



T.C.
NECMETTİN ERBAKAN NİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



UZAKTAN ETKİLEŞİMLİ KARA ARACI İÇİN
ENGELDEN KAÇIŞ VE OTONOM
YÖNLENDİRME KONTROLÜNÜN BULANIK
MANTIK İLE GERÇEKLEŞTİRİLMESİ

Huzeyfe TAŞDELEN

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Eylül-2023
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Huzeyfe TAŞDELEN tarafından hazırlanan “Uzaktan Etkileşimli Kara Aracı için Engelden Kaçış ve Otonom Yönlendirme Kontrolünün Bulanık Mantık ile Gerçekleştirilmesi” adlı tez çalışması 22/09/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Doç. Dr. Kadir SABANCI

.....

Danışman

Dr. Öğr. Hakkı SOY

.....

Üye

Doç. Dr. Muhammed Fahri ÜNLERŞEN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Şerife Yurdağül KUMCU
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Huzeyfe TAŞDELEN

23.10.2023

ÖZET

YÜKSEK LİSANS TEZİ

UZAKTAN ETKİLEŞİMLİ KARA ARACI İÇİN ENGELDEN KAÇIŞ VE OTONOM YÖNLENDİRME KONTROLÜNÜN BULANIK MANTIK İLE GERÇEKLEŞTİRİLMESİ

Huzeyfe TAŞDELEN

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Hakkı SOY

2023, 54 Sayfa

Jüri

Dr. Öğr. Üyesi Hakkı SOY

Doç. Dr. Kadir SABANCI

Doç. Dr. Muhammed Fahri ÜNLERŞEN

Otonom kontrol uygulamalarının gittikçe yaygınlaştığı günümüzde, yaşanan hızlı teknolojik gelişmelerden geleneksel kara araçları da önemli ölçüde etkilenmiştir. Otonom araçların kendilerini geleneksel araçlardan ayıran yeteneği, belirlenen hedefe insan müdahalesi olmadan ilerleyebilmeleridir. Otonom bir araç bulunduğu konumu referans alarak, ulaşması gereken hedef noktasına nasıl ulaşabileceğini belirleyebilir. Bu amaçla öncelikle hangi yöne doğru hareket etmesi gerektiğine karar verir, ardından önüne çıkan engelleri algılayıp aşarak ulaşması gereken konuma varır. Bu tez çalışmasında, üzerine ultrasonik sensörler yerleştirilerek prototip tasarımı yapılan kara aracı için yön tayini ve engelden kaçma görevlerinin yerine getirilmesine yönelik otonom kontrol uygulaması gerçekleştirilmiştir. Gerçekleştirilen deneysel çalışmada, araç direksiyon sisteminde tank paleti yerine geleneksel direksiyon sistemi kullanılması tercih edilmiştir. Direksiyon hareket kontrolü için servo motordan faydalanılmıştır. Araca hareket vermek için ise düşük hızlı yüksek torklu doğru akım motoru kullanılmıştır. Araç üzerinde kontrol ünitesi olarak Arduino Mega Programlama kartı bulunmaktadır. Ayrıca yön tayini için IMU sensör modülü ve konum bilgisi okuma için GPS modülü yer almaktadır. Otonom araç kontrolü için dört farklı senaryo üzerinde test çalışmaları yapılmış olup, elde edilen sonuçlar hedeflenen konuma ulaşmak için uygulanan yön tespit ve engelden kaçma algoritmalarının benzer amaçlı gerçek hayat senaryolarında uygulanabileceğini göstermiştir.

Anahtar Kelimeler: Otonom Araç, Yön Bulma, Engelden Kaçma, Konum Kontrolü

ABSTRACT

MS THESIS

IMPLEMENTATION OF OBSTACLE AVOIDANCE AND AUTONOMOUS GUIDANCE CONTROL WITH FUZZY LOGIC FOR REMOTE INTERACTIVE GROUND VEHICLE

Huzeyfe TAŞDELEN

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE
OF NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE IN ELECTRIC-ELECTRONICS
ENGINEERING**

Advisor: Asst. Prof. Dr. Hakki SOY

2023, 54 Pages

Jury

Asst. Prof. Dr. Hakki SOY

Assoc. Prof. Dr. Kadir SABANCI

Assoc. Prof. Dr. Muhammed Fahri ÜNLERŞEN

Nowadays, while autonomous control applications are becoming increasingly widespread, traditional land vehicles have also been significantly affected by technological developments. The ability of autonomous vehicles to distinguish themselves from traditional vehicles is that they can advance to the set target without human intervention. An autonomous vehicle can determine how to reach its target point by referencing its location. On this basis, it first decides the direction that should move, then perceives the obstacles in front of it, and then reaches the position it needs to reach by overcoming those obstacles. In this thesis study, a prototype land vehicle was designed by placing the necessary materials on it. An autonomous control application has been implemented to perform direction determination and obstacle avoidance tasks. In the experimental study, using a traditional steering system instead of a tank pallet in the vehicle steering system was preferred. A servo motor was used for steering movement control. A low-speed, high-torque direct current motor was used to move the vehicle. The vehicle has an Arduino Mega Programming card as a control unit. There is also a compass sensor module for direction determination and a GPS module for reading location information. Test studies have been carried out on four different scenarios for autonomous vehicle control, and the results have shown that the direction detection and obstacle avoidance algorithms applied to reach the targeted location can be applied in real-life scenarios with similar purposes.

Keywords: Autonomous Vehicle, Direction Finding, Obstacle Avoidance, Position Control

ÖNSÖZ

Çalışmam sırasında beni destekleyen ve arkamda duran tez danışmanım Dr. Öğr. Üyesi Hakkı SOY'a teşekkür ederim. Yüksek lisans eğitimim süresince maddi ve manevi desteklerini benden esirgemeyen kıymetli eşime, oğluma ve aileme şükranlarımı sunarım.

Huzeyfe TAŞDELEN
KONYA-2023

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
ŞEKİLLER	ix
ÇİZELGE LİSTESİ	x
SİMGELER VE KISALTMALAR	xi
1 GİRİŞ	1
2 KAYNAK ARAŞTIRMASI	3
2.1 Tezin Amacı	7
3 MATERYAL	9
3.1 Arduino Kontrol Kartı	9
3.2 SG90 Servo Motor	10
3.3 HMC58831 Manyetometre	11
3.4 HC-SR04 Ultrasonik Mesafe Sensörü	12
3.5 L298N Motor Sürücü Modülü	14
3.6 HC-06 Bluetooth Haberleşme	15
3.7 Güç Kaynağı.....	16
3.8 GY-NEO6MV2 GPS Modülü	17
4 METOT	18
4.1 Ackerman Prensibi ile Direksiyon Kontrolü.....	18
4.2 Tahrik Sistemi	19
4.3 Donanım Tasarımı ve Yazılım Geliştirme	20
4.3.1 Ultrasonik mesafe sensörü ile engelden kaçma	21
4.3.2 Manyetometre ile yön belirleme	22
4.3.3 Gps modülü ile konum belirleme.....	23
4.3.4 Hareket yönünün belirlenmesi	24
4.3.5 Hareket yönü kontrolü	27
4.3.6 Araç ile haberleşme için geliştirilen android uygulaması.....	29
4.3.7 Bulanık mantık ile dönüş açısı belirleme.....	31
5 ARAŞTIRMA SONUÇLARI VE TARTIŞMA	41
5.1 Bulanık Kontrol Uygulaması Test Sonuçları	41
5.2 Saha Testleri.....	45
5.2.1 Senaryo 1 saha testi çalışması.....	46
5.2.2 Senaryo 2 saha testi çalışması.....	48

5.2.3	Senaryo 3 saha testi çalışması.....	49
5.2.4	Senaryo 4 saha testi çalışması.....	51
5.2.5	Değerlendirme	53
6	SONUÇ VE ÖNERİLER.....	54
7	KAYNAKLAR	56

