak durulama işlemi yapılmış olur. Elde edilen değer bulanık mantık denetleyicinin sisteme uygulayacağı çıkış değeridir [43].

Altı robotun bulanık çıkarımından gelen veriler durulama bölümünde bulanık mantık üzerinde yaygın olarak kullanılan ağırlık merkezi yöntemini kullanmaktadır. Oluşturulan bulanık çıkarım kümesi şeklinin ağırlık merkezi bulunmaktadır ve kesin değer olarak bu merkeze karşılık gelen değer alınmaktadır.

Robotun bulanık mantık denetim kuralları 245 adettir ve kullanılan 141 adettir. Şekil 4.6'da ağırlık merkezi yöntemi ile durulaştırmanın matematiksel işlem grafiği ve bağlantı (1) ile formülü verilmiştir.

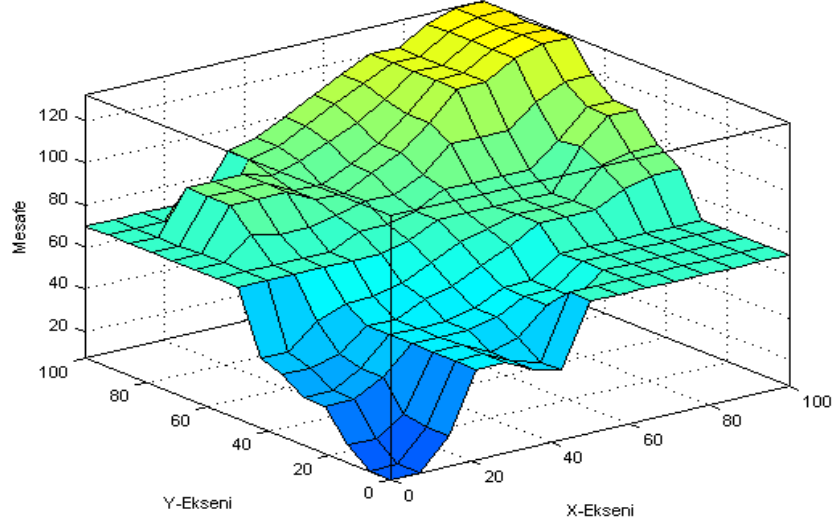


Şekil 4.6 Ağırlık Merkezi Yöntemi ile Durulaştırma

4.1.7. Matlab Fuzzy Toolbox ile Oluşturulan Sistemin Çıktısı

Matlab Fuzzy Toolbox, Bulanık mantık arayüzü kullanılarak bir sistemin Matlab üzerinde oluşturulması, düzeltilmesi, değiştirilmesi ve analizini grafiksel olarak yapabilmesini sağlayan bir araç kutusudur. Bu araç kutusu üzerinde beş temel araç söz konusudur. Bunlar: bulanık çıkarım sistemi, kural editörü, üyelik fonksiyonu editörü, kural görüntüleyici ve yüzey görüntüleyici bölümleridir.

Altu robot için oluşturulan bulanık sistemin yüzey görüntüsü Şekil 4.7’de verilmiştir.



Şekil 4.7 Oluşturulan Sistemin Matlab Çıktısı

4.2. Mikrodenetleyici Yazılımın Yapısı

Altu robotun mikrodenetleyici yazılımı beş aşamadan oluşmaktadır. Bu aşamalar;

- Ultrasonik
- Kablosuz haberleşme birimi
- Motor sürücü ünitesi
- Kontrol algoritması mikrodenetleyici yazılımının bölümleridir.

4.2.1. Ultrasonik sensör

Altu robot üzerinde dört adet HC-SR04 Ultrasonik sensör kullanılmıştır. Bu sensörlerin dizilimi Şekil 3.13’de verilmiştir. Altu robot üzerinde kullanılan sensörde 4 adet pin bulunmaktadır. Bunlar; Vcc, Gnd, Trig, Echo pinleridir. Trig pininden yaklaşık 10 μ s’lik bir puls gönderilmektedir ve sensör kendi içerisinde 40 KHz frekansında bir sinyal üretilip verici ünitesine göndermektedir. Bu ses sinyali bir nesneye çarpar ve geri dönerek yansır. Nesnenin sensörden uzaklığı ile doğru orantılı olarak echo pini bir süre lojik 1 olur ve sonra lojik sıfıra düşer. Kısaca sensörün çalışma mantığı lojik 1 olma süresini bulmaktır. Sensörün kontrolüne ilişkin yazılım kodlaması EK-10’da verilmiştir.

4.2.2. Kablosuz haberleşme sensörü

Altu robotu slave modda, PC yazılımı da Master modda iletişim protokolünü kurduktan sonra yazılımlar arası haberleşme başlamaktadır. Bluetooth modülün iletişime başlamasında iletişimin güvenli olması için otomatik olarak şifre istenmektedir. Bu şifre “1234”dir. Son eşlenen slave cihazı hatırlamak için herhangi bir özelliğe sahip değildir. Yani herhangi bir cihaz ile de eşleşebilmektedir. Bunun yazılımı yazarken dikkate alınması gerekmektedir. Sensörün temel yazılımı EK-11’de verilmiştir.

4.2.3. Hareket ünitesi

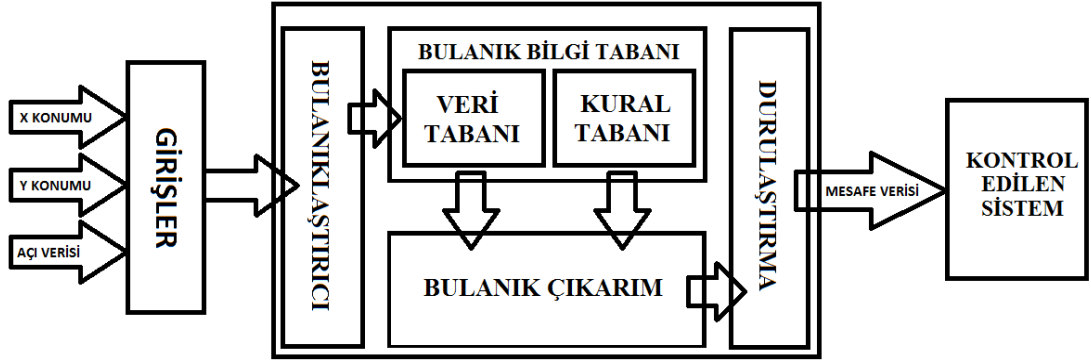
Altu robotun hareket kontrol ünitesini oluşturan L298 entegresi, yüksek akıma karşı dayanıklı ve standart TTL lojik seviyelerde (0-5V) kontrolü sağlanabilmektedir. Altu robot üzerinde L298 motor sürücü arabirimi bir adet kullanılmıştır ve teknik özellikleri incelendiğinde; sürücü üzerinde 2 adet kanal bulunmaktadır. Her kanal 2A’e kadar akım çekebilmektedir.

Robot üzerinde 2 adet 12V 500 RPM’lik redüktörlü DC motor, 2 adet standart Servo motor ve 2 adet sarhoş teker kullanılmıştır. DC Motorların kontrolü, mikroişlemciden gelen kodların L298 Motor sürücü entegresi ile işlenip çıkış alınması ile oluşturulmuştur. DC motorların kontrollerini sağlayan temel kodlar EK-12’de verilmiştir.

Servo motorların kontrolünü mikroişlemciden gelen kodların PWM aralıkları ile ifade edilmesiyle oluşturulmuştur. Servo motorların kontrolünü sağlayan temel kodlar EK-13’de açıklamalarıyla birlikte verilmiştir.

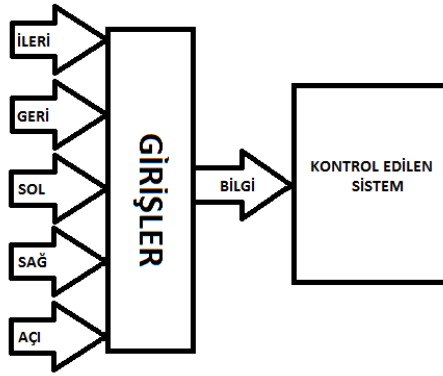
4.2.4. Kontrol algoritması

Tez çalışması için tasarlanan sistemin bulanık kontrol blok diyagramı Şekil 4.8’de verilmiştir. Bulanık kontrol algoritmasında 3 adet giriş kullanıcıdan alınıp bulanık bilgi tabanına gönderilip bulanıklaştırılmaktadır. Bu bilgiler 141 adet bulanık kural tabanı içerisinde taranıp çıkış değişkeni bulunup bulanık çıkarım yapılmaktadır. Daha sonra bu bilgi durulayıcı birimine gönderilmektedir. Durulayıcıdan çıkan mesafe bilgisi kontrol edilen sisteme gönderilmektedir. EK-14’de bulanık mantık tabanı için oluşturulmuş programın tanımlama ve kural yapısı verilmiştir.



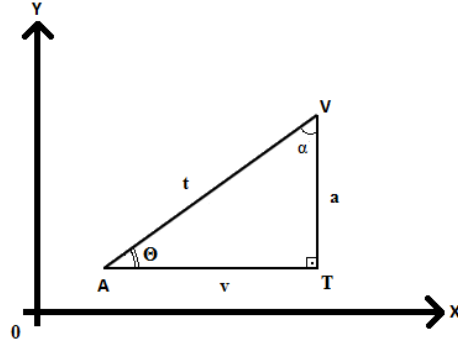
Şekil 4.8 Bulanık Kontrol Blok Diyagramı

Tasarlanan sistemin geleneksel kontrol blok diyagramı Şekil 4.9’da verilmiştir. Girişe uygulanan değişkenler alınıp kontrol edilen sisteme gönderilmektedir.



Şekil 4.9 Geleneksel Kontrol Blok Diyagramı

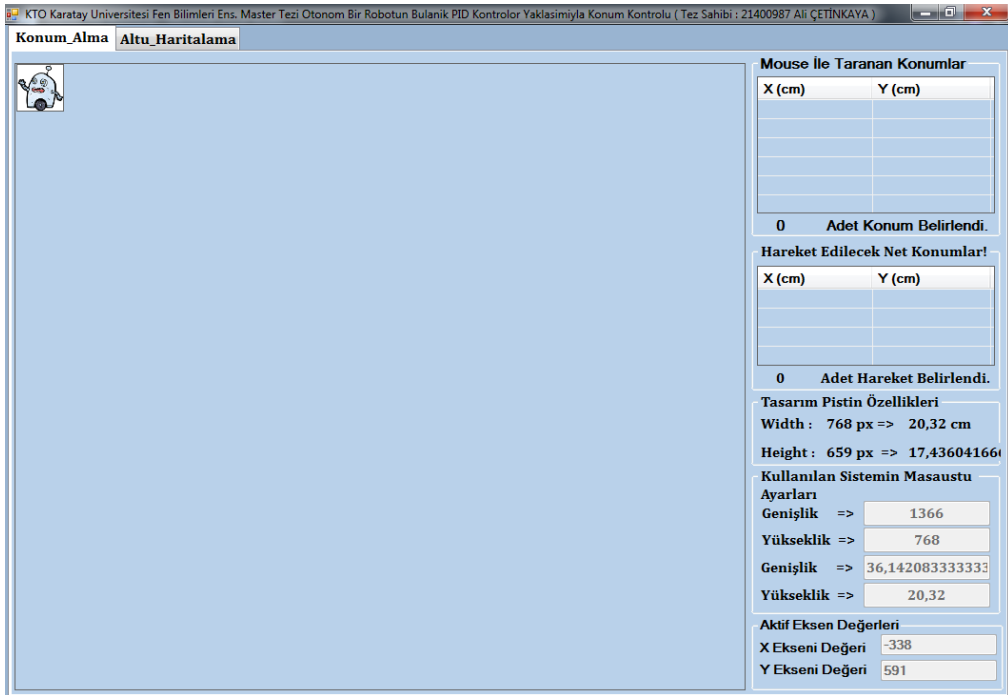
Tasarlanan sistemin mesafe hesaplama diyagramı Şekil 4.10’da verilmiştir. Diyagram üzerinde görülen A sembolü Altu robotun başlangıç konumunu, V sembolü robotun gideceği hedef noktayı ve T sembolü A ve V noktalarının X ekseninde kesiştiği noktayı temsil etmektedir. “ t ” sembolü Altu robotun başlangıç konumu ile hedef nokta arasındaki mesafeyi, “ a ” hedef noktanın X ekseninde yüksekliğini ve “ v ” sembolü ise başlangıç noktasının kesişim noktasına olan uzaklığını temsil etmektedir. Θ sembolü robotun başlangıç konumundan hedef noktasına ulaşması için kullanılan açı değişkeni, α sembolü $90 - \Theta$ ile ifade edilmektedir.



Şekil 4.10 Mesafe Hesaplama Diyagramı

4.3. PC Yazılımın Yapısı

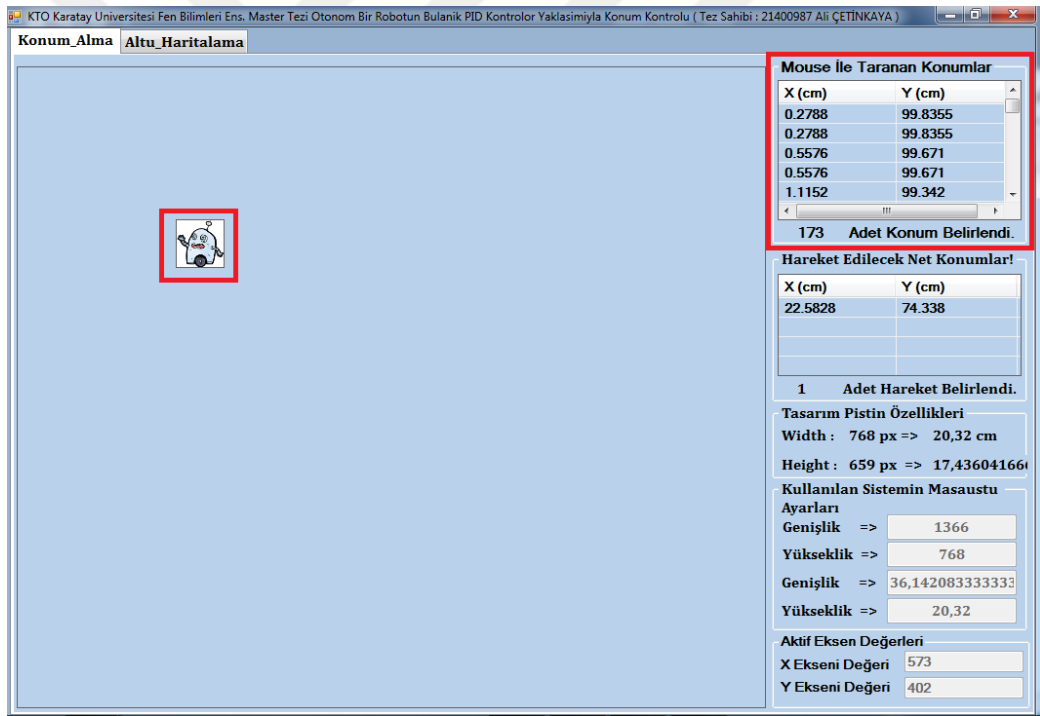
Tezin masaüstü yazılım bölümünün tasarlanmasında Visual Studio 2015 c# yazılımı kullanılmıştır. Tasarlanan yazılımın kullanım kolaylığı ve Robot ile sorunsuz iletişimi konusunda özel bir çaba harcanmış, her bölüm kendi içerisinde ayrı ayrı ele alınarak denemelerden sonra gerekli düzeltme ve sadeleştirmeler yapılmıştır. Masaüstü yazılımının oluşturulmasında iki bölüm söz konusudur. Birinci bölüm konum alma bölümü, ikinci bölüm ise alınan konum ile işlemlerin yapıldığı ve değerlerin robota aktarıldığı Altu Haritalama bölümüdür. Şekil 4.11'de C# yazılımında oluşturulan yazılımının konum alma ekranı verilmiştir.



Şekil 4.11 C# Yazılımında Oluşturulan Yazılımının Konum Alma Kısmı

Tasarlanan pist içerisinde robotun simgesi sol üst bölüme yerleştirilmiştir. Simgenin başlangıçta bulunduğu konum robotun pist içinde o anki konumunu etkilememektedir. Simge sol üste geçici olarak yerleştirilmiştir. Robotun simgesi farenin sol tuşuna basılı tutularak hareket ettirilir ve farenin bırakıldığı yer üzerinden hedef nokta belirlenerek işleme alınmaktadır.