ŞEKİLLER

Şekil 3.1 Otonom sürüş için kullanılan arduino mega kontrol kartı	9
Şekil 3.2 Arduino mega kontrol kartıyla birlikte kullanılacak sensörler	10
Şekil 3.3 Otonom sürüş yön kontrolü için kullanılan sg90 servo motor	11
Şekil 3.4 Otonom araç kontrolü yön tayini için kullanılan hmc5883l manyetometre	12
Şekil 3.5 Otonom araç kontrolü için hc-sr04 ultrasonik mesafe sensörü kullanımı	12
Şekil 3.6 Hc-sr04 ultrasonik mesafe sensörü mesafe ölçüm zamanlama diyagramı	13
Şekil 3.7 Araç üzerinde kullanılan l298n motor sürüşü modülü	14
Şekil 3.8 Hedef konum bilgisi gönderimi için kullanılan hc-06 bluetooth modülü	16
Şekil 3.9 Araç üzerinde güç kaynağı olarak kullanılan lityum-iyon pil	16
Şekil 3.10 Otonom araç kontrolünde konum belirleme için kullanılan GPS modülü	17
Şekil 4.1 Tasarlanan prototip araç için ackerman prensibi ile direksiyon kontrolü	18
Şekil 4.2 Ackerman prensibi ile direksiyon kontrolünde teker dönüş açısı hesabı	19
Şekil 4.3 Yapılan çalışmada kullanılan diferansiyel kutusu bloğu iç yapısı	20
Şekil 4.4 Kullanılan donanım birimlerinin kontrolör ile bağlantısı.....	20
Şekil 4.5 Konya ile Erzurum arasında araç hareket yönünün belirlenmesi	26
Şekil 4.6 Hedef konum ve araç konumuna göre değişen bölgeler.....	27
Şekil 4.7 App Inverter ile uygulama arayüzü geliştirme	29
Şekil 4.8 Mobil uygulama için cihaz ve çözünürlük seçimi ekranı	30
Şekil 4.11 Otonom aracın ön kısmında yer alan ultrasonik sensörler	33
Şekil 4.12 Çalışmada kullanılan bulanık mantık algoritması	33
Şekil 4.13 Bulanık mantık giriş üyelik fonksiyonları	34
Şekil 4.14 Bulanık mantık çıkış üyelik fonksiyonları.....	35
Şekil 4.15 Üyelik fonksiyonlarının görsel karşılıkları.....	36
Şekil 4.16 Alan merkezi durulaştırma yöntemi uygulaması.....	37
Şekil 4.17 Üçgende benzerlik yöntemi	38
Şekil 5.1 Benzetim 1 sonuçları	42
Şekil 5.2 Benzetim 2 sonuçları	42
Şekil 5.3 Benzetim 3 sonuçları	43
Şekil 5.4 Benzetim 4 sonuçları	43
Şekil 5.5 Benzetim 5 sonuçları	44
Şekil 5.6 Senaryo 1 için otonom sürüş kontrolü test 1 davranışı.....	46
Şekil 5.7 Senaryo 1 için otonom sürüş kontrolü test 2 davranışı.....	47
Şekil 5.8 Senaryo 1 için otonom sürüş kontrolü test 3 davranışı.....	47
Şekil 5.9 Senaryo 2 için otonom sürüş kontrolü test 1 davranışı.....	48
Şekil 5.10 Senaryo 2 için otonom sürüş kontrolü test 2 davranışı.....	48
Şekil 5.11 Senaryo 2 için otonom sürüş kontrolü test 3 davranışı.....	49
Şekil 5.12 Senaryo 3 için otonom sürüş kontrolü test 1 davranışı.....	50
Şekil 5.13 Senaryo 3 için otonom sürüş kontrolü test 2 davranışı.....	50
Şekil 5.14 Senaryo 4 için otonom sürüş kontrolü test 1 davranışı.....	51
Şekil 5.15 Senaryo 4 için otonom sürüş kontrolü test 2 davranışı.....	52

ÇİZELGE LİSTESİ

Çizelge 3.1 Sg90 servo motor özellikleri.....	11
Çizelge 3.2 Hc-sr04 ultrasonik mesafe sensörü özellikleri.....	13
Çizelge 3.3 Bluetooth modül özellikleri	15
Çizelge 4.1 Gps modülden okunan nmea kelimeleri	24
Çizelge 4.2 Bulanık mantık kural tablosu.....	35
Çizelge 5.1 Bulanık mantık test değerleri.....	41
Çizelge 5.2 Arduino benzetim sonuçları.....	45
Çizelge 5.3 Saha testi konum değerleri	46

SİMGELER VE KISALTMALAR

Simgeler

Ω	Ohm
V	Volt
mA	Miliamper
μ S	Mikrosaniye
°	Derece

Kısaltmalar

I/O	Input/Output
AVR	Augmented Voltage Right
Mhz	MegaHertz
GPS	Global Positioning System
DC	Direct Current
Kgf·cm	Kilogram Kuvvet Santimetre
I2C	Inter-Integrated Circuit
VCC	Collector Collector Voltage
GND	Ground
Cm	Santimetre
Khz	Kilohertz
PWM	Pulse Width Modulation
NMEA	National Marine Electronic Association
DC	Direct Current
RX	Receiver
TX	Transmitter

1 GİRİŞ

Son yıllarda yaşanan teknolojik gelişmelere paralel olarak farklı alanlarda, insan hayatını kolaylaştıracak çok sayıda yeni uygulama hayata geçirilmiştir. Bunlardan biri de otomotiv endüstrisinde ortaya çıkan otonom (özerk) sürüş (autonomous driving) uygulamasıdır. Yakın gelecekte otonom sürüş desteğine sahip araçların yaygınlaşması ile birlikte karayollarında daha güvenli ve konforlu yolculuk yapılması mümkün hale gelecektir. Otonom araçların askeri uygulamalar yanında arama ve kurtarma faaliyetleri, tarımsal üretim gibi çok farklı alanlarda kullanılması beklenmektedir (Faisal ve ark., 2019).

Otonom araçlar üzerlerinde yer alan kamera, RaDAR ve LiDAR gibi ortama algılamaya yönelik sensörler yardımıyla çevresini tarayarak ilerleme yeteneği sağlar. Bulduğu noktadan kendisine hedef gösterilen konuma ilerleyen bir otonom araç, hedefine ulaşabilmek için temel olarak yön tayini yapma ve engelleri aşma gibi fonksiyonları yerine getirmek zorundadır. Otonom araçlar geliştirilirken, üstlenecekleri göreve ve kullanılacakları senaryoya uygun olarak donanım ve yazılım tasarımı yapılır. Birden fazla sensörden toplanan verilerin birleştirilmesini ve böylece daha yüksek doğrulukta veri sağlanmasını amaçlayan sensör füzyonu tekniği, günümüzde otonom sürüş kontrolünde önemli bir yer tutmaktadır. Bunun yanında yapay zekâ desteği ile otonom sürüş yeteneği önemli ölçüde iyileştirilebilir (Royo ve ark., 2019).

Bu tez çalışmasında engebeli ve engellerle donatılmış arazi koşullarında hedef olarak gösterilen bir coğrafi konuma ulaşmak üzere prototip tasarımı yapılan bir kara aracı için otonom sürüş uygulaması gerçekleştirilmiştir. Tasarlanan kara aracı üzerinde engel algılama amacıyla düşük maliyetli ultrasonik sensörlere yer verilmiştir. Aracın direksiyon sisteminde servo motor kullanılmıştır. Anlık konum takibi GPS modülü ile yapılmaktadır. Hedefe ilerlerken karşılaşılan engellerin aşılabilmesi için kısaca IMU (inertial measurement unit) olarak isimlendirilen, yapısında ivmeölçer (accelerometer), jiroskop (gyroscope) ve manyetometre (magnetometer) sensörlerini barındıran modülden toplanan veriler referans alınarak bulanık kontrol uygulamasıyla yönlendirme yapılmıştır. Geliştirilen Android uygulaması ile cep telefonundan araca Bluetooth bağlantısı ile kablosuz olarak hedef koordinat bilgisi gönderilerek, başlangıç noktası konumundan hedefe yön tayini gerçekleştirilmiştir. Ayrıca hedefe doğru ilerlerken önüne çıkan engelleri aşabilmesi için araca engelden kaçış yeteneği kazandırılmıştır. Tez çalışması

kapsamında geliştirilen donanım ve yazılım ile farklı alanlarda yapılacak benzer uygulamalar için örnek teşkil edebilecek bir platform ortaya koyulması amaçlanmıştır.

2 KAYNAK ARAŞTIRMASI

Ahn ve Kang (2019) çalışmalarında; iki boyutlu lazer sensör ölçümlerine ve yol özelliklerine dayalı, otonom araçlar için bir parçacık filtreleme yeri belirleme yöntemi sunmaktadır. Bordürler ve yol işaretleri gibi yol özelliklerini tespit ederek, 2D lazer mesafe ölçer ölçümleri kullanılarak ızgara tabanlı bir özellik haritası oluşturulmuştur. Otonom aracın konumunu doğru bir şekilde tahmin etmesi için bir parçacık filtresi kullanılmıştır. Önerilen yöntemin performansı doğrulamak için deneyimler, doğru Diferansiyel Küresel Konumlandırma Sistemleri (DGPS) verileriyle karşılaştırılmıştır (Ahn ve Kang, 2019).

Armağan ve Kumbasar (2018) çalışmalarında; TORCS oyun, simülasyon ortamında bulanık mantık tabanlı otonom araç kontrol sisteminin tasarımı yapmıştır. Aracın, hiçbir bariyere çarpmadan ve yolun dışına çıkmadan yarışı tamamlayabilmesi için, aracın otonom bir şekilde ilerleyerek bulanık mantık ve klasik kontrol yapılarından oluşan akıllı bir sistem ile kontrolü geliştirilmiştir. Bununla beraber, aracın pistin dışına çıkmadan ilerleyebilmesi ve virajlarda pist içinde kalabilmesi için bulanık mantık tabanlı bir direksiyon kontrol sistemi geliştirilmiştir (Armagan ve Kumbasar, 2018).

Fan ve arkadaşları (2021) çalışmalarında; KITTI ve PASCAL VOC olmak üzere iki tür halka açık veri seti kullanmıştır. İlk olarak, KITTI veri seti, LiDAR nokta bulutları aracılığıyla elde edilen nesnelere LiDAR segmentasyonu için derinlemesine veri bilgisi sağlanmıştır. LiDAR bulut noktaları aracılığıyla nesne segmentasyonunun performansı, görüntülerde ilgilenilen bölgeyi bulmak için kullanılmaktadır. Daha sonra sinir ağı tarafından nesne tespiti için kullanılan PASCAL VOC veri seti ile ağ eğitilmiştir. Bu şekilde otonom araçlarda mekânı tanımak için kullanılan LiDAR sistemi daha verimli kullanılabilir (Fan ve ark., 2021).

Jo ve Sunwoo (2014) çalışmalarında; canlıların hareket davranışlarından esinlenen Attractor Dynamic Approach (ADA) tabanlı otonom araçlar için yeni bir yerel yol planlayıcı önermektedir. ADA'yı tanımlayan orijinal işlevler, onu holonomik olmayan aracın kısıtlamalarına uyarlamak ve bir darbe senaryosu tespit edildiğinde tepkisini iyileştirmek için değiştirilmiştir. Mevcut yaklaşım, otonom araçlar için iyi bilinen bir simülatörde, aracın bu gibi durumlarda aracın güvenliğini sağlayan yerel yollar üretebildiği üç temsili çalışma durumu altında doğrulanmıştır (Jo ve Sunwoo, 2014).

Jo ve arkadaşları (2019) çalışmalarında; otonom arabalar için tasarlanmış yol haritası için bir harita oluşturma algoritması önermektedir. Yol haritası oluşturma

algoritması, veri toplama, veri işleme ve yol modelleme olmak üzere üç adımdan oluşmaktadır. Azaltılmış miktarda kontrol parametresi ile yol geometrisini doğru bir şekilde temsil etmek için kademeli bir düzeltme algoritması önerilmiştir. Geliştirilen harita oluşturma algoritması, çeşitli yol geometrisi koşulları altında deneysel çalışmalarla doğrulanmıştır. Sonuçlar, oluşturulan yol haritasının otonom sürüş için kullanmak için yeterince doğru ve güvenilir olduğunu göstermektedir (Jo ve ark., 2019).

Li ve arkadaşları (2023) çalışmalarında; STM32 kontrol sistemine dayalı çok yönlü bir mobil lojistik sıralama aracı önerilmektedir. Bu araştırma, çok sensörlü geri besleme ve gerçek zamanlı kontrole dayanmaktadır. Deneyler ile kontrol sistemi tasarımının etkili olduğu kanıtlanmıştır. Kontrol sisteminin her zaman özel ve yeniden geliştirilmesi zor olduğu sorununun çözmek için uygulanabilir bir şema sağlanmıştır (Li ve ark., 2023).

Martinez ve arkadaşları (2017) çalışmalarında; kendi kendine sürüş teknolojisindeki zorluklar ile kör insanlar için navigasyon yardımcılarının geliştirmiştir. Konseptin bir kanıtı olarak, otomotiv endüstrisinde kullanılan bir donanım platformuna dayanan, görme engelliler için bir engellerden kaçma sistemi oluşturulmuştur. Çevreyi algılamak için, otonom arabalar için tasarlanmış Stixels algoritmasının bir uygulaması uyarlanmıştır (Martinez ve ark., 2017).

De Lima ve arkadaşları (2019) çalışmalarında; otonom bir araca mesafe sensörlerini kurmak ve bunları bir STM32 mikro denetleyici kullanarak düşük seviyeli bir CAN veri yoluna entegre etmek için kullanılan konfigürasyon sunmuştur. Bu ağ, verilerin işlendiği ve çeşitli diğer yüksek seviyeli düğümlerde kullanıldığı ROS'a (Robot İşletim Sistemi) bağlanan bir düğüme sahiptir. Temel amaç, STM32 mikrokontrolör ile yeterli işlem süresi sağlamak, CAN BUS ROS bağlantı düğümünde minimum hata ile verileri almak ve bunu araçta çeşitli işlevleri yürütmek için kullanmaktır (De Lima ve ark., 2019).

Gul ve arkadaşları (2019) çalışmalarında; Navigasyon işlemi, küresel ve yerel navigasyon üzerinde çalışmıştır. Küresel navigasyon tipinde, yol planlama için ön ortam bilgisinin bilinmesi gerekir ve çevrimdışı mod olarak da adlandırılır. Dijkstra algoritması, A* algoritması, görünürlük grafiği, yapay potansiyel alan yöntemi ve hücre ayrıştırma yöntemi gibi yöntemler küresel navigasyon işleminde sıklıkla kullanılmaktadır. Yerel navigasyonda ise robotun konumu ve yönüne harici sensörler vasıtasıyla karar verilmektedir. Bu navigasyon, çevrim içi mod olarak da bilinir ve kızılötesi sensör, ultrasonik sensör, lazer ve kamera gibi harici ekipmanlardan faydalanır. Burada,

sensörlerden gelen bilgiye göre robotun yönünü otomatik olarak belirlemek ve değiştirmek için YSA, BM, ANFIS, PSO ve GA gibi yöntemler kullanılmaktadır. Ayrıca, mobil robot navigasyonu robotik alanda en çok çalışılan konulardan bir tanesidir ve ulaşım, endüstri ve kurtarma robotları gibi farklı uygulamaları da kapsamaktadır. Yol planlama uygulamaları araştırmacıların son yirmi yıldır üzerinde çalıştığı, otonom mobil robot navigasyonu konusunun en önemli parçalarından biridir. Yol planlama işlemi, gidilecek yolun toplam miktarını en aza indirirken çarpma durumunun da önüne geçilmesine dayanır. Burada, mobil robot hareketi esnasında engellerin zamana göre sadece konumlarının değiştiği durumlarda statik yol planlaması hem konumlarının hem de eğilimlerinin değiştiği durumlarda dinamik yol planlaması uygulanır (Gul ve ark., 2019).

Malyejev ve arkadaşları (2009) çalışmalarında; küresel konum belirleme sistemi (GPS), kilometre sayacı ve elektronik pusulaya sahip bir insansız kara aracının konum tahmini problemini ele alınmıştır. Kokusuz Kalman filtresi (UKF), sensörlerden gelen bilgileri birleştirmek için kullanılmıştır. Doğruluğu ve sağlamlığı artırmak için, elektronik pusula çıktısını kalibre etme yöntemini ayrıca önermiştir. Sonuç olarak kısa GPS kesintilerinde bile önerilen yöntemin faydalı olabileceği görülmektedir (Malyavej ve ark., 2009).

Sariff ve Buniyamin (2006) çalışmalarında; bir robotun ortamda gezinmesi için en uygun yolu üreten algoritmalara odaklanan otonom mobil robot yol planlamasına genel bir bakış sunmuştur. Gezinme görevini tamamlamak için, algoritmalar ortamın veya çalışma alanının haritasını okuyacak ve ardından robotun nesnelere ve engellere çarpmadan çalışma alanından geçmesi için serbest yollar üretmeye çalışılmaktadır. Önceki araştırmalardan elde edilen simülasyon ve deneysel sonuçlar, algoritmaların otonom robot navigasyonu için optimal bir yol (kısa, pürüzsüz ve sağlam) üretmek için önemli bir rol oynadığını ve aynı zamanda uygun algoritmaların zaman alıcı problemler olmadan pratik olarak kullanılabilir kadar hızlı çalışabileceğini kanıtlanmıştır (Sariff ve Buniyamin, 2006).

Hachour (2008) çalışmasında; bilinmeyen bir ortamda mobil robotun hedefe yönelik yol planlaması için yeni bir algoritma sunmuştur. Bu algoritma ile mobil robotun sabit engeller arasında gezinmesi ve hedefe engellere çarpmadan ulaşmak için yolu bulması sağlanmıştır. Bu algoritma ile robota başlangıç konumundan son konuma hareket etme imkânı sağlanılmıştır. Yol bulma stratejisi, sabit bilinmeyen engellerle birlikte bilinmeyen bir ortamın ızgara haritası biçiminde tasarlanmıştır. Robot, hedefe doğru

karşısına çıkan engelleri algılayarak ve kaçınarak bilinmeyen ortam içinde hareket edebilmektedir. Görev gerçekleştirilirken, yoldaki engellerden kaçınarak ve zaman, enerji ve mesafe gibi maliyetleri en aza indirerek mobil robot için en uygun veya uygulanabilir bir yol tasarlatılması gerekmektedir (Hachour, 2008).