Birinci bölümde Altu robotun simge ile gösterildiği alanda, fare ile kullanıcı tarafından robotun pist içerisinde gitmesi gereken koordinatlar belirlenmektedir. Robotun c# yazılımı üzerinde belirlenen alan içerisinde fare ile gittiği tüm noktaların gösterildiği alan “Mouse ile taranan konumlar” bölümüdür. Bu alan içerisinde Mouse hareketi ile gezilen konumların sayısı “... Adet konum belirlendi.” ile gösterilmektedir. Bu durum Şekil 4.12 ile verilmiştir.



Şekil 4.12 Oluşturulan Yazılımının Mouse ile Taranan Konumlar

Altu robotun alan içerisinde gezdirilmesiyle oluşturulan koordinatların farenin sol tık olayı ile belirlenen hedef nokta ve noktaların gösterildiği “Hareket Edilecek Net Konumlar” bölümüdür. Bu bölümde X ve Y koordinatları ayrı ayrı cm cinsinden ele alınmaktadır. Bu alan içerisinde farenin sol tık hareketi ile net şekilde belirlenen verilerin adeti bölümün sonunda “... Adet konum belirlendi.” bölümünde gösterilmektedir. Bu durum Şekil 4.13’da verilmiştir.

Hareket Edilecek Net Konumlar!	
X (cm)	Y (cm)
22.5828	74.338

1 Adet Hareket Belirlendi.

Şekil 4.13 Yazılımda Mouse ile Oluşturulan Net Konumlar

Ayrıca robot alanının değerleri Şekil 4.14’de gösterildiği gibi “Tasarlanan pistin özellikleri” kısmında gösterilmiştir. Yazılım üzerindeyken farenin masaüstü yazılımında oluşturulan harita üzerinde herhangi bir nokta bulunduğu değerlerinin gösterildiği yer Şekil 4.14’de gösterildiği gibi “Aktif Eksen değerleri” bölümüdür. Alandaki bu değerler piksel cinsinden gösterilmiştir.

KTO Karatay Üniversitesi Fen Bilimleri Ens. Master Tezi Otonom Bir Robotun Bulanık PID Kontrolör Yaklaşımıyla Konum Kontrolü (Tez Sahibi : 21400987 Ali ÇETİNKAYA)

Konum_Alma Altu_Haritalama

Mouse ile Taranan Konumlar

X (cm)	Y (cm)
0.2788	99.8355
0.2788	99.8355
0.5576	99.671
0.5576	99.671
1.1152	99.342

173 Adet Konum Belirlendi.

Hareket Edilecek Net Konumlar!

X (cm)	Y (cm)
22.5828	74.338

1 Adet Hareket Belirlendi.

Tasarım Pistin Özellikleri

Width : 768 px => 20,32 cm

Height : 659 px => 17,43604166

Kullanılan Sistemin Masaüstü Ayarları

Genişlik => 1366

Yükseklik => 768

Genişlik => 36,142083333333

Yükseklik => 20,32

Aktif Eksen Değerleri

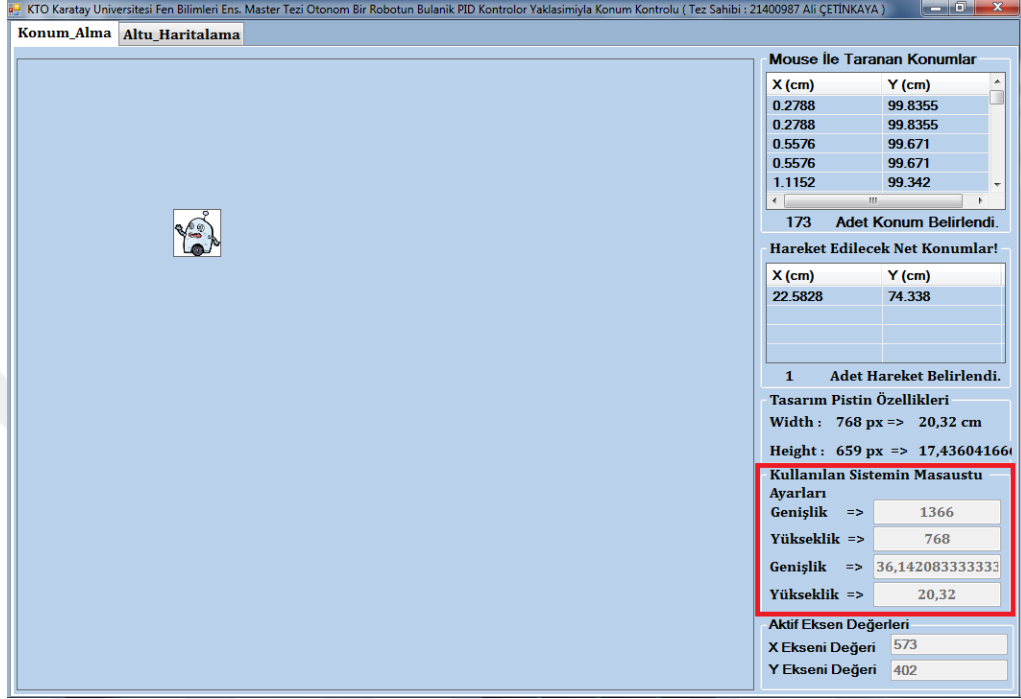
X Eksen Değeri 573

Y Eksen Değeri 402

Şekil 4.14 Yazılımda Oluşturulan Pistin Özellikleri ve Aktif Eksen Değerleri

Tezin masaüstü yazılımında, kullanıcıdan herhangi bir pist uzunluğu, masaüstü yazılımının kullanıldığı monitör büyüklüğü ve mevcut ekran çözünürlüğü gibi haritalama sürecinde büyük önem taşıyan değerlerin otomatik olarak alınması sağlanmıştır.

Bu anlatılan veriler c# yazılımında Altu Haritalama kısmında “Kullanılan Sistemin Masaüstü Ayarları” bölümünde gösterilmiştir. Bu durum Şekil 4.15’de verilmiştir.



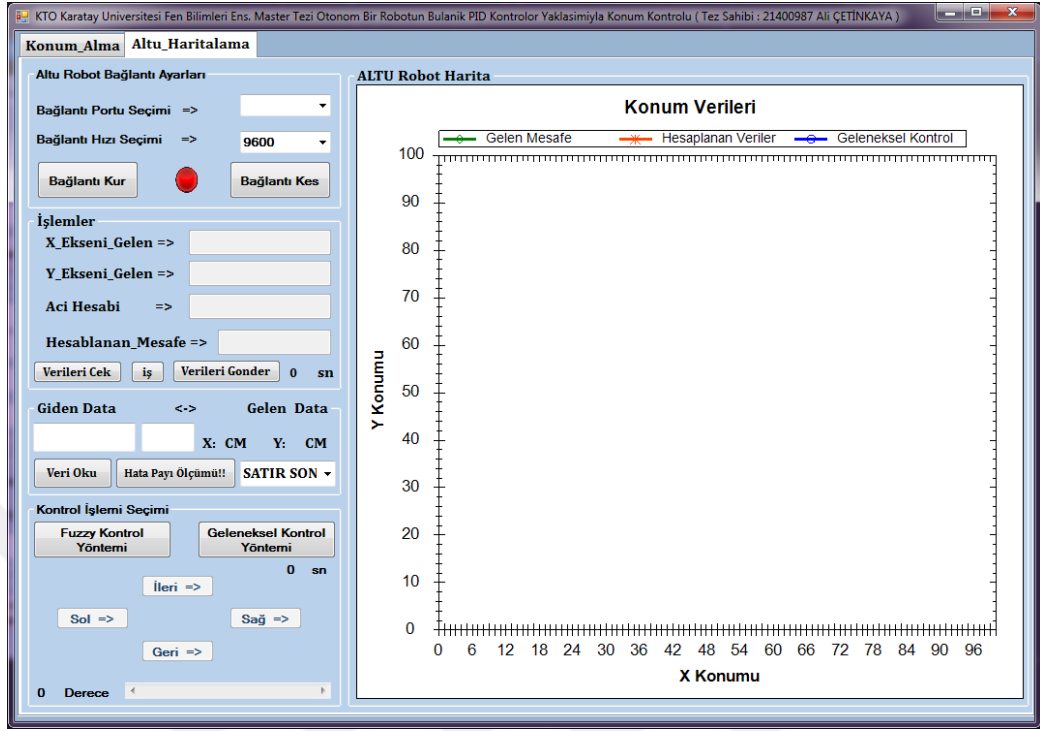
Şekil 4.15 Kullanılan Sistemin Masaüstü Ayarları

Tezin hazırlandığı bilgisayarın masaüstü ayarları genişlik 1366 piksel yükseklik 768 piksel olarak ele alınmıştır. Piksel değerlerinin Altu robota aktarılması santimetre üzerinden yapılmıştır. Santimetre olarak oluşturulan değerlerin piksel ve inç karşılıkları aşağıdaki gibidir:

$$1 \text{ inç} = 96 \text{ px} = 2.54 \text{ cm}$$

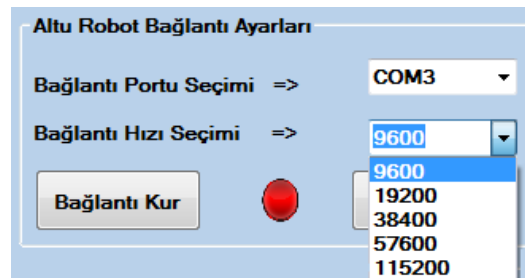
$$1080 \text{ px} = 28.6 \text{ cm' dir.}$$

Masaüstü yazılımının ikinci kısmını oluşturan haritalama aşaması, konum alma bölümünden gelen verilerin işleme alındığı, harita üzerinde Altu robotun hareketlerinin izleneceği ve kablosuz iletişim protokollerinin başlatılacağı kısımdır. Şekil 4.16’da haritalama, bağlantı ayarları, işlemler, giden data -gelen data ve kontrol işlemi seçiminin bulunduğu bölüm verilmiştir.



Şekil 4.16 C# Yazılımında Oluşturulan Haritalama Kısmı

Bu aşamada ilk olarak kablosuz iletişim protokollerinin oluşturulmasıyla Altu robotun pist üzerindeki yerinin yazılıma alınmasıyla başlatılır. Bu durum “Altu Robot Bağlantı Ayarları” kısmında görülmektedir. Bu kısımda bağlantı ayarları mevcuttur. Bunlar bağlantı protokol portu ve bağlantı hızının seçimidir. Bu işlemlerin yapıldığı pencere Şekil 4.17’de verilmiştir.



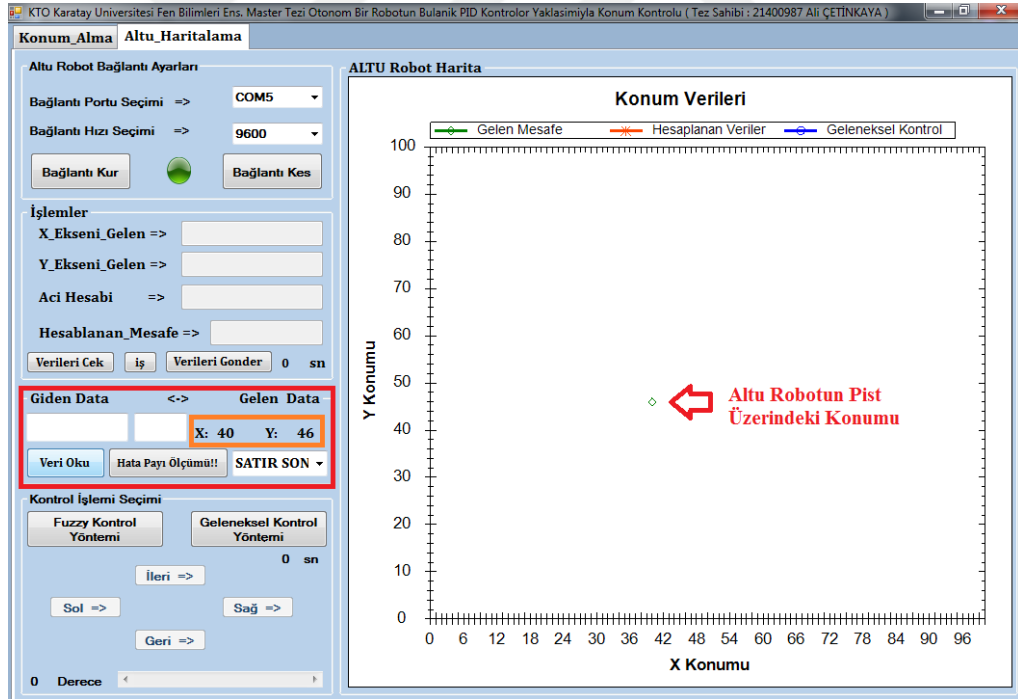
Şekil 4.17 C# Yazılımında Oluşturulan Altu Robot Bağlantı Ayarları

Bağlantı ayarlarında port seçimi ve bağlantı hızının seçiminde bağlantı başarılı ise kullanıcıya yanıt olarak bu bölümde bulunan kırmızı renkli durum ikonu yeşil renge dönmektedir. Bağlantının başarılı olmasına ilişkin ekran görüntüsü Şekil 4.18’de verilmiştir.



Şekil 4.18 Başarılı Bağlantı Durumu

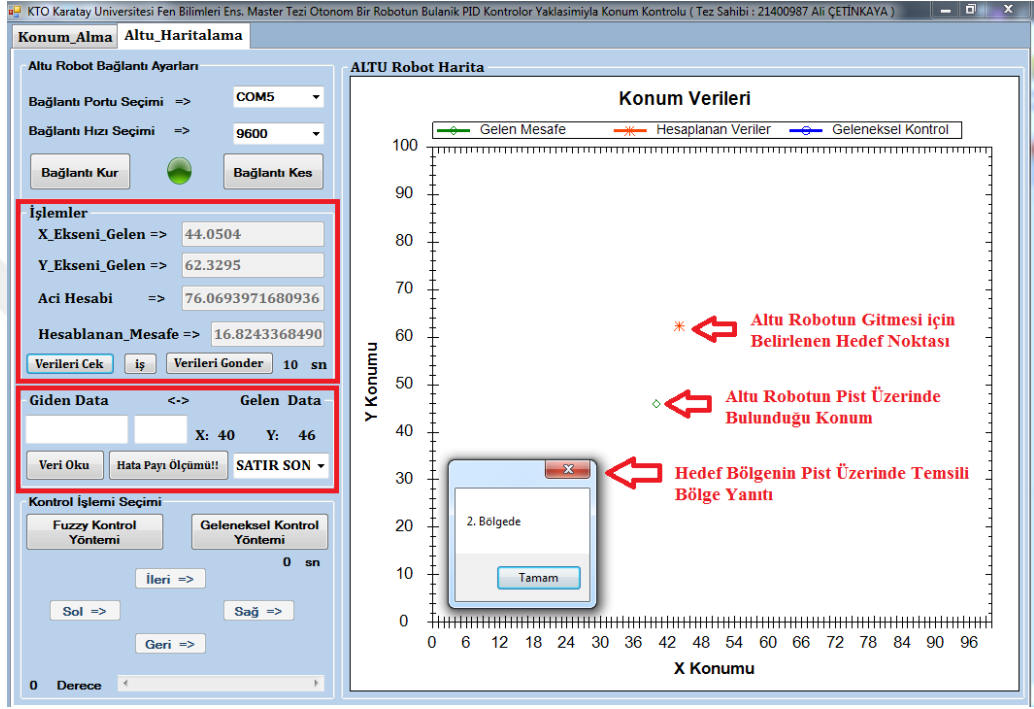
Başarılı şekilde kurulan iletişim ile Altu robotun pist üzerindeki konumu Şekil 4.19’da görülen “Giden ve Gelen Data” bölümü ile okunmaktadır. Koordinat okuma işlemi “Veri Oku” butonu ile gerçekleştirilmektedir. Okunan X eksen koordinatı ve Y eksen koordinatı Şekil 4.19’daki gibi “Giden ve Gelen Data” bölümünde sayısal çıktı olarak gösterilmiştir.



Şekil 4.19 Altu Robotun Pist Üzerindeki Konumu

Altu robotun iletişim protokolleri kurulduktan sonra yazılım içerisinde konum alma kısmında oluşturulan verilerin harita üzerine aktarılması gerekmektedir. Bunun için c# yazılımında Haritalama kısmının işlemler bölümü kullanılır. Bu bölümde “Verileri Çek!!” butonu ile konum verilerinin harita üzerinde gösterilmesi sağlanmaktadır.

MessageBox ile verinin o an nerede olduğu gösterilir. Harita üzerinde turuncu renkli yıldız şekliyle gösterdiğimiz yerde robotun gitmesini istediğimiz hedef nokta, yeşil renkli baklava dilimi şekliyle gösterilen yer ise robotun pist üzerinde o an bulunduğu konumu göstermektedir (Şekil 4.20).