Bayram ve arkadaşları (2023) çalışmalarında; belirli sensörler yardımıyla prototip tasarımı yapılmış insansız kara aracının geri besleme kontrolü ile yol takibini gerçekleştirmiştir. Mobil robotun yörünge takibi için model öngörülü kontrol yöntemi kullanılmıştır. Bu kontrol yöntemi ile elde edilen bir referans yolda hatasız bir şekilde ilerleyen hayali bir araca göre konum ve yönelme hatalarına göre direksiyon açısının elde edilmesi sağlanmıştır. Takip edilen yol bilinen geometrik fonksiyonlar ve bazı eğriler ile tasarlanmıştır. Deneyler ile hatanın uygun sınırlar içerisinde kaldığı gösterilmiştir (Bayram ve ark., 2023).

Hassani ve arkadaşları (2018) çalışmalarında; serbest segmentlere dayalı geliştirilmiş bir algoritma ve bir dönüm noktası stratejisi ile statik bir ortamda robot yol planlama probleminin çözümünü gerçekleştirmiştir. Mobil robot için güvenli bir yol aramak ve robotu önüne çıkan engellere çarpmadan istenilen hedefe hareket ettirmek için dönüm noktası yaklaşımı benimsenmiştir. Önerilen algoritmada yolun güvenliği ve uzunluğu olarak iki hedef belirlenmiştir. Mobil robotun etkili hareketi için ise kayan kipli kontrol önerilmiştir. Sonuç olarak önerilen yaklaşım sayesinde istenilen hedefe, çarpışma olmaksızın ulaşılabilir (Hassani ve ark., 2018).

Aki ve Dirik (2020) çalışmalarında; derin öğrenme tabanlı PID dürüş kontrol ile otonom sürüş yapabilen bir simülasyon aracı tasarlamışlardır. Aracın direksiyon açısı, hız bilgisi ve aracın ön kısmına sabitlenen 3 adet kamera vasıtasıyla gerekli veriler alınarak derin öğrenme ile PID kontrolü sağlanmıştır. Simülatörün içindeki pistte bir tam tur atarak aracın sınırları belirlenmiştir. PID ile sürüşün başarılı olmasının yanında derin öğrenme modeli ile başarı oranı artırılmıştır (Aki ve Dirik, 2020).

Şanlı (2018) çalışmasında; RC kontrollü bir araç üzerine konumlandırılmış bir kamera ile yapay sinir ağları kullanarak aracı yolda stabil hareketini sağlamayı amaçlamıştır. Aracın şeritte doğru hareketi için RPI geliştirme kartı ile kameradan elde edilen veriler Python programlama dili kullanarak TensorFlow tekniği ile görüntü işlenmiştir (Şanlı, 2018).

Gürtaş (2018) çalışmasında; genel olarak literatür taraması şeklinde yapay zekanın otonom araçlar üzerindeki etkisini araştırmış olup özelde yapay zekâ ile ACC

yani önündeki aracın hızına ivmesi ve durmasına bağlı olarak lidar sensörler ile takibin sağlanmasını araştırmıştır (Gürtaş, 2018).

Bingöl ve arkadaşları (2019) çalışmalarında; NVIDIA TX2 geliştirme kullanarak otonom bir sürüşe sahip ackerman prensibi ile çalışan bir araç tasarlamışlardır. Yol bilgisi lidar sensör ve stereo kamera ile sağlanmıştır. Bu şekilde evrimsel sinir ağları (CNN) ve uzun kısa-vade hafıza ağları (LSTM) kullanılarak aracın en iyi sürüş deneyimi ile hedefe ulaşması sağlanmıştır (Bingöl ve ark., 2019).

Beşdok ve Özçelik (2009) çalışmalarında; GPS'in dezavantajları olan anlık yanlış bilgilendirme ve uydudan kısa süreli kopmaların önüne geçmesi için Amerikan matematiksel sistem teoristi Rudolf Kalman tarafından geliştirilen kalman filtresi ile bu saplamaları incelemek üzere birtakım simülasyonlar ile otonom hava aracı navigasyonları incelemiştir (Beşdok ve Özçelik, 2009).

Nennioğlu ve Köroğlu (2018) çalışmalarında; otonom araçların gittikçe yaygınlaştığı günümüzde, otonom sürüş için önemli işlevlerden biri olan hareket planlaması konusunda çeşitli teknikler üzerinde literatür araştırması yapmıştır (Nennioğlu ve Köroğlu, 2018).

Çetinkaya (2017) çalışmasında; bulanık mantık kontrolüyle Arduino geliştirme kartı kullanarak prototip araç üzerine yerleştirdiği çeşitli çevre birimi ile (servo, ultrasonik mesafe sensörü, vb.) C# tabanlı bir uygulama ile uzaktan kontrollü bir aracın bulanık mantık algoritması ile pist üzerinde hareketini sağlamıştır (Çetinkaya, 2017).

2.1 Tezin Amacı

Sahip olduğu sevk ve idare etme yeteneğini geliştirdiği teknolojik aygıtlara entegre etmek isteyen insanoğlunun, son dönemlerde üzerinde oldukça fazla zaman harcadığı konulardan biri de otonom araçlardır. Otonom araçların sahip olması beklenen en temel özelliklerden birisi varılmak istenilen hedefe ulaşmak için insan zekasını taklit ederek rotasını belirlemesidir. Örnek vermek gerekirse insan gideceği hedefe ulaşmak için önce hangi yolu izleyeceğini belirler ve daha sonra belirlediği rota düzeninde engelleri (örneğin kaldırımlar, karşıdan karşıya geçen bir yaya, önüne manevra yapan bir araba) dikkate alarak en güvenli şekilde hedefe doğru hareket eder. Karşısına çıkan engelleri aşamaması durumunda (örneğin yoğun trafik, caddenin yol çalışması nedeniyle kapalı olması) hedefe ulaşmak için yeni bir rota belirler. Bu tez çalışmasında otonom kara aracı kontrolü için Android uygulama ile akıllı telefon üzerinden araca konum

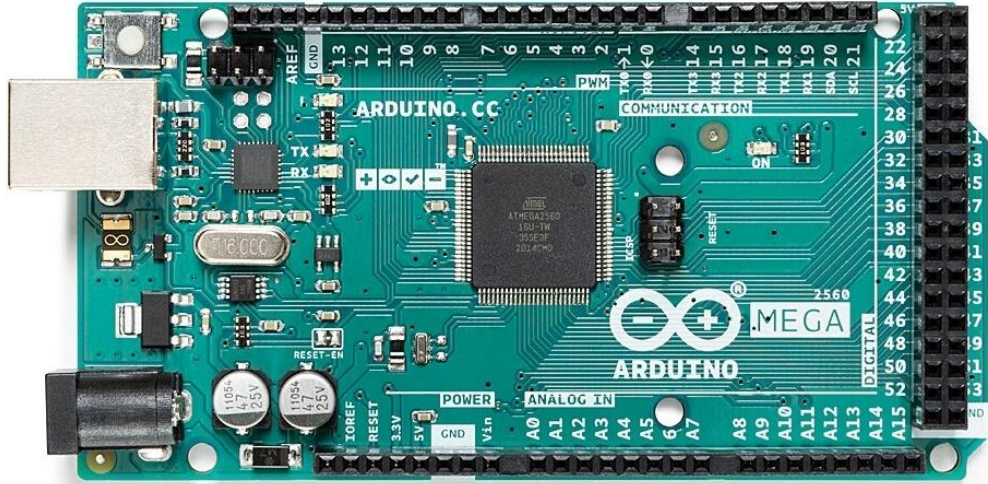
gönderilerek hedefe ulaşmak üzere yön tayini ve engelden kaçma algoritmaları uygulaması gerçekleştirilmiştir.

3 MATERYAL

Bu çalışmada otonom araç kontrolü gerçekleştirmek üzere materyal olarak; hobi amaçlı kara aracı platformu ve elektronik devre elemanları (Arduino Mega kontrol kartı, çeşitli sensörler ve eyleyiciler) kullanılmıştır. Bu materyallerin teknik özellikleri hakkında aşağıda detaylı olarak bilgi verilmiştir.