Şekil 4.20 Altu Robotun Pist Üzerindeki Konumu ve Hedef Noktasının İşareti

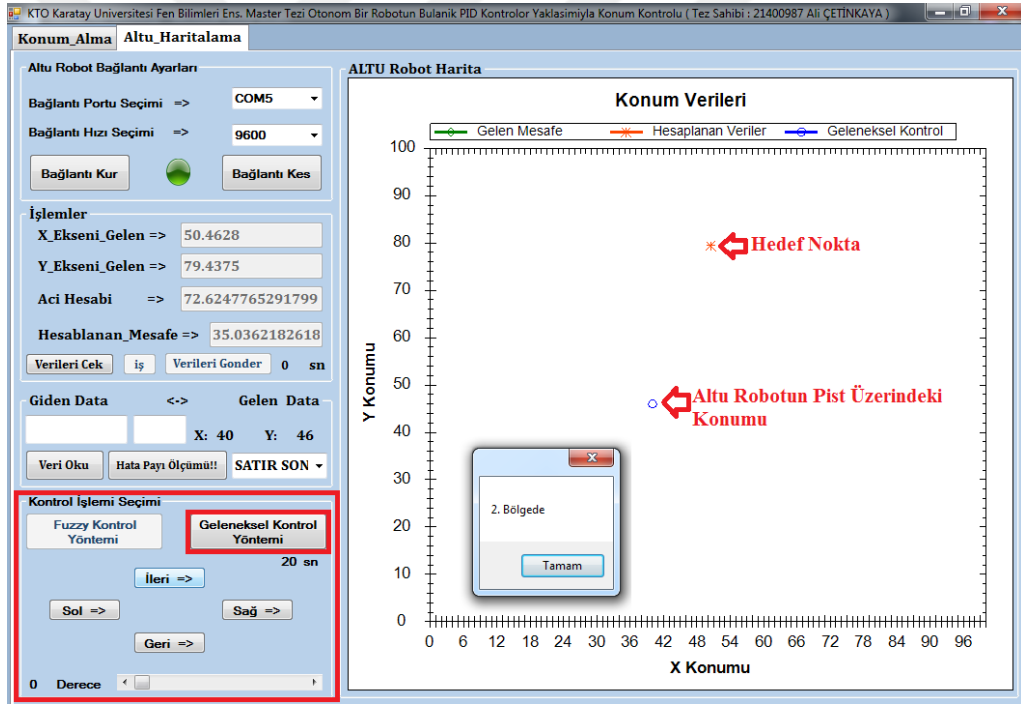
Konum verilerin (x, y) Altu robot üzerine aktarılması işlemler bölümünde yer alan “Verileri Gönder” butonu ile sağlanır. İşlemler bölümünde bulunan X_ekseni_gelen, Y_ekseni_gelen, Aci_hesabi ve Hesaplanan mesafe verileri robotun gitmesi gereken koordinat, açı ve mesafesini matematiksel fonksiyonlar ile hesaplanmasıyla oluşturulmuştur. Bu verilerden X_ekseni_gelen, Y_ekseni_gelen ve Aci_Hesabi verileri Altu robota kablosuz iletişim ile gönderilmektedir.

Altu robotun kontrol işlemindeki kullanılan masaüstü yazılımı geleneksel kontrol işlemi içinde kullanılmaktadır. Bu kontrolde Altu robot manuel olarak hareket ettirilmiştir. Geleneksel kontrol yöntemi tasarlanan masaüstü yazılımının haritalama bölümünde bulunmaktadır ve Şekil 4.21’de verilmiştir.



Şekil 4.21 Geleneksel Kontrol Yöntemi

Geleneksel kontrol yönteminde Altu robot çalıştırılırken robot üzerinden gelen veriler harita üzerinde mavi renkli daire şeklinde (Şekil 4.22) gösterilmiştir.

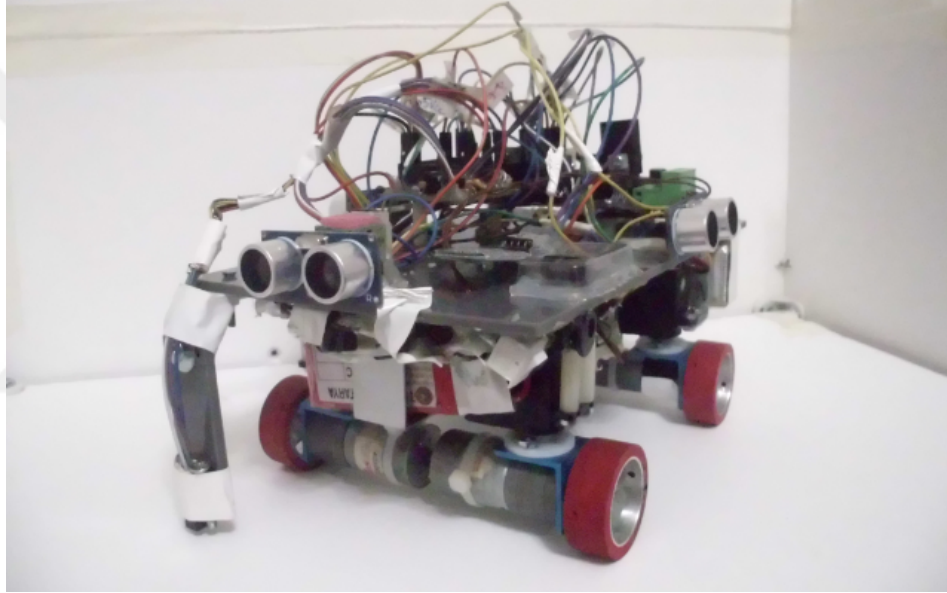


Şekil 4.22 Geleneksel Kontrol Yöntemi ile Altu Robotun Pist Üzerinde Konumu

Tez çalışması için oluşturulan masaüstü yazılımında haritalama bölümde iki farklı çizim olacaktır. İlk çizim Fuzzy ile yönlendirilen ve başlangıç noktasının yeşil renk ile gösterilmiş olan robotun gitmesi gereken koordinat işlemlerinin turuncu işaretlerden oluştuğu kısımdır. İkinci çizim Geleneksel kontrol yöntemi ile yönlendirilen ve başlangıç noktasının mavi renk ile gösterilmiş olan robotun gitmesi gereken koordinat işlemlerinin turuncu işaretlerinde oluştuğu bölümdür.

4.4. Tasarlanan Robot ve Çalışma Alanı

Yapılan çalışmalar sonucunda tez için tasarlanan Altu robotun son görünümü Şekil 4.23’de verilmiştir. Altu robotun çalışma pisti içerisinde çevre ile olan etkileşimi ultrasonik HC-04 mesafe sensörleri ile sağlanmıştır. Robot üzerinde kullanılan ultrasonik mesafe sensörlerinin dezavantajı çalışma alanının diğer algılayıcı sensörlere göre kısıtlı kalmasıdır. Sensörün çalışma alanı yazılım ile 400 cm’ye kadar arttırılıp 1cm’ye kadar da düşürülebilmektedir. Altu robotta bu sensörlerin çalışma alanı 100 cm ile sınırlandırılmıştır.



Şekil 4.23 Altu Robotun Son Görünümü

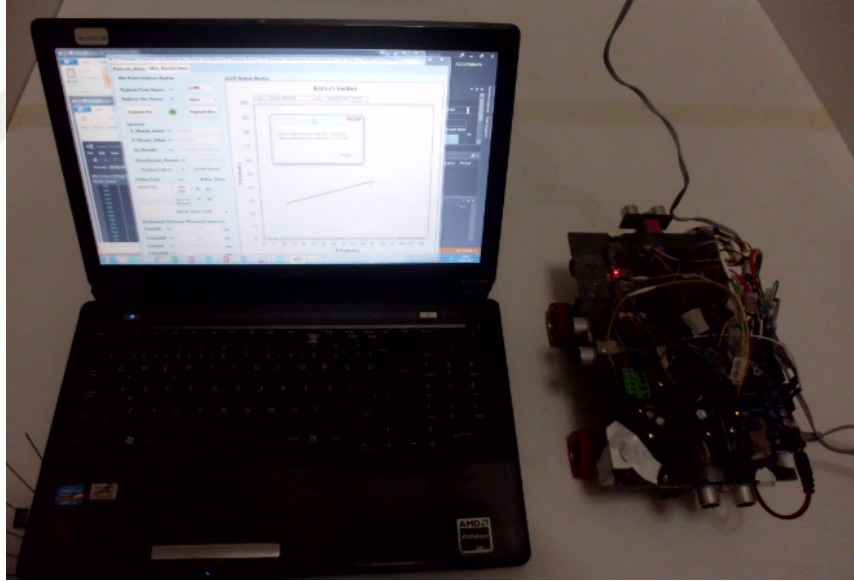
Altu robot için oluşturulan pistin yapısı strafor, pist engelsiz ve sabit ebatlarda 1 m² büyüklüğe sahiptir. 30 cm en – 50 cm boy uzunluğuna sahip ve 90 derece açı ile pistin kenarına tutturulmuş dört adet “L” parça ile pistin çevresi kapatılmıştır.

Pistin çevresinin kapatılmasının sebebi sensörlerden yansıyan ses dalgasının yüzeyle yaptığı açıyla ilişkilidir. Geliş açısı ne kadar ufak olursa, sesin yansıma yapmadan yüzeyi sıyrması ihtimali o kadar yükselir ve bu şekilde hatalı bir ölçüm sonuçları alınmaktadır. Kaygan yüzeyler, yansıtıcı özellikleri ile bu sorunun büyümesine yol açmaktadır. Şekil 4.24’de bu tezde geliştirilmiş gösterilen Altu robotun çalışma alanı gösterilmiştir.



Şekil 4.24 Altu Robotun Çalışma Alanı

Şekil 4.25’de Altu robotun pist içerisindeki görünümü verilmiştir.



Şekil 4.25 Altu Robotun pist içerisinde görünümü

5. UYGULAMALAR

Bu çalışmada, bulanık kontrolör yaklaşımı ile sabit pist ortamında otonom bir robotun konum kontrolünün uygulaması gerçekleştirilmiştir. Ayrıca sonuçlar içerisinde geleneksel kontrol yöntemi de tasarlanmış ve denemeleri yapılmıştır.

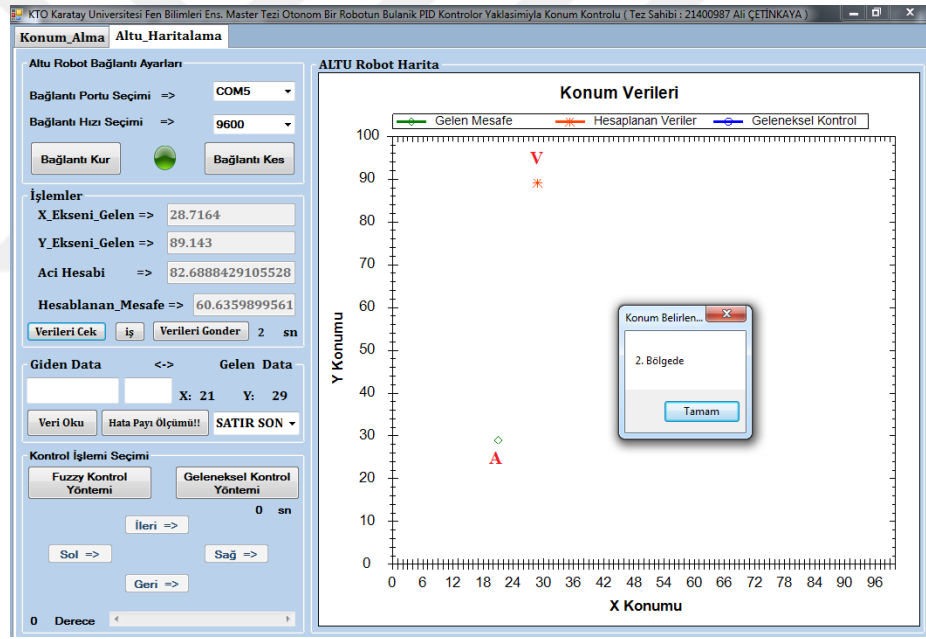
Bulanık kontrolör yaklaşımı ile robotun test ortamında rahatlıkla hareket ettiği ve istenilen hedeflere gittiği görülmüştür. Sonuçları daha ayrıntılı işleyebilmek için farklı senaryolar oluşturulmuş ve senaryo sonuçları masaüstü yazılımı üzerinde izlenmiştir.

Altu robotun Bulanık kontrolör algoritması ile robotun güç harcamasını minimuma çekmek amaçlanmış ve robotun gitmesi gerektiği koordinat ile işlemler tamamlanınca gittiği koordinat arasındaki verilerinin hata payları ve işlem süresi için harcanan güç sarfiyatı ayrı ayrı hesaplanarak kullanıcıya yüzdeler dilim olarak masaüstü yazılımı üzerinden gösterilmiştir. Bulanık kontrolör algoritması ile Altu robota esnek çalışabilme yetenekleri ve kazandırılmıştır. Masaüstü yazılımı üzerinde robotun çalıştırıldığı pist oluşturulmuş, robotun hareketleri ve kullanıcıya giden veriler çizimsel olarak gösterilmiş ve test sonuçları hata payı ölçümü olarak gösterilmiştir.

Geleneksel kontrol yönteminde ise robotun enerji sarfiyatı artmış, işlem zamanı süresi artarak pist üzerinde gitmesi gereken mesafe bilgisini operatörden aldığı komutlar ile gerçekleştirilmiştir. Komut parçalarının artması robot üzerinde dengesiz hareketlere ve hedefe varmada sorun oluşturmuştur. 10 milisaniye hareket ile robotun hareketleri her kontrolör yönteminde denenmiş ve senaryolar ayrı ayrı sonuçlandırılmıştır.

5.1. Birinci Uygulama

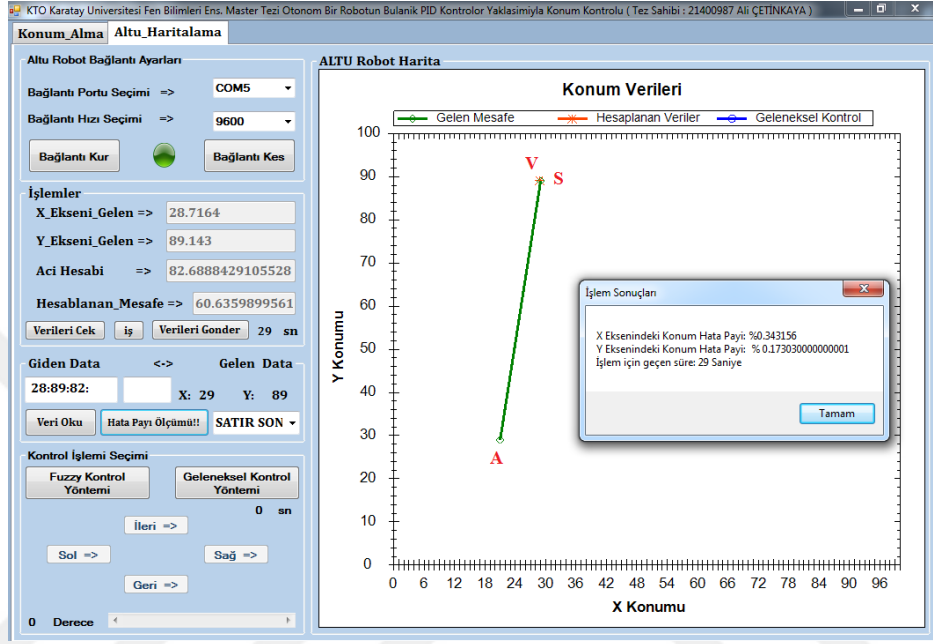
Birinci uygulamada Altu robot pist içinde başlangıç konumu X: 21 cm ve Y: 29 cm yerleştirilmiştir. Konum alma sekmesinde hedef nokta olarak X: 28 cm ve Y: 89 cm belirlenmiştir. Kontrol işlemi seçiminde “Fuzzy kontrol yöntemi” seçilerek yazılım üzerinde veri oku butonu ile Altu robotun pist üzerindeki koordinatları harita üzerinde yeşil renkli baklava dilimi olarak belirtilmiştir. Hedef konumu ise “Verileri Çek” butonuna basıldığında turuncu yıldız şeklinde harita üzerinde belirtilmiştir. Şekil 5.1’de Altu robotun birinci uygulama için program görüntüsü verilmiştir. Şekil üzerinde belirtilen “A” başlangıç konumu olarak, “V” robotun ulaşması gereken hedef nokta olarak gösterilmiştir.