3.1 Arduino Kontrol Kartı

Otonom kara aracı uygulamasında kontrol ünitesi olarak Arduino Mega kartı kullanılması tercih edilmiştir. Arduino kütüphane zenginliği, kaynak erişim kolaylığı ve maliyet avantajı sebebiyle birçok uygulamada yaygın olarak kullanılmaktadır. Şekil 3.1’de gösterildiği gibi Arduino Mega kontrol kartı üzerinde ATmega2560 mikrodenetleyici bulunmaktadır. Ayrıca mikrodenetleyiciyi destekleyen çeşitli yan donanım ve devre elemanları vardır. Bu devre elemanları arasında kristal osilatör (bazı modellerde seramik rezonatör) ve 5V regülatör devresi yer almaktadır. Üzerinde dahili bootloader programı yer aldığı için Arduino Mega kartının programlanması için harici bir programlayıcıya ihtiyaç duyulmaz (Arduino, 2023).



Şekil 3.1 Otonom sürüş için kullanılan Arduino Mega kontrol kartı

Arduino Mega kontrol kartı için yazılım tasarımı çalışmalarında kullanılmak üzere kendi entegre geliştirme ortamı (IDE) mevcuttur. Bunun haricinde Java tabanlı Processing dili kullanılması mümkündür. Arduino kartların yaygın kullanılmasındaki

sebeplerden biri de geniş kütüphane desteği sayesinde uygulama geliştirme kolaylığıdır. Ayrıca Şekil 3.2’de gösterildiği gibi çok sayıda Arduino uyumlu sensör mevcuttur. Bu tez çalışmasında engelden kaçış için ultrasonik mesafe sensörü, konum belirleme için GPS modülü ve yön tayini için IMU modülü kullanılmıştır.



Şekil 3.2 Arduino Mega kontrol kartıyla birlikte kullanılacak sensörler

3.2 SG90 Servo Motor

Tez çalışmasında gerçekleştirilen otonom araç yön kontrolü uygulamasında yüksek çıkış gücü ile birlikte hem hafif hem ergonomik yapısından dolayı SG90 servo motor kullanılmıştır. SG90 servo motor sayesinde aracın 0° ile 180° derece arasında dönüşü yüksek torkta sağlanabilir. SG90 servo motor kontrolü için karmaşık kablo gruplarına gerek yoktur. Yaygın kütüphane desteği ile Arduino kontrol kartlarıyla rahatlıkla kontrol edilebilir. SG90 servo motor ekstra bir dişli sistemine ihtiyaç duyulmadan standart kutu içeriği kontrol kartına kolayca bağlanabilir (Towe, 2017).



Şekil 3.3 Otonom sürüş yön kontrolü için kullanılan SG90 servo motor

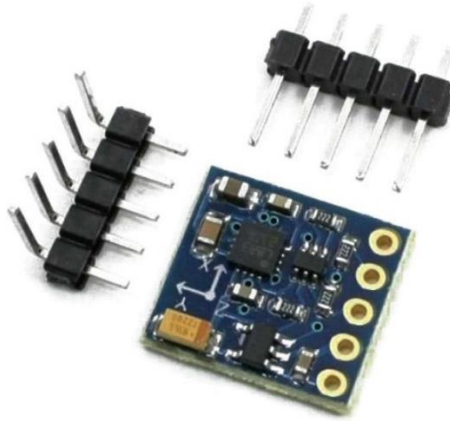
Çizelge 3.1 SG90 servo motor özellikleri

Ağırlık	9 gr
Boyutlar	22,2 mm X 11,8 X 31 mm
Stall Torku	1.8 kgf·cm
Çalışma Hızı	0.1 s/60 derece
Çalışma Voltajı	4.8 (~5)
Ölü Bant Genişliği	10 µs
Sıcaklık Aralığı	0 °C – 55 °C

3.3 HMC5883L Manyetometre

Otonom sürüş kontrolünde aracın hedefe ulaşabilmesi için hassas yön tayini yapabilmesi gerekir. Yapılan uygulamada bu yeteneği sağlamak için dünyanın manyetik alanını algılayabilen HMC5883L manyetometre kullanılmıştır. Manyetometre, manyetik alanı algılayarak anlık manyetik alan büyüklüğüne göre yön tayini yapmaktadır. Yön tayini yapılırken kuzey kutbu referans alınarak araç hareket yönü belirlenmiştir. Şekil 3.4'te HMC5883L manyetometre içeren GY-271 modülü gösterilmiştir.

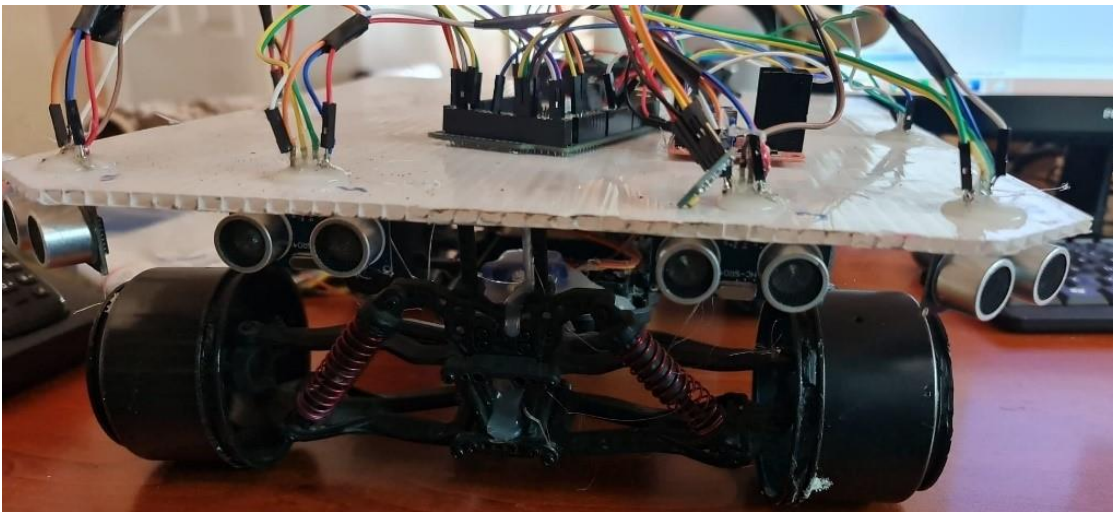
HMC5883L manyetometre için izin verilen maksimum besleme gerilimi 3.6 V olup mikrodenetleyici ile 3.3 V uygulanarak çalıştırılabilir. Veri alışverişi için I2C haberleşme protokolü kullanılmaktadır. Yön tayini yapılmadan önce bazı konfigürasyon ayarlarının yapılmasına ihtiyaç vardır. HMC5883L ilk çalıştığında I2C protokolü ile konfigürasyon ayarlarının yapılması gerekir. Daha sonra üç farklı eksen için ayrı ayrı anlık manyetik alan kuvveti değerleri okunabilir.



Şekil 3.4 Otonom araç kontrolü yön tayini için kullanılan HMC5883l manyetometre

3.4 HC-SR04 Ultrasonik Mesafe Sensörü

Otonom araç hedefe ilerlerken önüne çıkan engelleri aşabilmesi gerekmektedir. Bunun için öncelikle engellerin algılanması ve araca olan uzaklıklarının ölçülmesine ihtiyaç vardır. Yapılan çalışmada engel algılama ve uzaklık ölçümü için düşük maliyetli HC-SR04 ultrasonik mesafe sensörü kullanılmıştır. HC-SR04 sensörü besleme ve veri aktarımı için dört adet pine sahiptir. Bu pinlerden ikisi besleme gerilimi uygulanması için kullanılan VCC ve GND iken, diğerleri karşılaşılan engellere olan mesafeyi ölçmek için sinyalleşmeyi yöneten TRIGGER ve ECHO pinleridir. TRIGGER tetikleme, ECHO ise yankı sinyali için kullanılır (Elec, 2017). Şekil 3.5’de gerçekleştirilen uygulamada araç üzerinde HC-SR04 sensörü kullanımı gösterilmiştir.