Şekil 5.1 Birinci Uygulamanın Senaryo Haritası

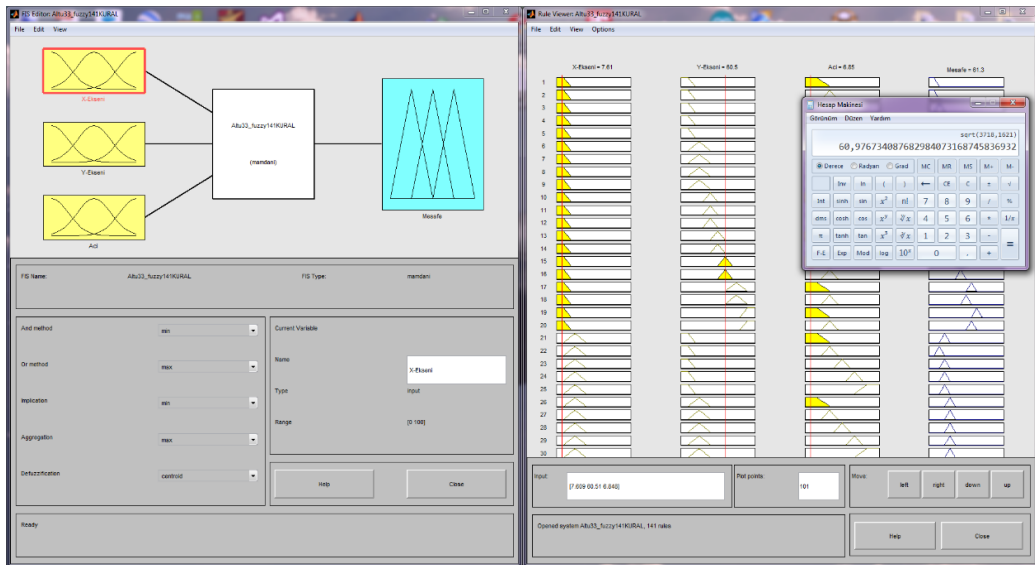
Birinci uygulama sonucunda Altu robot hedef nokta olarak belirlenen X: 28 cm ve Y: 89 cm koordinatına gönderilmiştir. Varılan konumun işlemsel doğruluğu için “Hata payı ölçüm” butonuna basılmalıdır. Açılan “İşlem sonuçları” penceresinden X eksenindeki hata payı, Y eksenindeki hata payı ve Altu robotun harita üzerinde belirtilen “A” noktasından “V” noktasına hareketi esnasında geçen işlem süresi ölçümü verilmiştir. “S” işareti ise Altu robotun harita üzerinde yapılan ölçümler sonucu son konumu olarak gösterilmiştir. Birinci uygulama için ölçülen işlem değerleri ve rota sonucu Şekil 5.2’de verilmiştir.

İşlem sonucu X ekseninde 1cm ve Y ekseninde 0cm konum hata payı ile robot 60.53cm mesafe gitmiştir. Bu sonuca göre X Eksenindeki konum hata payı %0.345156, Y eksenindeki konum hata payı %0.173030 olarak hesaplanmıştır ve işlem için geçen süre 29 saniye olarak ölçülmüştür.



Şekil 5.2 Açıklanan Birinci Uygulamanın Sonuç Haritası

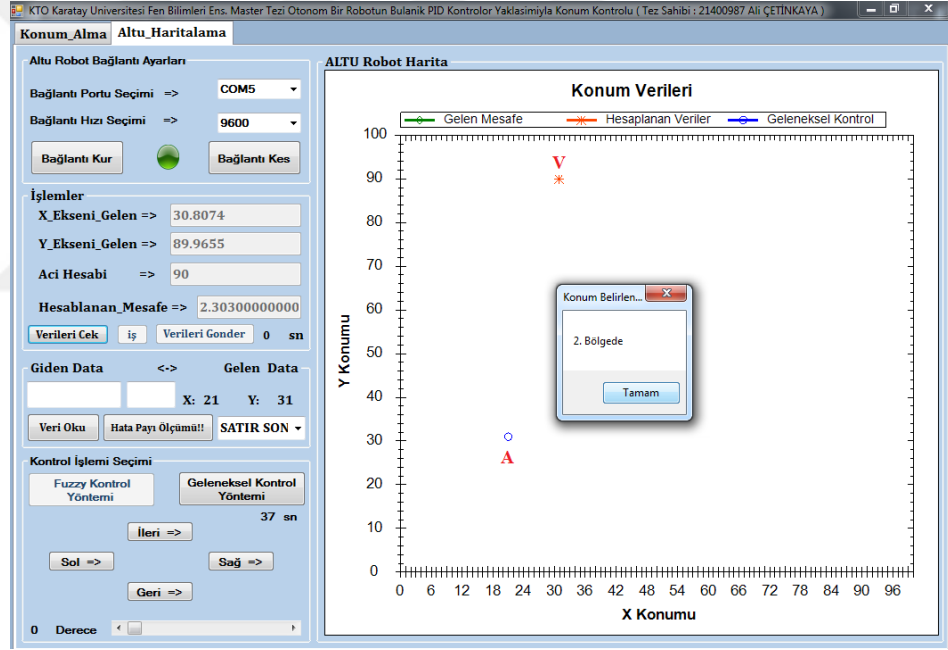
Şekil 5.3'de Matlab programında denemesi yapılan birinci uygulamanın sonucu verilmiştir.



Şekil 5.3 Açıklanan Birinci Uygulamanın Matlab Sonuç Haritası

5.2. İkinci Uygulama

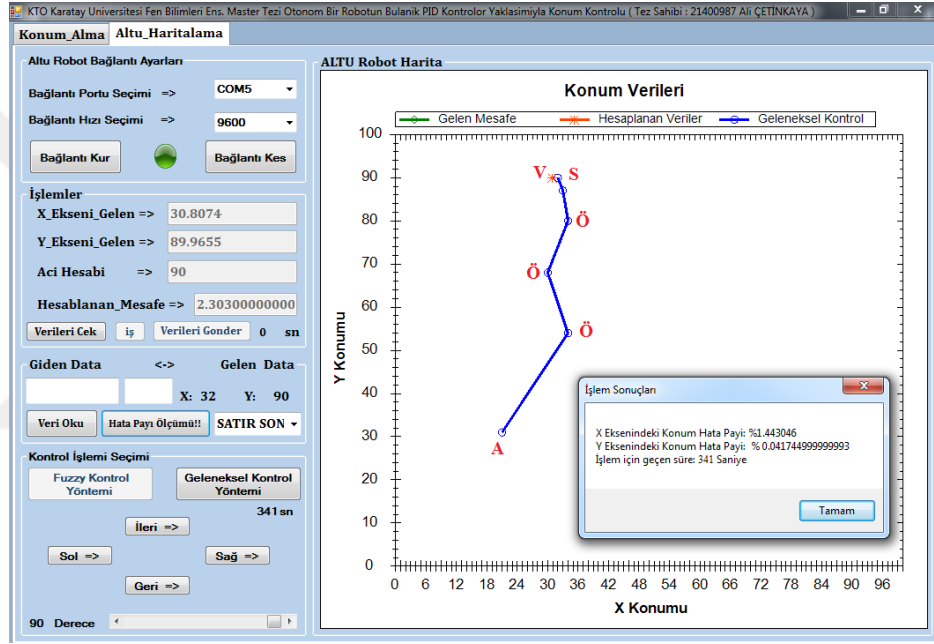
İkinci uygulamada Altu robot pist içinde başlangıç konumu X: 21 cm ve Y: 31 cm yerleştirilmiştir. Konum alma sekmesinde hedef nokta olarak X: 30 cm ve Y: 89 cm belirlenmiştir. Kontrol işlemi seçiminde “Geleneksel” kontrol yöntemi seçilerek yazılım üzerinde veri oku butonu ile Altu robotun pist üzerindeki koordinatları harita üzerinde mavi renkli daire olarak belirtilmiştir. Hedef konumu ise “Verileri Çek” butonuna basıldığında turuncu yıldız şeklinde harita üzerinde belirtilmiştir. Şekil 5.4’de Altu robotun ikinci uygulama için program görüntüsü verilmiştir. Şekil üzerinde belirtilen “A” başlangıç konumu olarak, “V” robotun ulaşması gereken hedef nokta olarak gösterilmiştir.



Şekil 5.4 İkinci Uygulamanın Senaryo Haritası

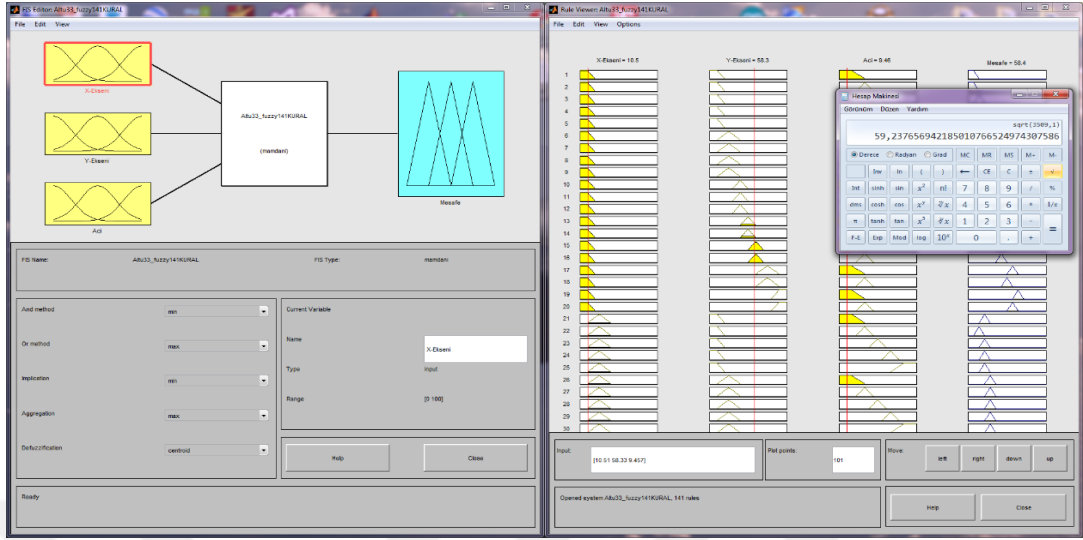
İkinci uygulama sonucunda Altu robot hedef nokta olarak belirlenen X: 30 cm ve Y: 89 cm koordinatına gönderilmiştir. Varılan konumun işlemsel doğruluğu için “Hata payı ölçüm” butonuna basılmalıdır. Açılan “İşlem sonuçları” penceresinden X eksenindeki hata payı, Y eksenindeki hata payı ve Altu robotun harita üzerinde belirtilen “A” noktasından “V” noktasına hareketi esnasında geçen işlem süresi ölçümü verilmiştir.

Şekil 5.5 üzerinde rota bilgisinin doğruluğu için hareket esnasında Altu robottan bulunduğu konumun verileri alınarak harita üzerinde “Ö” olarak gösterimi yapılmıştır. “S” işareti ise Altu robotun harita üzerinde yapılan ölçümler sonucu son konumu olarak gösterilmiştir. İkinci uygulama için ölçülen işlem değerleri ve rota sonucu Şekil 5.4’de verilmiştir. İşlem sonucu X ekseninde 2cm ve Y ekseninde 1cm konum hata payı ile robot 63.83cm mesafe gitmiştir. Bu sonuca göre X eksenindeki konum hata payı %1.443046, Y eksenindeki konum hata payı %0.041744 olarak hesaplanmıştır ve işlem için geçen süre 341 saniye olarak ölçülmüştür.



Şekil 5.5 Açıklanan İkinci Uygulamanın Sonuç Haritası

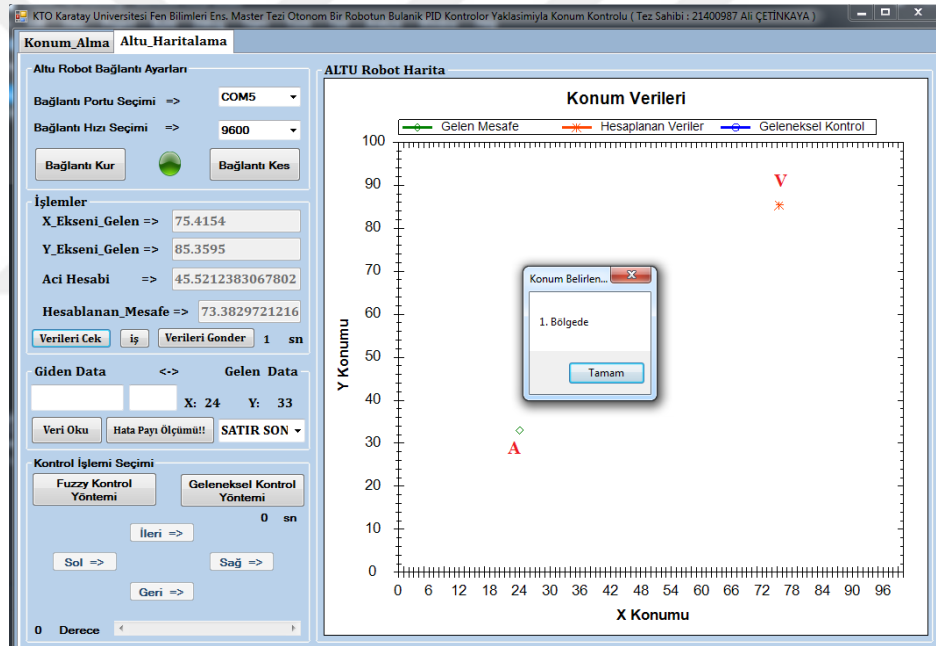
Şekil 5.6’da Matlab programında denemesi yapılan ikinci uygulamanın sonucu verilmiştir.



Şekil 5.6 Açıklanan İkinci Uygulamanın Matlab Sonuç Haritası

5.3. Üçüncü Uygulama

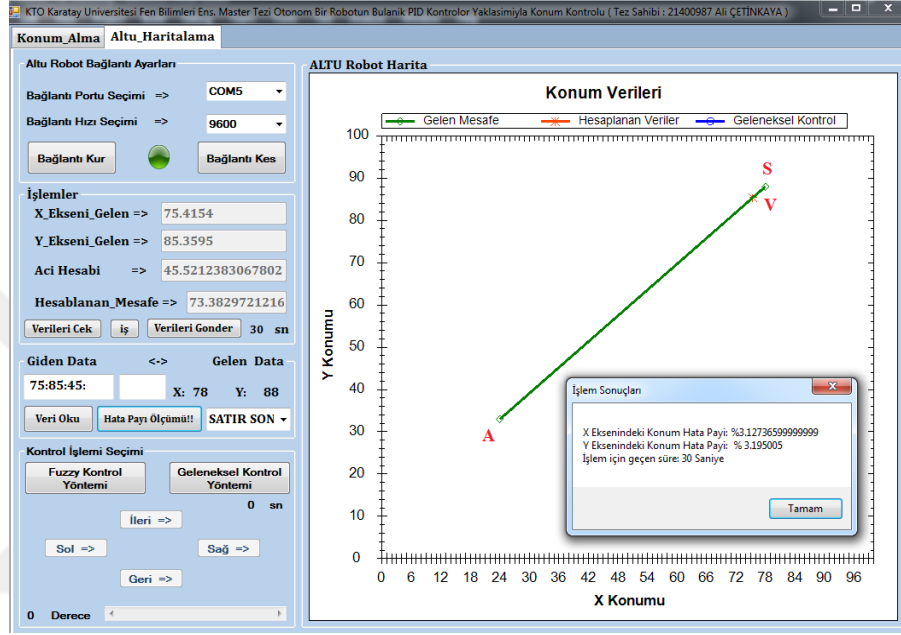
Üçüncü uygulamada Altu robot pist içinde başlangıç konumu X: 24 cm ve Y: 33 cm yerleştirilmiştir. Konum alma sekmesinde hedef nokta olarak X: 75 cm ve Y: 85 cm belirlenmiştir. Kontrol işlemi seçiminde “Fuzzy kontrol yöntemi” seçilerek yazılım üzerinde veri oku butonu ile Altu robotun pist üzerindeki koordinatları harita üzerinde yeşil renkli baklava dilimi olarak belirtilmiştir. Hedef konumu ise “Verileri Çek” butonuna basıldığında turuncu yıldız şeklinde harita üzerinde belirtilmiştir. Şekil 5.7’de Altu robotun birinci uygulama için program görüntüsü verilmiştir. Şekil üzerinde belirtilen “A” başlangıç konumu olarak, “V” robotun ulaşması gereken hedef nokta olarak gösterilmiştir.



Şekil 5.7 Üçüncü Uygulamanın Senaryo Haritası

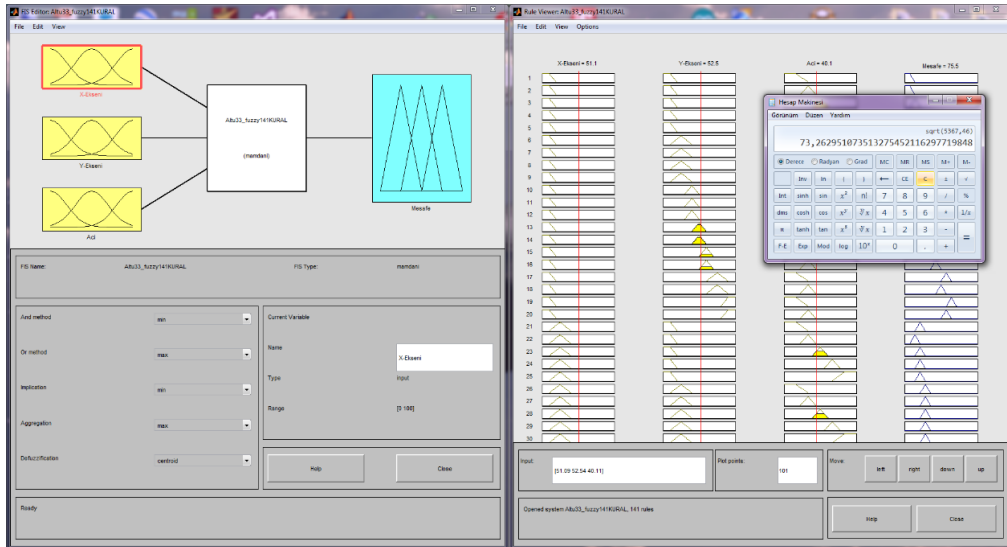
Üçüncü uygulama sonucunda Altu robot hedef nokta olarak belirlenen X: 75 cm ve Y: 85 cm koordinatına gönderilmiştir. Varılan konumun işlemsel doğruluğu için “Hata payı ölçüm” butonuna basılmalıdır. Açılan “İşlem sonuçları” penceresinden X eksenindeki hata payı, Y eksenindeki hata payı ve Altu robotun harita üzerinde belirtilen “A” noktasından “V” noktasına hareketi esnasında geçen işlem süresi ölçümü verilmiştir. “S” işareti ise Altu robotun harita üzerinde yapılan ölçümler sonucu son konumu olarak gösterilmiştir. Yapılan uygulama için ölçülen işlem değerleri ve rota sonucu Şekil 5.8’de verilmiştir.

İşlem sonucu X ekseninde 3cm ve Y ekseninde 3cm konum hata payı ile robot 77.1cm mesafe gitmiştir. Bu sonuca göre X eksenindeki konum hata payı %3.127365, Y eksenindeki konum hata payı %3.195005 olarak hesaplanmıştır ve işlem için geçen süre 30 saniye olarak ölçülmüştür.



Şekil 5.8 Açıklanan Üçüncü Uygulamanın Sonuç Haritası

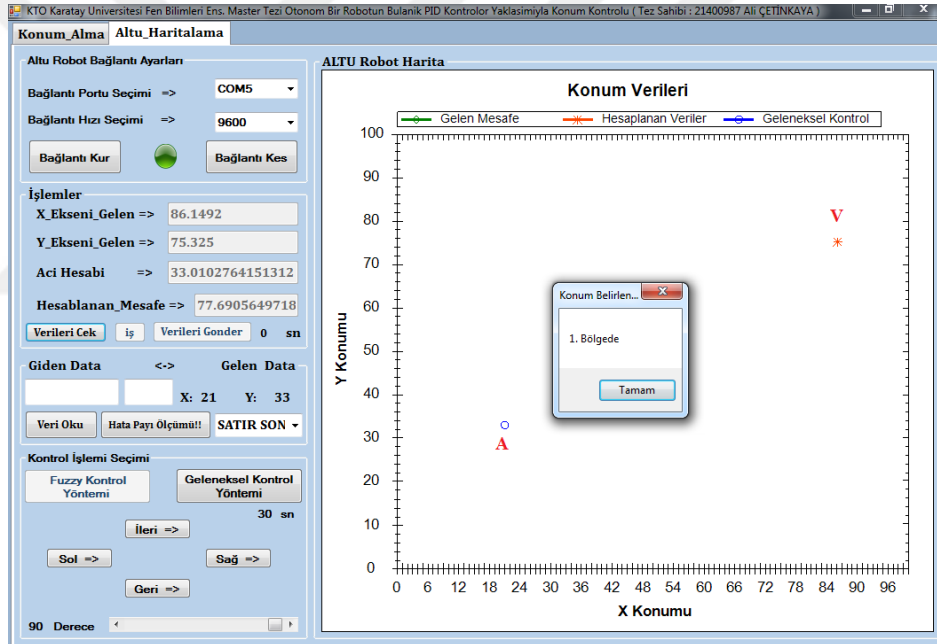
Şekil 5.9'de Matlab programında denemesi yapılan üçüncü uygulamanın sonucu verilmiştir.



Şekil 5.9 Açıklanan Üçüncü Uygulamanın Matlab Sonuç Haritası

5.4. Dördüncü Uygulama

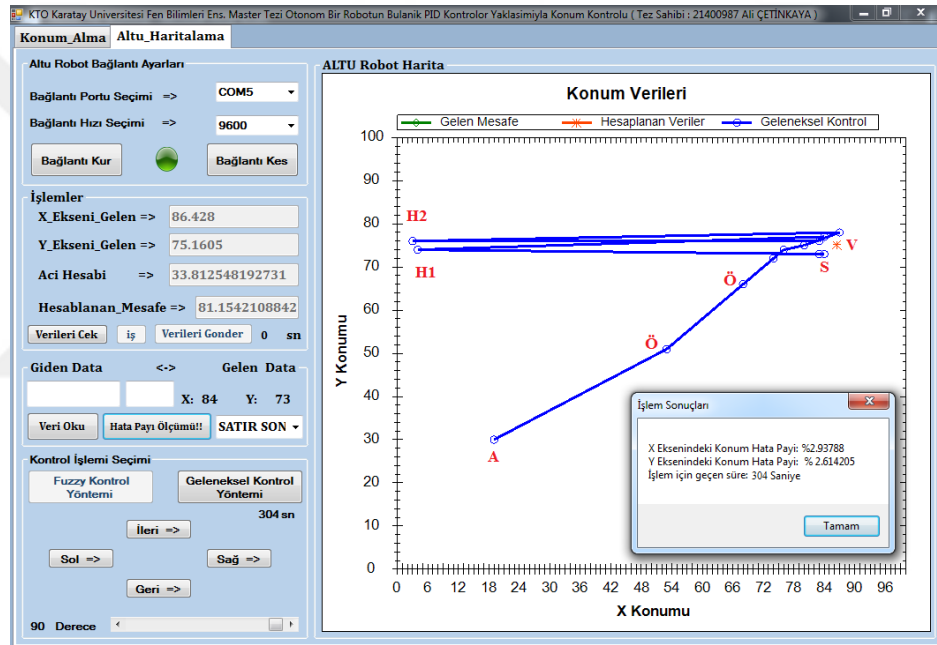
Dördüncü uygulamada Altu robot pist içinde başlangıç konumu X: 21 cm ve Y: 33 cm yerleştirilmiştir. Konum alma sekmesinde hedef nokta olarak X: 86 cm ve Y: 75 cm belirlenmiştir. Kontrol işlemi seçiminde “Geleneksel” kontrol yöntemi seçilerek yazılım üzerinde veri oku butonu ile Altu robotun pist üzerindeki koordinatları harita üzerinde mavi renkli daire olarak belirtilmiştir. Hedef konumu ise “Verileri Çek” butonuna basıldığında turuncu yıldız şeklinde harita üzerinde belirtilmiştir. Şekil 5.10’de Altu robotun ikinci uygulama için program görüntüsü verilmiştir. Şekil üzerinde belirtilen “A” başlangıç konumu olarak, “V” robotun ulaşması gereken hedef nokta olarak gösterilmiştir.



Şekil 5.10 Dördüncü Uygulamanın Senaryo Haritası

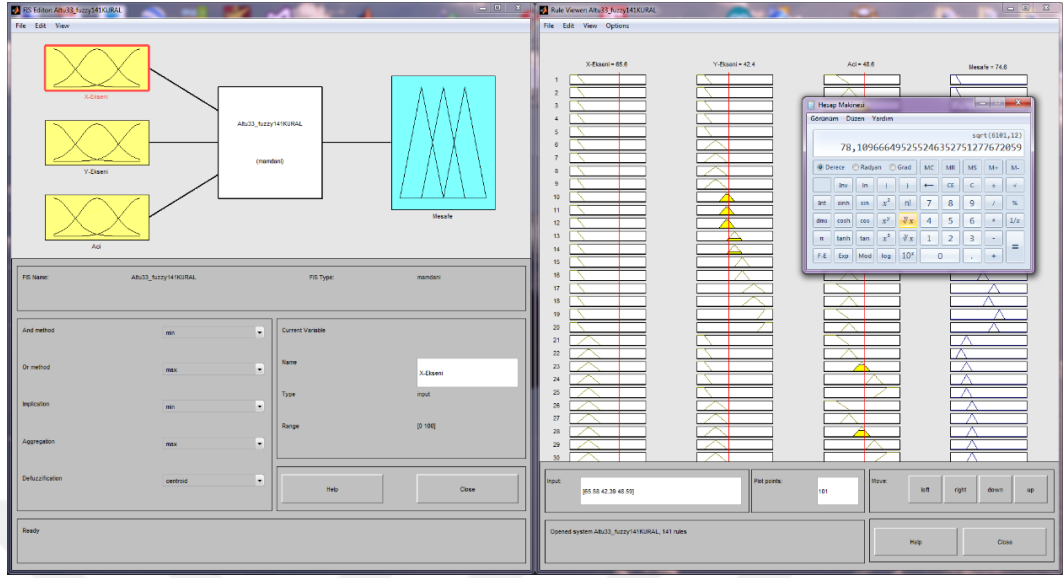
İkinci uygulama sonucunda Altu robot hedef nokta olarak belirlenen X: 30 cm ve Y: 89 cm koordinatına gönderilmiştir. Varılan konumun işlemsel doğruluğu için “Hata payı ölçüm” butonuna basılmalıdır. Açılan “İşlem sonuçları” penceresinden X eksenindeki hata payı, Y eksenindeki hata payı ve Altu robotun harita üzerinde belirtilen “A” noktasından “V” noktasına hareketi esnasında geçen işlem süresi ölçümü verilmiştir. Şekil 5.11 üzerinde rota bilgisinin doğruluğu için hareket esnasında Altu robotun bulunduğu konumun verileri alınarak harita üzerinde “Ö” olarak gösterimi yapılmıştır.

Harita üzerinde “H1” ve “H2” olarak belirtilen noktalar Altu robotun hareket esnasında ultrasonik sensörlerden alınan ölçümlerin hatasını göstermektedir. “S” işareti ise Altu robotun harita üzerinde yapılan ölçümler sonucu son konumu olarak gösterilmiştir. Yapılan uygulama için ölçülen işlem değerleri ve rota sonucu Şekil 5.11’de verilmiştir. İşlem sonucu X ekseninde 2cm ve Y ekseninde 2cm konum hata payı ile robot 97.43cm mesafe gitmiştir. Bu sonuca göre X eksenindeki konum hata payı %2.93788, Y eksenindeki konum hata payı %2.614205 olarak hesaplanmıştır ve işlem için geçen süre 304 saniye olarak ölçülmüştür.



Şekil 5.11 Açıklanan Dördüncü Uygulamanın Sonuç Haritası

Şekil 5.12’de Matlab programında denemesi yapılan dördüncü uygulamanın sonucu verilmiştir.



Şekil 5.12 Açıklanan Dördüncü Uygulamanın Matlab Sonuç Haritası

6. SONUÇ ve ÖNERİLER

Altu robotun kontrolü için Bulanık yaklaşımı ile Geleneksel kontrol yöntemi olmak üzere iki farklı kontrol yöntemi tasarlanarak sistem denemeleri yapılmıştır. Uygulamalar bölümünde bu iki kontrol yöntemi de test edilerek, bu testlerin sonuçları rota takibi verilerinin de olduğu harita bilgisi üzerinde verilmiştir.

Altu robotun yazılımının ve donanımının kullanıcıya sağladığı en önemli avantajı hareketleri eşzamanlı olarak yaptırılabilmesidir. Kullanılan her iki kontrol yöntemi için kendi yörüngesine bağlı olarak gideceği koordinat verileri önceden tanımlı değildir.

Robot üzerine kullanılan ultrasonik sensörler sabit bir mekânda hedefe ulaşma işlemlerinde konum algılama için kullanılmıştır. Bu sayede robotun bulunduğu yerin haritasını çıkarmadan, Altu robota ek bir güç harcatmadan hedefe ulaşmayı sağlamıştır.

Uygulama bölümünde ele alınan senaryolar ile bu tez çalışmasının literatür araştırmasında bahsi geçen konular üzerindeki etkisini ve doğruluğunu ortaya koymuştur. Bulanık yaklaşımı ile yapılan uygulamaların sonucu 29 – 30 saniye sürdüğü ölçülmüştür. Geleneksel yaklaşım ile yapılan uygulamaların sonucu 300 – 341 saniye sürdüğü ölçülmüştür. Bu sonuçlar doğrultusunda Bulanık yaklaşımı kullanılan yöntemin geleneksel yöntemle göre on kat daha hızlı ve doğru çalıştığı görülmüştür. Gelişime açık olması ve modüler olması açısından Altu robot bu tez adına amaçlanan görevlerini yerine getirmiştir.

İleride harici sensörler ile daha uzun mesafelere ulaşabilmek için Altu robot üzerine dijital elektronik pusula kullanarak yön tayini yapılabilir. Tez çalışmasında kullanılan bluetooth modül yerine Wi-fi modül kullanılarak kendine özgü ağ kurup dış saha ortamına aktarım yapılabilir veya internet üzerinden kontrol sağlanabilir. Wi-fi modül ile oluşturulan ağa birden fazla Altu robot bağlanıp farklı görevler yaptırılabilir. Tez çalışmasında tasarlanan robot üzerinde daha güçlü bir mikroişlemci kullanılarak ses ile komut gönderimi yapılabilir. Ayrıca haritalama işleminin hassasiyetinin artırılması için yapay zeka ve kamera kullanılarak daha başarılı sonuçlar alınabileceği düşünülmektedir.