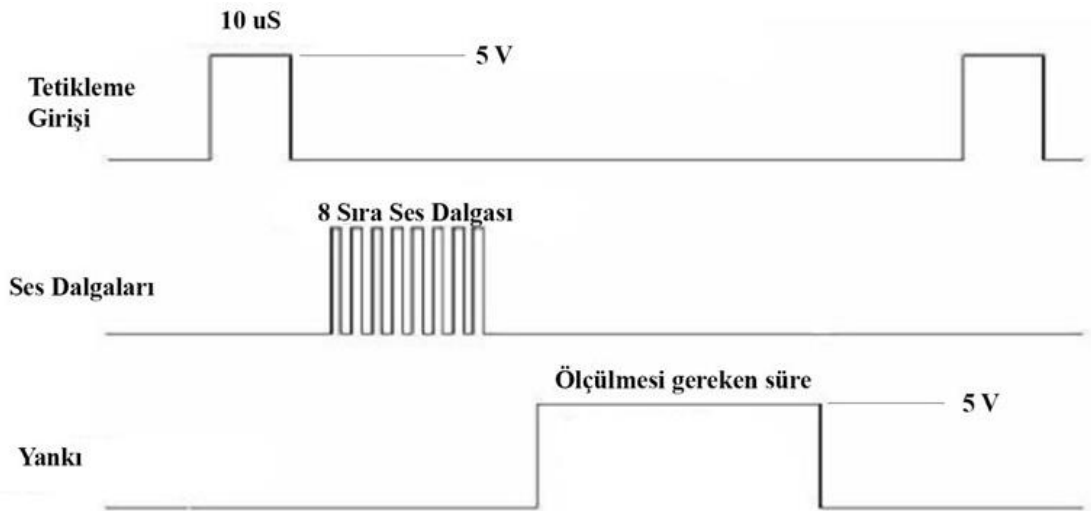


Şekil 3.5 Otonom araç kontrolü için hc-sr04 ultrasonik mesafe sensörü kullanımı

HC-SR04 sensörü 5 V besleme gerilimi ile çalışmaktadır. En az 2 santimetre mesafeyi ölçerken, en fazla 4 metre mesafeden ölçüm yapabilmektedir. Mesafe ölçümünü başlatmak için tetikleme girişine 10 μ S darbe gönderilmesi gerekmektedir. Ardından sensör 40 kHz'de bir ses sinyali gönderir ve yankı pinini etkinleştirir. Gönderilen ses sinyali engele çarpıp geri döndüğünde yankı pininden yakalanır. Bu süreçte geçen zaman hesaplanarak engele olan mesafe bulunur. HC-SR04 sensör ile mesafe ölçümünde kullanılan zamanlama diyagramı Şekil 3.6'da gösterilmiştir. Ayrıca HC-SR04 sensörüne ait çalışma özellikleri Çizelge 3.2'de verilmiştir (Elec, 2017).

Çizelge 3.2 HC-SR04 ultrasonik mesafe sensörü özellikleri

Çalışma Voltajı	DC 5V
Çalışma Akımı	15mA
Çalışma Frekansı	40Hz
Maksimum Ölçme Mesafesi	4m
Minimum Ölçme Mesafesi	2cm
Maksimum Açısı	15 Derece
Trigger giriş Sinyali	10us TTL Darbe



Şekil 3.6 HC-SR04 ultrasonik mesafe sensörü mesafe ölçüm zamanlama diyagramı

Tetikleyici girişinde mikro saniye mertebesinde sinyal üretilebilmesi için mikrodenetleyici zamanlayıcı (timer) birimi kullanılmaktadır. Böylece 10 μ S süre ile tetikleme pinine çıkış verilir. Yankı pininin giriş yüksek seviyede olduğu süreyi

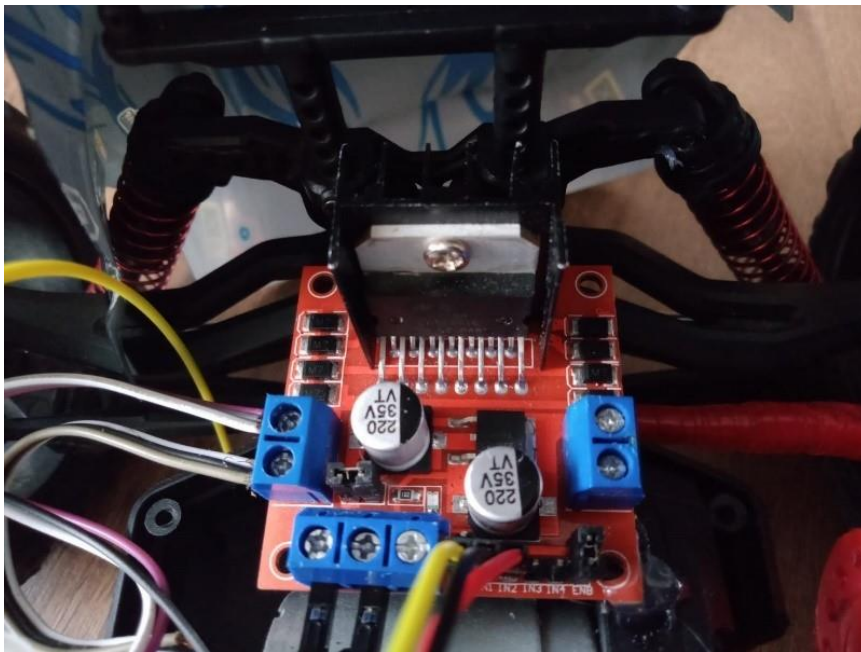
ölçebilmek için ise mikrodenetleyici sinyal yakalama (capture) birimi kullanılmaktadır. Mikrodenetleyici tarafından ölçülen frekans değeri, zamana dönüştürülerek geçen süre hesaplanır. Bu süre daha sonra ses hızı ile çarpılır ve sinyal gidip gelirken aynı mesafeyi kat ettiğinden ikiye bölünür:

$$\frac{t_e * 34000}{2} \quad (1)$$

Yukarıda verilen formülasyonda t_e saniye olarak geçen zamandır. Yapılan hesaplama ile engel ile araç arasındaki mesafe santimetre (cm) olarak ölçülür (Elec, 2017).

3.5 L298N Motor Sürücü Modülü

Otonom araç tahrik motorunun kontrol edilebilmesi için motor sürücüyeye ihtiyaç vardır. Yapılan çalışmada L298N motor sürücü entegresi içeren motor sürücü modülü kullanılmıştır. Kullanılan motor sürücü sayesinde motor yön ve hız ayarı yapılmıştır. L298N motor sürücü 12 V besleme gerilimi ile aynı anda iki farklı motoru sürülebilmektedir. Kanal başına 2 A akım çıkış sağlayabilmektedir. Şekil 3.7’de otonom araç üzerinde kullanılan L298N motor sürücü modülü gösterilmiştir.



Şekil 3.7 Araç üzerinde kullanılan L298N motor sürüsü modülü

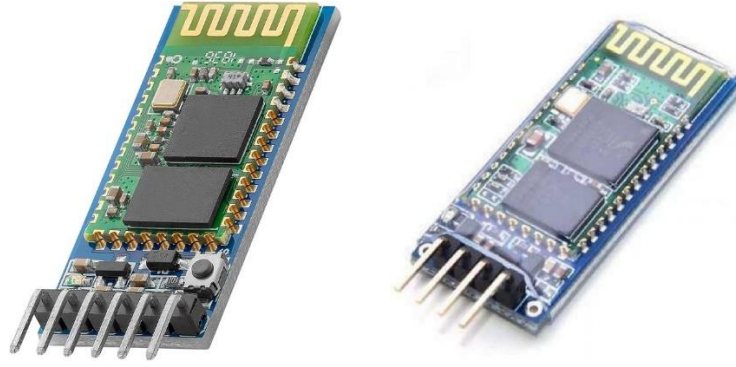
Motor sürücü modülünün çalışması için girişine gerekli besleme gerilimi verilir. Hangi kanal kullanılacak ise ilgili kanalın yetki pinine üzerinden PWM çıkış sinyali üretilir. PWM sinyali için görev çevrim (duty cycle) değeri değiştirilerek çıkış gerilimi ayarlanmış olur. Motor yönünü ayarlamak için ise her kanal için iki adet giriş pini bulunmaktadır. İlgili pinler yüksek seviyeye getirilerek motorun dönüş yönü belirlenir.

3.6 HC-06 Bluetooth Haberleşme

Yapılan otonom kontrol uygulamasında ulaşılabilecek hedef koordinat bilgisinin araca gönderilmesi için haberleşme protokolü olarak Bluetooth kullanılmıştır. Bu şekilde Android işletim sistemine sahip akıllı cep telefonu için geliştirilen mobil uygulama ile hedef koordinat bilgisi araca Bluetooth bağlantısı üzerinden gönderilmiştir. Bluetooth teknolojisi kısa mesafede veri aktarımı ihtiyaçlarını karşılamak için geliştirilmiş bir haberleşme protokolüdür. Bluetooth çalışma frekansı olarak 2.4 GHz ISM bandını (2400-2483.5 MHz) kullanır. Alıcı ile verici arasında doğrudan görüş hattı olması durumunda veri aktarım menzili verici gücüne bağlı olarak 100 metreye kadar çıkabilir. Bluetooth haberleşme için pratikte genellikle HC-05 ve HC-06 olmak üzere iki farklı modül kullanılmaktadır. Şekil 3.8’de gösterilen bu iki modül aralarındaki fark, usta (master) ve çirak (slave) çalışma modu yeteneklerinden kaynaklanmaktadır. HC-05 hem kendisine gelen bağlantı isteklerine cevap verirken, hem de başka cihazlara bağlantı isteği gönderebilir. HC-06 ise yalnızca kendisine gelen bağlantı isteklerini cevaplayabilir, başka bir cihaza bağlantı isteği yollayamaz. Bu temelde HC-05 hem usta hem de çirak modunda çalışabilirken, HC-06 sadece çirak modunda çalışabilmektedir. HC-05 ve HC-06 modüllerinin çalışma özellikleri Çizelge 3.3’de verilmiştir

Çizelge 3.3 HC-06 Bluetooth modül özellikleri

Çalışma frekansı	2,4 GHz ISM bandı
Hassasiyet	≤ -80 dBm
Çıkış gücü	$\leq +4$ dBm
Asenkron hız	2,1 Mbps / 160 Kbps
Senkron hız	1 Mbps / 1 Mbps
Akım	50 mA



Şekil 3.8 Hedef konum bilgisi gönderimi için kullanılan HC-06 Bluetooth modülü

Yapılan çalışmada cep telefonu ile bağlantı sağlamak üzere araç üzerinde HC-06 Bluetooth modülüne yer verilmiştir. HC-06 Bluetooth modülü 3.3V besleme gerilimi ile çalışmaktadır. Modül üzerinde VCC, GND, Rx ve Tx olmak üzere 4 adet pin bulunmaktadır. Arduino Mega kontrol kartı ile haberleşme için Arduino Mega kartı Tx pini ile HC-06 modülü Rx pini ve Arduino Mega kartı Rx pini ile HC-06 modülü Tx pini birbirlerine bağlanmalıdır (Geleceği Yazanlar Ekibi, 2020).

3.7 Güç Kaynağı

Otonom araç kontrolü için tüm donanımları besleyebilecek gerilim kaynağına ihtiyaç vardır. Şekil 3.9’da gösterilen 3.7 V Li-Ion pillerden üç adet birbirlerine seri bağlanarak araç üzerinde 11.1 V besleme gerilimi elde edilmiştir.



Şekil 3.9 Araç üzerinde güç kaynağı olarak kullanılan lityum-iyon pil

3.8 GY-NEO6MV2 GPS Modülü

GPS (Global Positioning System) dünya yörüngesindeki uydular yardımı ile konum belirlemek için oluşturulmuş küresel konumlandırma sistemidir. İlk olarak askeri amaçla kullanılmış olsa da sonrasında sivil uygulamalar için de kullanıma açılmıştır (Karaali ve Yıldırım, 2020). Bir nesnenin dünya üzerinde coğrafi konumunu hassas şekilde belirleyebilmek için enlem ve boylam olarak tanımlanan koordinat bilgisinden faydalanılır. Enlemler dünyayı ekvatora paralel olarak keserken, boylamlar Dünya'nın kuzey kutbundan güney kutbuna doğru dikey şekilde kesmektedir. Enlem ve boylam bilgisine bağlı olarak konum derece şeklinde ifade edilir (Hülako ve Kapucu, 2018).



Şekil 3.10 Otonom araç kontrolünde konum belirleme için kullanılan GPS modülü

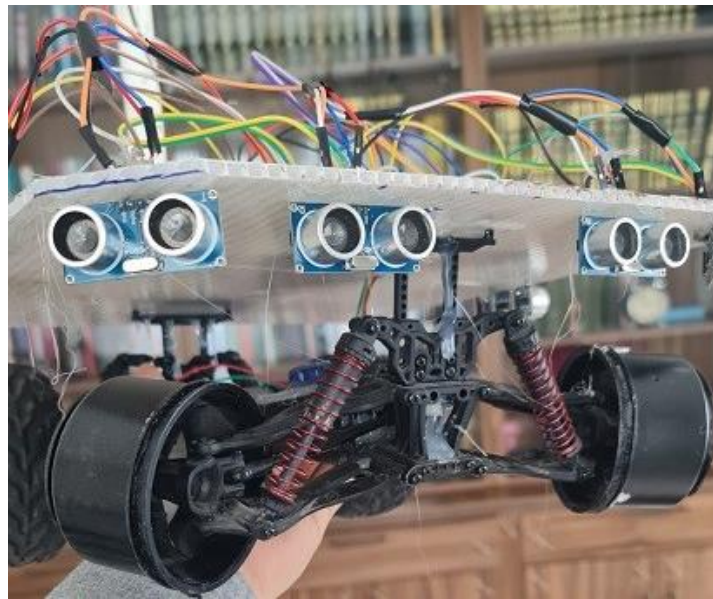
Yapılan çalışmada araç anlık konum bilgisinin belirlenmesi için düşük maliyetli GY-NEO6MV2 GPS modülü kullanılmıştır. Bu modül 3.3-5 V besleme gerilimi ile çalışmaktadır. UART haberleşme protokolü kullanarak NEO6MV2 GPS modülü ile haberleşme yapılmaktadır. GY-NEO6MV2 modülü ve anten bağlantısı Şekil 3.10'da görülmektedir. GPS modüle bağlanan anten yardımıyla uydulardan gönderilen sinyaller alınabilir. GPS sisteminde konum belirleme için en az 4 uydu ile bağlantıya ihtiyaç vardır. Uydulardan gönderilen konum belirleme sinyalleri NMEA (national marine electronic association) cümleleri şeklinde alınmaktadır. GPS modülü bağlandığı uydulardan NMEA cümlelerini aldıktan sonra mikrodenetleyiciye gönderir. Bu şekilde araç anlık konum bilgisinin belirlenmesi sağlanır (Hülako ve Kapucu, 2018).

4 METOT

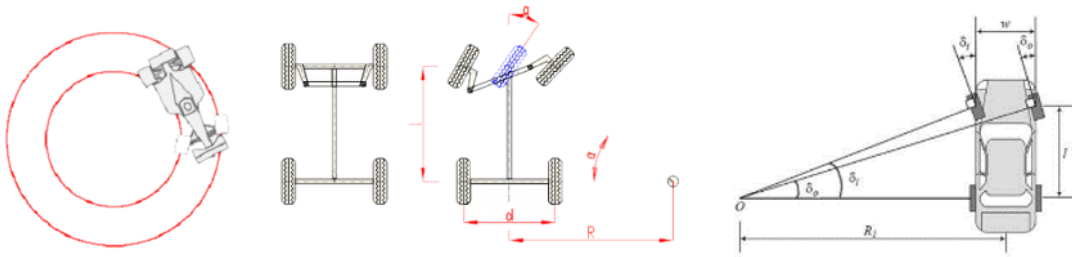
4.1 Ackerman Prensipli ile Direksiyon Kontrolü

Kara araçları için yön kontrolünde iki farklı yönlendirme sistemi kullanılmaktadır. Bunlardan birincisi genelde ağır arazi şartlarında hareket eden araçlar için kullanılan palet sistemidir. Bu sistemde yönlendirme iki paletin birbirine göre hareketlenmesiyle sağlanır. Bu çalışmada ağır arazi şartları dışında otomobil ve benzeri taşıtlarda sıklıkla tercih edilen aks-rot destekli geleneksel direksiyon sistemi kullanılmıştır. Literatürde farklı şekillerde gerçekleştirilen direksiyon kontrol yöntemleri bulunmaktadır.

Yapılan çalışmada direksiyon kontrolü için Ackerman prensibi kullanılması tercih edilmiştir. Ackerman prensibi, aracın hedefe ulaşması için direksiyon sisteminde hangi tekerinin kaç derece açı ile dönüş yapması gerektiğini hesaplamayı sağlar. Dört tekerli araçlarda direksiyon sistemi dönüşü sağlayan her tekere aynı açı ile hükmederse, dışarda kalan tekerler daha da dışarıya doğru ötelenir. Bunu engellemek için Ackerman prensibi kullanılarak virajı dışarıdan alan tekerler ile içeriden alan tekerler için farklı dönüş açıları hesaplanır. Böylece dışardan dönen tekerler için öteleme olmaksızın kontrollü bir şekilde dönüş hareketi sağlanır. Otonom sürüş kontrolünde tasarlanan prototip araç üzerinde dönüş kontrolü için kullanılan direksiyon sistemi Şekil 4.1’de gösterilmiştir.