EKLER

Ek Açıklama

EK-1: ARDUİNO MEGA ŞEMATİĞİ

EK-2: ARDUİNO MEGA GÖRÜNÜMÜ

EK-3: ARDUİNO MEGA PİN BAĞLANTI ŞEMASI

EK-4: L298 MOTOR SÜRÜCÜ ŞEMASI

EK-5: L298 MOTOR SÜRÜCÜ GÖRÜNÜMÜ

EK-6: L298 MOTOR SÜRÜCÜ PİN BAĞLANTI ŞEMASI

EK-7: HC-SR04 ULTRASONİK SENSÖR ÇALIŞMA ŞEMASI

EK-8: HC05 KABLOSUZ HABERLEŞME MODÜLÜN KULLANILAN PİNLERİ

EK-9: HC05 KABLOSUZ HABERLEŞME MODÜLÜ DEVRE ŞEMASI

EK-10: ULTRASONİK SENSÖR İÇİN TASARLANAN PROGRAM

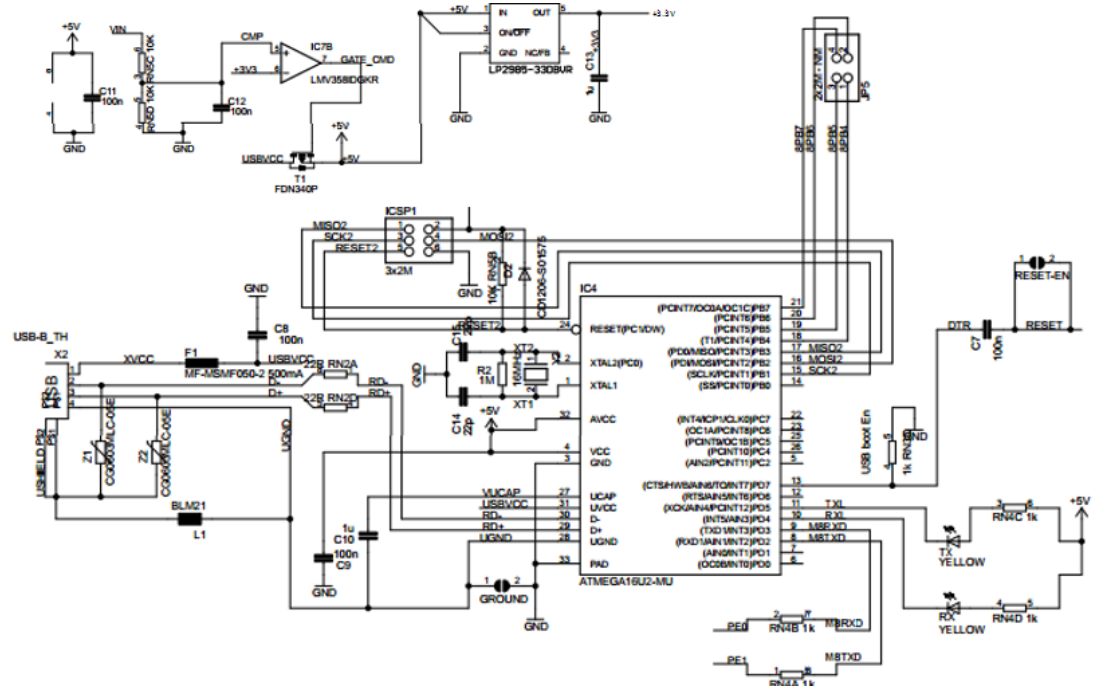
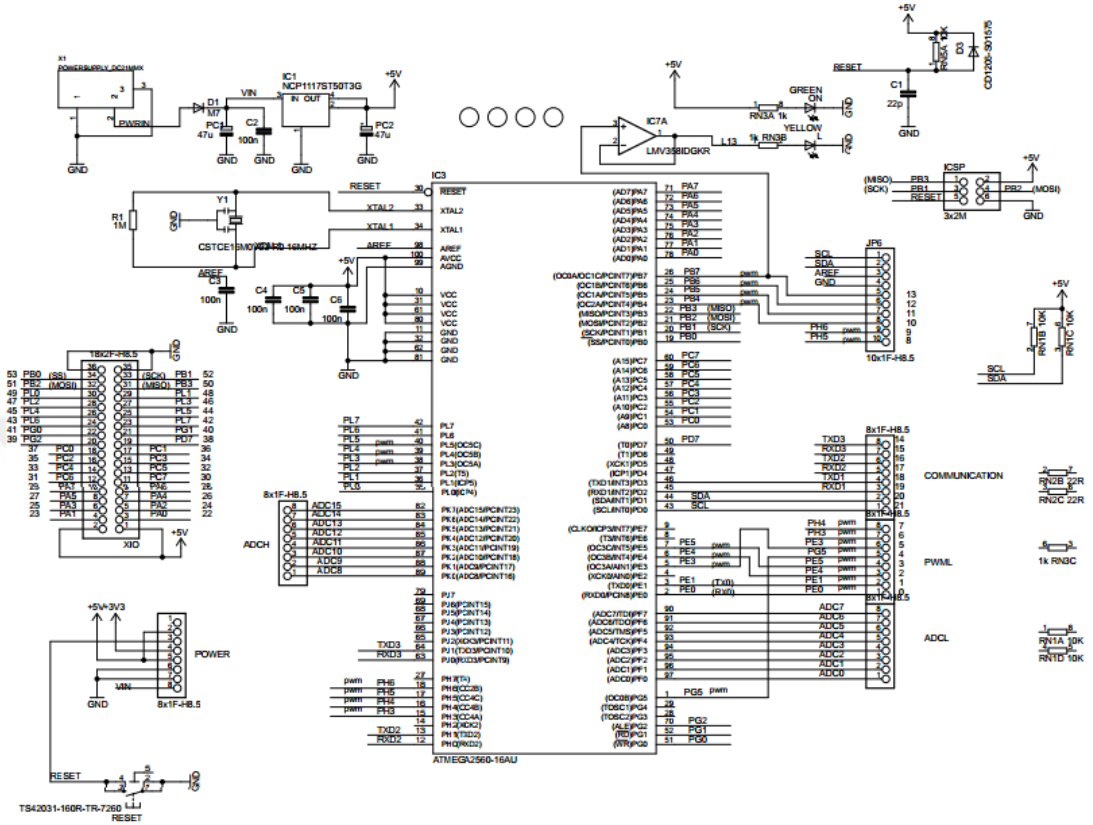
EK-11: KABLOSUZ HABERLEŞME İÇİN TASARLANAN PROGRAM

EK-12: DC MOTOR KONTROLÜ İÇİN TASARLANAN PROGRAM

EK-13: SERVO MOTOR KONTROLÜ İÇİN TASARLANAN PROGRAM

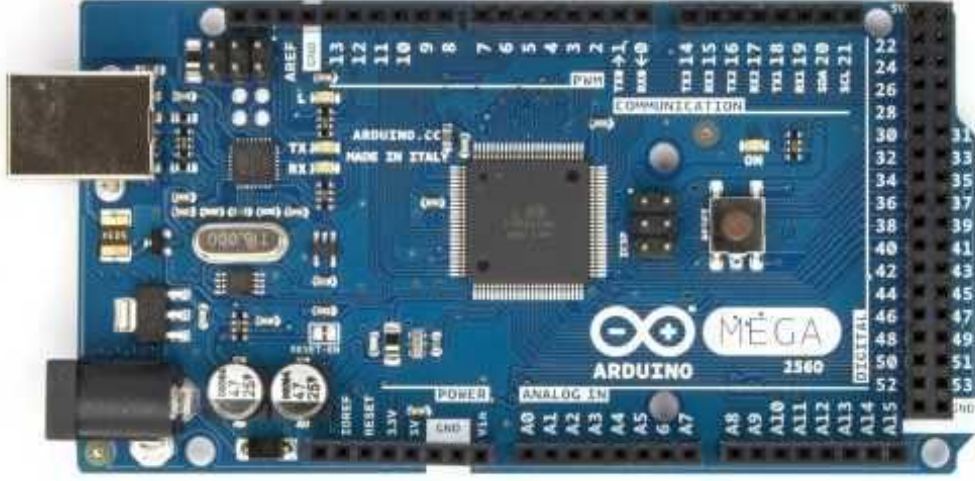
EK-14: BULANIK MANTIK PROGRAMININ TANIMLAMA BÖLÜMÜ

EK-1



ARDUİNO MEGA ŞEMATIĞI

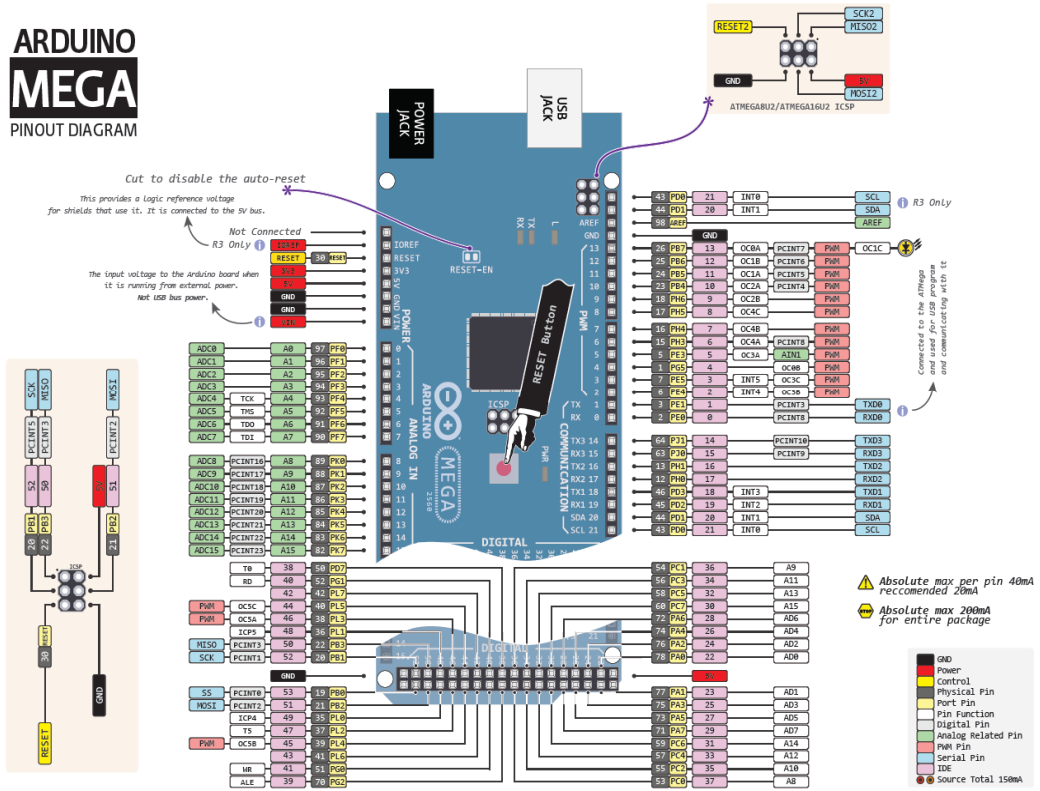
EK-2



ARDUİNO MEGA GÖRÜNÜMÜ

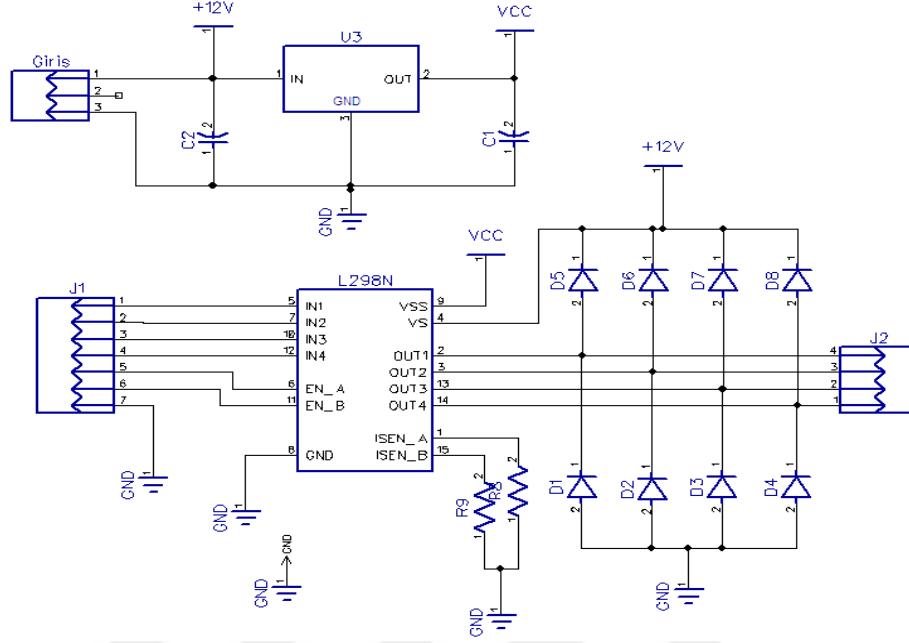
EK-3

ARDUINO MEGA PINOUT DIAGRAM



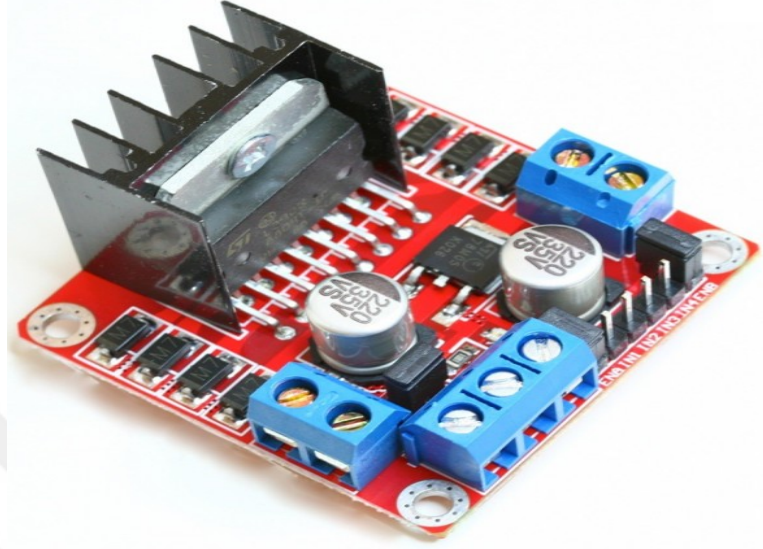
ARDUİNO MEGA PİN BAĞLANTI ŞEMASI

EK-4



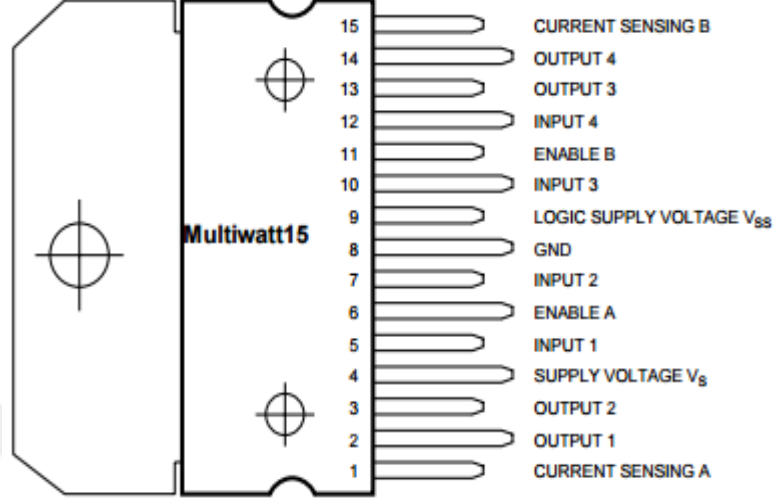
L298 MOTOR SÜRÜCÜ ŞEMASI

EK-5



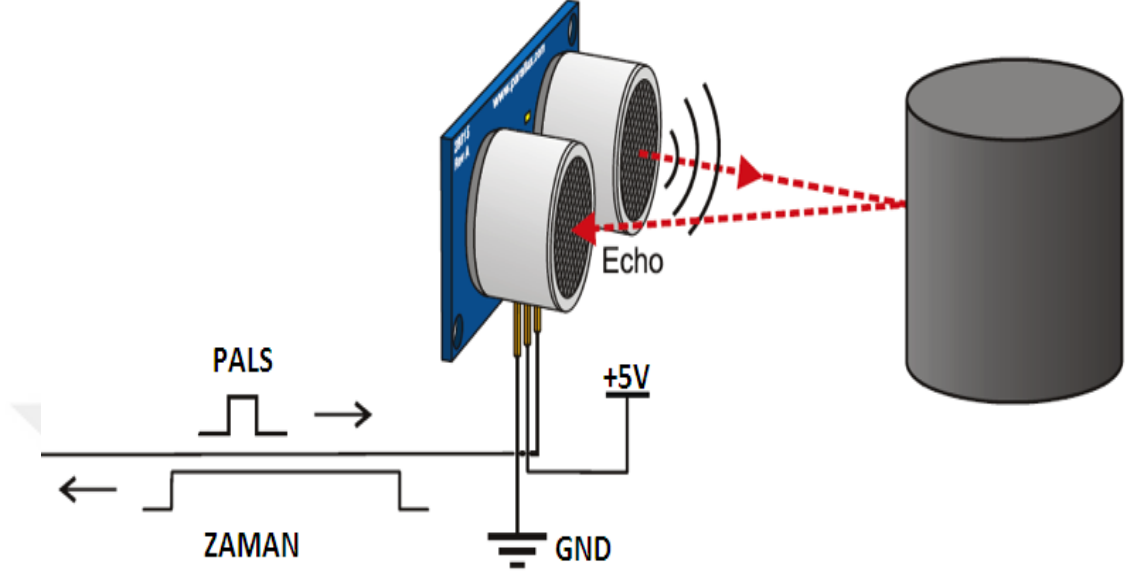
L298 MOTOR SÜRÜCÜ GÖRÜNÜMÜ

EK-6



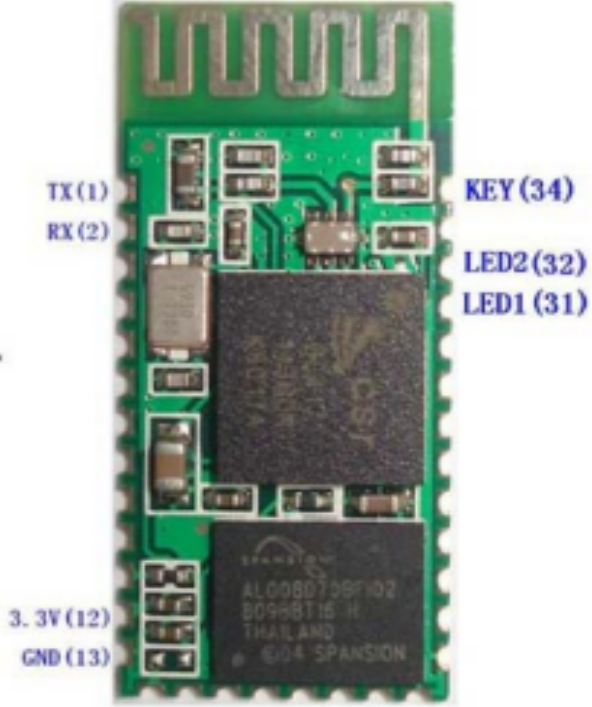
L298 MOTOR SÜRÜCÜ PİN BAĞLANTI ŞEMASI

EK-7



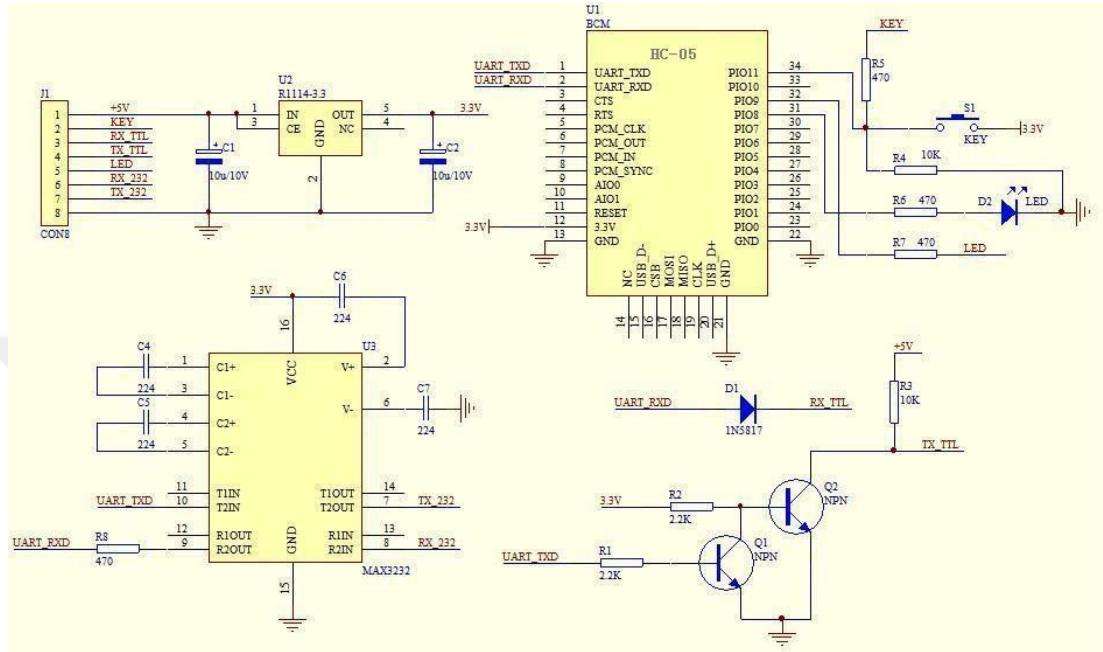
HC-SR04 ULTRASONİK SENSÖR ÇALIŞMA ŞEMASI

EK-8



HC05 KABLOSUZ HABERLEŞME MODÜLÜ GÖRÜNÜMÜ

EK-9



HC05 KABLOSUZ HABERLEŞME MODÜLÜ DEVRE ŞEMASI

EK-10

```
#include <NewPing.h>

#define TRIG_PIN          12 // Arduino üzerinde pin tanımlaması.
#define ECHO_PIN          11 // Arduino üzerinde pin tanımlaması.
#define MAKSIMUM_UZAKLIK 100 // Ultrasonik sensörün çalışma aralığının //
belirlenmesi

NewPing Ultrasonik_sensor(TRIG_PIN , ECHO_PIN, MAKSIMUM_UZAKLIK);

void setup()
{
  Serial.begin(115200); // Mikrodenetleyici ile PC bağlantı hızı.
}

void loop() // sonsuz döngünün başlangıç noktası.
{

  delay(50); // 50 milisaniye gecikme verildi.
  Serial.print("Mesafe: "); // seri porttan “Mesafe: ” yazdırıldı.
  // HC05 Ultrasonik sensörlerden gelen verileri cm olarak “Ultrasonik_sensör
  // .Ping_cm” kullanıcıya iletir.
  Serial.print(Ultrasonik_sensor.ping_cm());
  // Seri porttan ultrasonik sensörün cm cinsinden verisi yazdırılır.
  Serial.println("cm"); // seri porttan “cm” yazdırılır.

}
```

EK-11

```
void loop() // döngü bloğu
{
  if (Serial.available() > 0)
  // Seri port üzerinde bir veri var mı diye “if” döngüsü ile seri portun kontrolü yapılıyor.
  {
    if (gelen == 'F') // Seri port üzerinden ‘F’ karakteri geldi mi?
    {
      motor_ileri(); // motor ileri fonksiyonuna dallan.
      gelen = 'S'; // gelen değişkenine ‘S’ verisini yükle.
    }
    else if(gelen == 'B') // Seri port üzerinden ‘B’ karakteri geldi mi?
    {
      motor_geri(); // motor geri fonksiyonuna dallan.
      gelen = 'S'; // gelen değişkenine ‘S’ verisini yükle.
    }
    else if(gelen == 'S') // Seri port üzerinden ‘S’ karakteri geldi mi?
    {
      motor_dur(); // motor dur fonksiyonuna dallan.
      gelen = 'S'; // gelen değişkenine ‘S’ verisini yükle.
    }
    else if(gelen == 'L') // Seri port üzerinden ‘L’ karakteri geldi mi?
    {
      motor_sol(); // motor sol fonksiyonuna dallan.
      gelen = 'S'; // gelen değişkenine ‘S’ verisini yükle.
    }
    else if(gelen == 'R') // Seri port üzerinden ‘R’ karakteri geldi mi?
    {
      motor_sag(); // motor sag fonksiyonuna dallan.
      gelen = 'S'; // gelen değişkenine ‘S’ verisini yükle.
    } } // döngü bloğunun bitişi.
```

EK-12

// Altu robotun Sol Tekerinin Kontrol Pinleri

int motor1Pin1 = 51; // Sol motorun birinci bağlantı kablosunun tanımlandığı pin

int motor1Pin2 = 50; // Sol motorun ikinci bağlantı kablosunun tanımlandığı pin

int enable1Pin = 4; // Sol motorun PWM değerininin sağlandığı pin.

// Altu robotun Sağ Tekerinin Kontrol Pinleri

int motor2Pin1 = 53; // Sağ motorun birinci kablosunun tanımlandığı pin

int motor2Pin2 = 52; // Sağ motorun ikinci kablosunun tanımlandığı pin

int enable2Pin = 5; // Sağ motorun PWM değerininin sağlandığı pin.

void setup() // sonsuz döngü başlangıcı

{

// Altu robotun yazılımın tanımlamaların yapıldığı yer.

// Bu bölüm mikroişlemci üzerinde pin durumlarının ayarladığı bölümdür.

// OUTPUT: tanımlandığı pinin çıkış olarak ayarlandığı bölümdür.

// INPUT: tanımlandığı pinin giriş olarak ayarlandığı bölümdür.

pinMode(motor1Pin1, OUTPUT); // Mikroişlemcinin 51. Pini çıkış yapıldı.

pinMode(motor1Pin2, OUTPUT); // Mikroişlemcinin 50. Pini çıkış yapıldı.

pinMode(enable1Pin, OUTPUT); // Mikroişlemcinin 4. Pini çıkış yapıldı.

pinMode(motor2Pin1, OUTPUT); // Mikroişlemcinin 53. Pini çıkış yapıldı.

pinMode(motor2Pin2, OUTPUT); // Mikroişlemcinin 52. Pini çıkış yapıldı.

pinMode(enable2Pin, OUTPUT); // Mikroişlemcinin 5. Pini çıkış yapıldı.

// Bu bölümde tanımlanan pinler sisteme ilk elektrik verildiğinde parazitlerden

// etkilenmesin diye bu pinlere HIGH ve LOW değişkenlerini atılmaktadır.

// HIGH: O Pinin dijital 1 (5V) seviyesin de olmasını sağlar.

// LOW: O Pinin dijital 0 (GND) seviyesin de olmasını sağlar.

```
digitalWrite(enable1Pin, HIGH); // Mikroişlemcinin 4. Pini HIGH yapıldı.  
digitalWrite(enable2Pin, HIGH); // Mikroişlemcinin 5. Pini HIGH yapıldı.  
digitalWrite(motor1Pin1, LOW); // Mikroişlemcinin 51. Pini LOW yapıldı.  
digitalWrite(motor1Pin2, LOW); // Mikroişlemcinin 50. Pini LOW yapıldı.  
digitalWrite(motor2Pin1, LOW); // Mikroişlemcinin 53. Pini LOW yapıldı.  
digitalWrite(motor2Pin2, LOW); // Mikroişlemcinin 52. Pini LOW yapıldı.
```

// Mikroişlemci üzerinde bulunan dijital pinler 1 veya 0 değerlerine karşılık gelen işlemleri yapmaktadır. Analog pinler ise 0 ile 255 arasındaki tüm değerlere karşılık gelen işlemleri yapmaktadır.

```
// Altu robot üzerinde bulunan mikroişlemcinin 4. ve 5. pinleri analog çıkış olarak, //  
50., 51., 52. ve 53. pinler dijital çıkış olarak kullanılmaktadır.
```

```
}
```

```
void loop() // Sonsuz döngü başlangıcı
```

```
{
```

```
// Yazılımın döngü bloğu.
```

```
motor_geri(); // motor geri fonksiyonuna dalkan.
```

```
delay(1000); // 1000 milisaniye (1 saniye) bekleme verildi.
```

```
motor_ileri(); // motor ileri fonksiyonuna dalkan.
```

```
delay(1000); // 1000 milisaniye (1 saniye) bekleme verildi.
```

```
motor_sag(); // motor sağ fonksiyonuna dalkan.
```

```
delay(1000); // 1000 milisaniye (1 saniye) bekleme verildi.
```

```
motor_sol(); // motor sol fonksiyonuna dalkan.
```

```
delay(1000); // 1000 milisaniye (1 saniye) bekleme verildi.
```

```
}
```

```
void motor_geri() // Robotun Geri Hareketi.
```

```
{
```

```
digitalWrite(motor1Pin1, HIGH); // Mikroişlemcinin 51. Pini HIGH yapıldı.
```

```
digitalWrite(motor1Pin2, LOW); // Mikroişlemcinin 50. Pini LOW yapıldı.
```

```
digitalWrite(motor2Pin1, LOW); // Mikroişlemcinin 53. Pini LOW yapıldı.
```

```

    digitalWrite(motor2Pin2, HIGH); // Mikroişlemcinin 52. Pini HIGH yapıldı.
    analogWrite(enable1Pin, 255); // Mikroişlemcinin 4. Pini 255 yapıldı.
    analogWrite(enable2Pin, 255); // Mikroişlemcinin 5. Pini 255 yapıldı.
}
void motor_ileri() // Robotun İleri Hareketi.
{
    digitalWrite(motor1Pin1, LOW); // Mikroişlemcinin 51. Pini LOW yapıldı
    digitalWrite(motor1Pin2, HIGH); // Mikroişlemcinin 50. Pini HIGH yapıldı
    digitalWrite(motor2Pin1, HIGH); // Mikroişlemcinin 53. Pini HIGH yapıldı.
    digitalWrite(motor2Pin2, LOW); // Mikroişlemcinin 52. Pini LOW yapıldı.
    analogWrite(enable1Pin, 255); // Mikroişlemcinin 4. Pini 255 yapıldı.
    analogWrite(enable2Pin, 255); // Mikroişlemcinin 5. Pini 255 yapıldı.
}
void motor_sag() // Robotun Sağ Hareketi.
{
    digitalWrite(motor1Pin1, LOW); // Mikroişlemcinin 51. Pini LOW yapıldı
    digitalWrite(motor1Pin2, HIGH); // Mikroişlemcinin 50. Pini HIGH yapıldı
    digitalWrite(motor2Pin1, LOW); // Mikroişlemcinin 53. Pini LOW yapıldı.
    digitalWrite(motor2Pin2, HIGH); // Mikroişlemcinin 52. Pini HIGH yapıldı.
    analogWrite(enable1Pin, 255); // Mikroişlemcinin 4. Pini 255 yapıldı.
    analogWrite(enable2Pin, 255); // Mikroişlemcinin 5. Pini 255 yapıldı.
}
void motor_sol() // Robotun Sol Hareketi.
{
    digitalWrite(motor1Pin1, HIGH); // Mikroişlemcinin 51. Pini HIGH yapıldı
    digitalWrite(motor1Pin2, LOW); // Mikroişlemcinin 50. Pini LOW yapıldı
    digitalWrite(motor2Pin1, HIGH); // Mikroişlemcinin 53. Pini HIGH yapıldı.
    digitalWrite(motor2Pin2, LOW); // Mikroişlemcinin 52. Pini LOW yapıldı.
    analogWrite(enable1Pin, 255); // Mikroişlemcinin 4. Pini 255 yapıldı.
    analogWrite(enable2Pin, 255); // Mikroişlemcinin 5. Pini 255 yapıldı.
}

```



```
void motor_dur() // Robotun Durma Hareketi.  
{  
    digitalWrite(motor1Pin1, LOW); // Mikroişlemcinin 51. Pini LOW yapıldı  
    digitalWrite(motor1Pin2, LOW); // Mikroişlemcinin 50. Pini LOW yapıldı  
    digitalWrite(motor2Pin1, LOW); // Mikroişlemcinin 53. Pini LOW yapıldı.  
    digitalWrite(motor2Pin2, LOW); // Mikroişlemcinin 52. Pini LOW yapıldı.  
    analogWrite(enable1Pin, 0); // Mikroişlemcinin 4. Pini 0 yapıldı.  
    analogWrite(enable2Pin, 0); // Mikroişlemcinin 5. Pini 0 yapıldı.  
}
```



EK-13

```
// Altu robotun çift servo kontrolü
#include <Servo.h> // Arduino yazılımı içerisinde Servo.h kütüphane dosyasını dahil
// edilmektedir.
Servo sol_servo; // Servo.h üzerinden yazılımda kullandığımız servoları temsil eden
Servo sag_servo; // sol ve sağ servo nesnelere oluşturulmuştur.
int pos = 0; // Servo.h üzerinde tanımlı nesnelere kullanıcının belirleyeceği açılar
// yazılım üzerinde tutacak olan değişken.
void setup() // Yazılıma başlamadan önce pin tanımlamaların yapıldığı bölüm.
{
  delay(5000); // 5000ms'ye yazılım ilk çalışmada bekletildi.
  sol_servo.attach(7); // Sol servonun sisteme tanıtılması gerçekleştirildi.
  sag_servo.attach(6); // Sağ servonun sisteme tanıtılması gerçekleştirildi.
  sol_servo.write(90); // Sol servo ilk çalışmada 90 dereceye alındı.
  sag_servo.write(90); // Sağ servo ilk çalışmada 90 dereceye alındı.
  delay(15); // Servoların hareketlerini gerçekleştirme için gerekli olan
// gecikme zamanı sisteme yükledi.
}
void loop() // Sonsuz döngü başlangıcı
{
  for (pos = 0; pos <= 180; pos += 1) // 0 ile 180 arasından for tanımlandı.
  {
    // İlk Servo motor 0 dereceden 180 dereceye 1
// derecelik artışlar ile hareket sağlamaktadır.
    sol_servo.write(pos); // for döngüsü içerisinde tanımlı olan pos değişkeni ile
// servo motor hareket ettiriliyor.
    delay(15); // Servo motor her hareketi için 15ms beklemek
// gerekmektedir. Bu süreyi "delay(15)" ile yazılıma tanıtılmaktadır.
  }
  for (pos = 180; pos >= 0; pos -= 1) // 180 ile 0 arasından for tanımlandı.
  {
    // İkinci Servo motor 180 dereceden 0 dereceye 1
// derecelik azalışlar ile hareket sağlamaktadır

```

```
sag_servo.write(pos);      // for döngüsü içerisinde tanımlı olan pos değişkeni ile
// servo motor hareket ettiriliyor.
delay(15);                 // for döngüsü içerisinde tanımlı olan pos değişkeni ile
// servo motor hareket ettiriliyor.
}
}
```



EK-14

```
void setup()

{

  Serial.begin(9600);

  // KONUM için bulanık girdi değişkeni oluşturuyoruz.

  Fuzzy_Girisi x_konum_fuzzy = Fuzzy_Girisi(1);

  // x_konum_fuzzy değişkeni için oluşturulan üyelik değişkenleri eklenmiştir.

  x_konum_fuzzy => Fuzzy değişkeni(XCK);      // X Çok Küçük
  x_konum_fuzzy => Fuzzy değişkeni (XKO);     // X Küçük Orta
  x_konum_fuzzy => Fuzzy değişkeni (XK);      // X Küçük
  x_konum_fuzzy => Fuzzy değişkeni (XO);      // X Orta
  x_konum_fuzzy => Fuzzy değişkeni (XB);      // X Büyük
  x_konum_fuzzy => Fuzzy değişkeni (XBO);     // X Büyük Orta
  x_konum_fuzzy => Fuzzy değişkeni (XCB);     // X Çok Büyük

  // x_konum_fuzzy girdi değişkenini Bulanık nesne haline getirilmiştir.

  Fuzzy => ekle_Fuzzy_girisi(x_konum_fuzzy); //

  //-----

  // y_konum_fuzzy için bulanık girdi değişkeni oluşturulmaktadır.

  Fuzzy_Girisi y_konum_fuzzy = Fuzzy_Girisi(2);

  // HIZ değişkeni için oluşturulan üyelik değişkenleri eklenmiştir.
```

```

y_konum_fuzzy=> Fuzzy deęişkeni (YCK); // Y Çok Küçük
y_konum_fuzzy=> Fuzzy deęişkeni (YKO); // Y Küçük Orta
y_konum_fuzzy=> Fuzzy deęişkeni (YK); // Y Küçük
y_konum_fuzzy=> Fuzzy deęişkeni (YO); // Y Orta
y_konum_fuzzy=> Fuzzy deęişkeni (YB); // Y Büyük
y_konum_fuzzy=> Fuzzy deęişkeni (YBO); // Y Büyük Orta
y_konum_fuzzy=> Fuzzy deęişkeni (YCB); // Y Çok Büyük
// y_konum_fuzzy girdi deęişkenini Bulanık nesne haline getirilmiştir.
Fuzzy => ekle_Fuzzy_girisi (y_konum_fuzzy);
//-----//
// aci_fuzzy için bulanık girdi deęişkeni oluşturulmuştur.
Fuzzy_Girisi aci_fuzzy = Fuzzy_Girisi(3);
// aci_fuzzy deęişkeni için oluşturulan üyelik deęişkenleri eklenmiştir.
aci_fuzzy=> Fuzzy deęişkeni (AK); // Açı Küçük
aci_fuzzy=> Fuzzy deęişkeni (AKO); // Açı Küçük Orta
aci_fuzzy=> Fuzzy deęişkeni (AO); // Açı Küçük
aci_fuzzy=> Fuzzy deęişkeni (ABO); // Açı Büyük Orta
aci_fuzzy=> Fuzzy deęişkeni (AB); // Açı Büyük
// aci_fuzzy girdi deęişkenini Bulanık nesne haline getirilmiştir.
Fuzzy -> ekle_Fuzzy_girisi (aci_fuzzy);
//-----

```

```

// mesafe için bulanık çıkış değişkeni oluşturulmuştur.

Fuzzy Cikisi mesafe_fuzzy = Fuzzy Cikisi(1);

// mesafe bulanık çıkış değişkeni için üyelik değişkenleri tanımlanmıştır.

mesafe_fuzzy=> Fuzzy değişkeni (MKS); // Mesafe Küçük Sınır
mesafe_fuzzy=> Fuzzy değişkeni (MKC); // Mesafe Küçük Çok
mesafe_fuzzy=> Fuzzy değişkeni (MKB); // Mesafe Küçük Büyük
mesafe_fuzzy=> Fuzzy değişkeni (MKO); // Mesafe Küçük Orta
mesafe_fuzzy=> Fuzzy değişkeni (MOK); // Mesafe Orta Küçük
mesafe_fuzzy=> Fuzzy değişkeni (MK); // Mesafe Küçük
mesafe_fuzzy=> Fuzzy değişkeni (M); // Mesafe
mesafe_fuzzy=> Fuzzy değişkeni (MB); // Mesafe Büyük
mesafe_fuzzy=> Fuzzy değişkeni (MOB); // Mesafe Orta Büyük
mesafe_fuzzy=> Fuzzy değişkeni (MBO); // Mesafe Büyük Orta
mesafe_fuzzy=> Fuzzy değişkeni (MBK); // Mesafe Büyük Küçük
mesafe_fuzzy=> Fuzzy değişkeni (MBC); // Mesafe Büyük Çok
mesafe_fuzzy=> Fuzzy değişkeni (MBS); // Mesafe Büyük Sınır

// mesafe değişkeni için oluşturulan üyelik değişkenleri eklenmiştir.

fuzzy->addFuzzyOutput(mesafe_fuzzy);

}

void kural1()

{

```

```

//-----FUZZY-----KURAL-1-----//

//   Bulanık ifadenin oluşturulması sağlanmıştır.

Fuzzy Kural_ifadesi eksenler_1 = FuzzyKuralOncülü();

//   Bulanık alt kümelerin ifade içerisine eklenmesi sağlanmıştır.

eksenler_1->AND işlemi yap(XCK, YCK);

//   Bulanık ikinci ifadenin oluşturulması sağlanmıştır.

Fuzzy Kural_ifadesi* aci_1 = FuzzyKuralOncülü();

//   Bulanık alt kümenin ifade içerisine eklenmesi sağlanmıştır.

aci_1->Tek tanım(AK);

//   Bulanık ifadenin tamamı bir bütün içene alınmıştır.

Fuzzy Kural_ifadesi_işlemi 1_kural_btn = FuzzyKuralOncülü();

//   Bulanık ifadeye tanımlı değerler yüklenmiştir.

1_kural_btn->And işlemi(eksenler_1, aci_1);

// Bulanık sonuç ifadesinin oluşturulması

Fuzzy Kural Sonucu 1_kural_sonucu = FuzzyKuralSonucu();

// Bulanık ifadeye sonuç alt kümesinin yerleştirilmesi yapılmıştır.

1_kural_sonucu->Çıkış(MKS);

// Kural tam olarak giriş ve çıkış ifadeleri ile oluşturuldu. Ve ismi "Kural1" oldu.

Fuzzy_Kurallari kural1 = FuzzyKurallari(1, 1_kural_btn, 1_kural_sonucu);

// oluşturulan Fuzzy nesnesine kural1 eklenmiştir.

Fuzzy=>Fuzzy Kural Ekle(kural1);

```

```
}  
  
void loop()  
  
{  
  
//Döngü kendi içerisinde kullanıcıdan gelen veriler doğrultusunda sürekli olarak  
  
// işleme alınmaktadır.  
  
Olçum_al(); // ölçüm al fonksiyonu ile fuzzy algoritması içine veriler gönderilir.  
  
    if(gelen_degerler) // ölçüm fonksiyonundan gelen verilerin yazılım içine  
    {                               // aktarılmasını sağlayan ve robotun pozisyona gitmesini  
                                   // sağlar.  
    }  
  
    else // gelen değerlerin yazılım sınırları dışında ise fonksiyonun  
    {                               // dallanacağı bölüm.  
    }  
  
}
```


KAYNAKLAR

- [1] Taşdemir C, 2012, Gezgin Robotlarda Otonom Devriye Sistemi ve Algılayıcı Füzyonu, Yüksek Lisans Tezi, Y.T.Ü. Fen Bilimleri Enstitüsü, İSTANBUL.
- [2] TUNCER A, 2013, Otonom araçlar için yol bulma probleminin GA ve FPGA ile çözümü ve gerçekleştirilmesi, Doktora Tezi, K.Ü. Fen Bilimleri Enstitüsü, KOCAELİ.
- [3] ÖZEREN A, 2014, Doğal yer imi kullanarak gezgin robot konum kontrolünün otonom olarak gerçekleştirilmesi, Yüksek Lisans Tezi, K.Ü. Fen Bilimleri Enstitüsü, KARABÜK.
- [4] LAFCI M, 2016, Dinamik engellerin bulunduğu ortamda gezgin robot için hareket planlama, Yüksek Lisans Tezi, K.Ü. Fen Bilimleri Enstitüsü, KARABÜK.
- [5] YUFKA A, 2010, Birden fazla gezgin robotla nesne taşıma için hareket planlanma ve kontrol, Yüksek Lisans Tezi, E.O.Ü. Fen Bilimleri Enstitüsü, ESKİŞEHİR.
- [6] DİLMEN E, 2012, Gezgin robotlarda dış ortamda GPS ile konumlandırma, Yüksek Lisans Tezi, E.O.Ü. Fen Bilimleri Enstitüsü, ESKİŞEHİR.
- [7] ÖZALP N, 2013, 3 Boyutlu Arazi Üzerinde Çoklu Otonom İnsansız Hava Aracı Rota Planlaması, Yüksek Lisans Tezi, H.H.O. Havacılık ve Uzay Teknolojileri Enstitüsü, İSTANBUL
- [8] NİKBAY K, 2015, Otonom Araçların Güzergah Takibi İçin Bir Uygulama, Yüksek Lisans Tezi, O.Ü. Fen Bilimleri Enstitüsü, İSTANBUL
- [9] AKÇAKAYA H, 2014, Otonom Sualtı Aracının Modellemesi ve Kontrolü, Yüksek Lisans, İ.T.Ü. Fen Bilimleri Enstitüsü, İSTANBUL
- [10] UYAR O, 2010, Bulanık mantık tabanlı indüksiyon motor koruma sistemi tasarımı ve gerçekleştirilmesi, Yüksek Lisans Tezi, S.Ü. Fen Bilimleri Enstitüsü, KONYA
- [11] ERENOĞLU I, 2006, Akıllı melez bulanık pid kontrolör yapısı, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İSTANBUL
- [12] CİVELEK Z, 2013, Bulanık pid kontrolör ile rüzgâr türbininin hatve açısının kontrolü, Yüksek Lisans Tezi, K. Ü. Fen Bilimleri Enstitüsü, KIRIKKALE
- [13] GÜZAY C, 2014, Bulanık pid kontrolörleri için birleştirme operatörüne dayalı yeni bir öz-ayarlı yöntem tasarımı, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İSTANBUL
- [14] KABASAKAL O, 2012, Bulanık pid kontrolörleri için çevrim içi kural ağırlıklandırma yöntemleri, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İSTANBUL
- [15] BULUT C, 2009, Melez kontrolör tasarımı ve uygulanması, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İSTANBUL

- [16] SAĞLAM G, 2007, Hiyerarşik bulanık mantık pid kontrolörleri, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İSTANBUL
- [17] ŞAHİN M, 2015, Kendinden ayarlamalı pid kontrolör ile iki eksenli gimbal uygulaması, Doktora Tezi, G.Ü. Fen Bilimleri Enstitüsü, ANKARA
- [18] GÜLLÜ A, 2010, Kontrolör arayüz tasarımı, Yüksek Lisans Tezi, M.Ü. Fen Bilimleri Enstitüsü, İSTANBUL
- [19] DURMUŞ H, 2015, Otonom robot ve kontrol birimi tasarımı, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İSTANBUL
- [20] AKSOY R., KURNAZ S., 2009, İnsansız kara araçları ve muhabere gereksinimleri, Havacılık ve Uzay Teknolojileri Dergisi, İSTANBUL
- [21] YÜKSEL M., İKİZOĞLU S., Bir mobil robotun hedef noktaya erişimi ve toplanan verilerin RF ile transferi, Kontrol Mühendisliği Bölümü, İ.T.Ü., İSTANBUL
- [22] ÖZTÜRK S., KUZUCUOĞLU E. A., 2013, Havaalanlarındaki uçuş güvenliğini tehlikeye atan yabancı nesnelere temizliği için çoklu-robot koordinasyonu yaklaşımı, Otomatik Kontrol Ulusal Toplantısı, MALATYA
- [23] WOLFER J., GEORGE A. C., 2006, Fuzzy logic control for robot maze traversal: an undergraduate case study, World Congress on Computer Science, BRAZIL
- [24] THILMANY J., 2012, Robotic Brooms, Mechanical Engineering (The Magazine of ASME) Erişim adresi: <https://www.asme.org/engineering-topics/articles/robotics/robotic-brooms>, erişim tarihi: 06 Ekim 2016.
- [25] BOUFERA F., DEBBAT F., ADOUANE F., KHELFI F. M., 2014, Mobile robot navigation using Fuzzy Limit-Cycles in Cluttered Environment, I. J. Intelligent System and Applications, DOI:10.5845
- [26] OKATAN A., GÖKALP G., 2004, Obstacle avoidance and Fuzzy control of a mobile robot, International Conference on Informatics ICI
- [27] OKATAN A., DIMIROVSKI M. G., 2002, Fuzzy logic navigation and control of a non-holonomic vacuum cleaner, Proceedings of the 10th Mediterranean Conference on Control and Automation, LISBON
- [28] KAYA A.A., YİĞİT Y., AMASYALI F.M., 2010, Otonom bir robotla statik ortamda nesne etiketleme, Bilimde Modern Yöntemler Sempozyumu-BMYS 2010, DİYARBAKIR
- [29] HÜLAKO H., KAPUCU S., 2014, Design and Construction of an Autonomous Unmanned Ground Vehicle (UGV) and Basic Waypoint Tracking, GAU Journal of Soc. App. Sciences, vol. 6, issue10

- [30] ARDA M., GÜLLÜ A., KUŞÇU H., 2013, Bulanık mantık yönteminin PID denetleyici performansına etkisi, 16. Ulusal Makine Teorisi Sempozyumu, ERZURUM
- [31] SARIEL S., AKIN L.H., 2004, Özerk Robotlar için bir geniş alan taraması planlama yöntemi, Havacılıkta İleri Teknolojiler ve Uygulamaları Sempozyumu HİTEK'04, İSTANBUL
- [32] YİNG H., 2000, Theory and application of a novel Fuzzy PID controller using a simplified Takagi-Sugeno rule scheme, Information Sciences 123 (2000) 281-293, USA
- [33] DOĞAN H., KAPLAN K., KUNCAN M., ERTUNÇ M.H., 2015, Süspansiyon sistemi kontrolünde PID ve Bulanık Mantık yaklaşımları, Otomatik Kontrol Ulusal Toplantısı TOK'2015, DENİZLİ
- [34] KAPLAN K., KAYA T., BARUTÇUOĞLU Z., 2001, Boğaziçi Üniversitesi Robotik Grubu Araştırmaları, İSTANBUL
- [35] BODRURLU T., SÖYLEMEZ T. M., 2016, Qball X4 Dört Pervaneli Rotor Sisteminin modellenmesi ve PID ve Bulanık Mantık Yapısına Dayalı Kontrolü, Otomatik Kontrol Ulusal Toplantısı TOK'2016, ESKİŞEHİR
- [36] TETİK M., SARAÇBAŞI N., 2016, Top ve Kiriş Sisteminin Modellenmesi, Bulanık Mantık ve PID Kontrolünün Gerçekleştirilmesi ve Karşılaştırılması, Otomatik Kontrol Ulusal Toplantısı, TOK'2016, ESKİŞEHİR
- [37] YAVAS U., KENDİR F., AVL, 2016, Otomotivde Kullanılan Uyarlamalı PID Denetleyicileri için Otomatik Kalibrasyon Sistemi, Otomatik Kontrol Ulusal Toplantısı TOK'2016, ESKİŞEHİR
- [38] TURHAN İ. H., 2016, Tekerlekli Gezgin Robot Pioneer 3-DX ile Sensör Tabanlı Rastgele Ağaç Yapısı Kullanarak Ortam Haritasının Çıkarılması Uygulaması, Otomatik Kontrol Ulusal Toplantısı TOK'2016, ESKİŞEHİR
- [39] ÖKSÜZ M. DÜLGER C. L., HALICIOĞLU R., 2016, Servo Pres için Bulanık Mantık Yaklaşımı ile PID Denetleyici Tasarımı, Otomatik Kontrol Ulusal Toplantısı TOK'2016, ESKİŞEHİR
- [40] RYU C., KWAK S., CHOI B., 2014, Desing of Fuzzy Logic Systems for Position and Balance Control of Segway Mobile Robot, International Conference on Advanced Technology Sciences, ICAT'14, ANTALYA
- [41] TASKIRAN E., TEKBAS H. O., SENTURK Z., YILDIZ H., DURNA Y., KOCER H., 2016, Wi-Fi Control of Mobile Robot Motion Types Based on Differential Drive Kinematics Modelling Approach, International Conference on Advanced Technology Sciences, ICAT'16, KONYA

[42] HACIMURTAZAOĞLU M., ALLAHVERDİ N., 2016, Controlling the Helicopter Takeoff and Landing with Fuzzy Logic System, International Conference on Advanced Technology Sciences, ICAT'16, ITALY

[43] ELMAS Ç., Yapay Zeka Uygulamaları, Seçkin Yayınları, 2011



ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Ali ÇETİNKAYA
Uyruğu : TC
Doğum Tarihi ve Yeri : 13.07.1988 Ankara
Telefon : +90 554 492 85-96
e-mail : aalicetinkayaa@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lisans	Selçuk Üniversitesi, Elektronik Öğretmenliği	2014

İş Deneyimleri

Yıl	Kurum	Görevi
-----	-------	--------

Yabancı Dil

İngilizce