Şekil 4.1 Tasarlanan prototip araç için ackerman prensibi ile direksiyon kontrolü



Şekil 4.2 Ackerman prensibi ile direksiyon kontrolünde teker dönüş açısı hesabı

Ackerman prensibi ile araç dönüş kontrolü sağlandığında iç ve dış tekerler için dönüş açıları arasındaki farklılık Şekil 4.2’de gösterilmiştir. Araç geometrisi (araç genişliği, ön ve arka aks mesafesi) ve dönülen viraj yarıçapına bağlı olarak iç ve dış teker dönüş açıları aşağıda verilen denklemler kullanılarak hesaplanır (Özden, 2018):

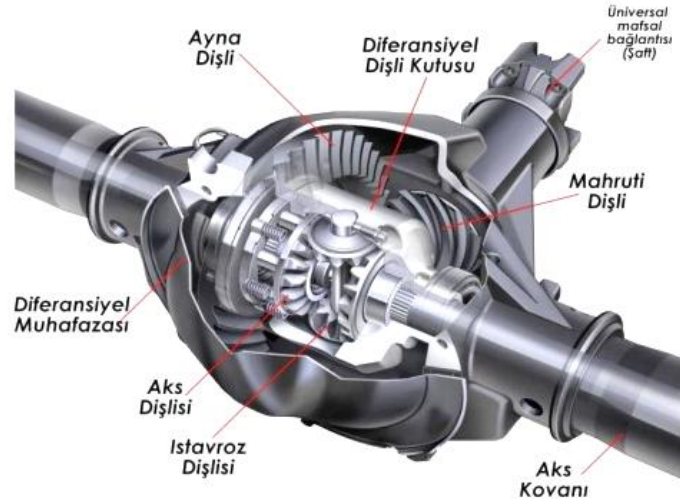
$$\tan \delta_i = \frac{l}{R_1 - \frac{w}{2}} \quad (2)$$

$$\tan \delta_o = \frac{l}{R_1 + \frac{w}{2}} \quad (3)$$

4.2 Tahrik Sistemi

Bu tez çalışmasında araç üzerinde tek bir doğru akım (DC) motoru kullanılarak hareket sağlanmıştır. DC motor, aracın ağırlığını kaldırabilecek torkta ve düşük hızda seçilmiş olup iki arka tekeri döndürebilmek için diferansiyel sistemi kullanılmıştır. Yapılan çalışmada kullanılan diferansiyel sistemi yapısı Şekil 4.3’te gösterildiği gibi farklı amaçlar için dişli çarklar içermektedir.

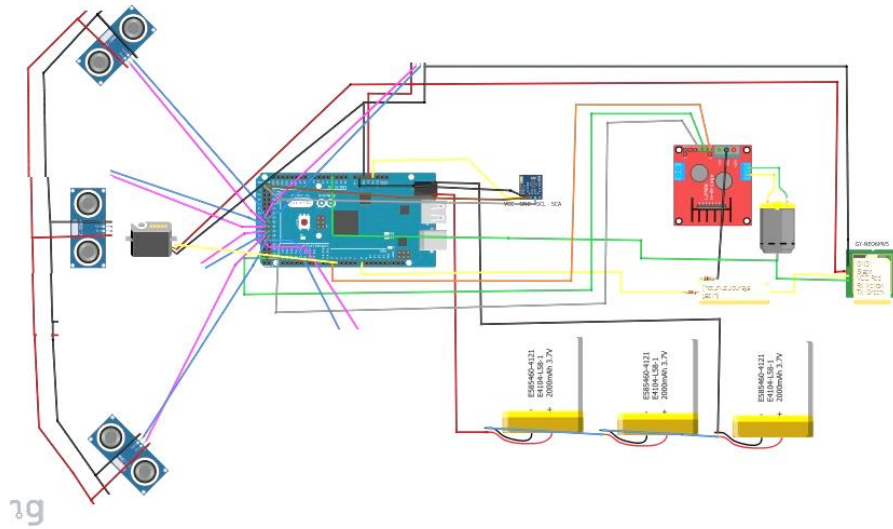
İki bisiklet sürücü yan yana düz yolda ilerlerken aynı açısal hıza sahiptir. Bu sürücüler viraj ile karşılaştıklarında, içeride kalan bisiklet sürücüsü dışardakine göre daha az yol alacaktır. Sürücüler yan yana virajı dönmek isterlerse, dışarıdaki sürücü daha fazla açısal hıza sahip olmalıdır. Benzer şekilde sürücülerden biri yol üzerinde hareketini kısıtlayan bir engelle karşılaştığında, birlikte yol alabilmeleri için daha çok tork üretmeye ihtiyaç duyar. Bu durum dört tekerli araçlar için de geçerlidir. Karşılaşılabilecek benzer problemlerin çözümü için araç üzerinde diferansiyel sistemi kullanılmıştır (Özden, 2018).



Şekil 4.3 Yapılan çalışmada kullanılan diferansiyel kutusu bloğu iç yapısı

4.3 Donanım Tasarımı ve Yazılım Geliştirme

Yapılan çalışmada otonom araç kontrolü için 6 adet HC-SR04 ultrasonik mesafe sensörü, 1 adet L298N DA motor sürücü modülü, 1 adet HMC5883L manyetometre, 1 adet GY-NEO6MV2 GPS modülü, 1 adet HC-06 Bluetooth haberleşme modülü ve son olarak sistemin giriş çıkış kontrolünü sağlamak amacıyla Arduino Mega kontrol kartı kullanılmıştır. Otonom araç kontrolünde kullanılan donanım birimlerinin Arduino Mega kontrol kartı ile bağlantıları Şekil 4.4'da gösterilmiştir.



Şekil 4.4 Kullanılan donanım birimlerinin kontrolör ile bağlantısı

4.3.1 Ultrasonik mesafe sensörü ile engelden kaçma

Otonom sürüş sırasında aracın karşılaştığı engellerin algılaması için HC-SR04 ultrasonik mesafe sensörü kullanılmıştır. Aracın sağ tarafında karşılaştığı engellere olan uzaklığının hesaplanması için oluşturulan kod bloğu aşağıda verilmiştir.

```
void engeldenetle(){
    delayMicroseconds(2);
    digitalWrite(trigs_5, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigs_5, LOW);
    sure_5 = pulseIn(echos_5, HIGH);
    sag_sensor = sure_5 * 0.034 / 2;
    delay(50);
}
```

Hazırlanan fonksiyon içerisinde 2 mikrosaniye bekleme sonrasında ultrasonik mesafe sensöründen 10 mikrosaniye süre ile ses sinyali gönderilir. Bu süre sonunda ses sinyali gönderimi durdurulur ve dinleme modu aktif edilerek engele çarpıp geri dönen ses sinyali referans alınarak arada geçen zaman “sure_5” değişkeni içinde saklanır. Ses sinyalinin 25°C’de ortam sıcaklığında hareket hızına bağlı olarak engel ile araç arasındaki mesafe hesaplanır ve “sag_sensor” değişkeni içinde saklanır. Araç üzerinde bulunan diğer ultrasonik sensörler için benzer şekilde engele olan uzaklık belirlenir. Tüm ultrasonik mesafe sensörleri için engellere olan uzaklık değerleri belirlendikten sonra karşılaşılan duruma göre engelden kaçış için gerekli olan dönüş açısı belirlenir.

Hazırlanan kod bloğunda üç mesafe sensöründen herhangi birisi için engele olan uzaklık önceden belirlenen eşik değerinden küçük olduğunda, karşılaşılan engelden kaçış için bulanık mantık tabanlı dönüş açısı belirleme fonksiyonu çalıştırılır ve elde edilen “aci” değeri ile direksiyon sistemini kontrol eden servo motor belirlenen açıya döndürülür. Dönüş açısı belirlendikten sonra L298N motor sürücü modülü üzerinde motor yön belirleme pinleri aktif edilir ve aracın ileri hareketi sağlanır.


```

if(sag_sensor < 20 || orta_sensor < 40 || sol_sensor < 20 ) {
    kurallar (sol_sensor,orta_sensor,sag_sensor);    }
servo.write(aci);
digitalWrite(uc1, HIGH);
digitalWrite(uc2, LOW);
analogWrite(hiz, 255);

```

Engele olan mesafe ölçümü için kullanılan sensörlerin tümü için ölçülen değerleri 5 cm'den daha küçükse, aracın engeller arasında sıkıştığı ve dönüş yaparak engellerden kaçamayacağı kabul edilir. Bu durumda aracın geri hareketi sağlanır.

```

if(sol_sensor < 5 && orta_sensor < 5 && sag_sensor < 5){
    servo.write(90);
    digitalWrite(uc1, LOW);
    digitalWrite(uc2, HIGH);
    analogWrite(hiz, 255);
    delay(3000);
}

```

4.3.2 Manyetometre ile yön belirleme

Bu çalışmada aracın hedef olarak verilen konuma engelleri aşarak hareketinde manyetometre ile yön bulma için oluşturulan kod bloğu aşağıda verilmiştir:

```

void anlikacibulma(){
    sensors_event_t event;
    mag.getEvent(&event);
    float heading = atan2(event.magnetic.y, event.magnetic.x);
    float declinationAngle = 0.22;
    heading += declinationAngle;
    if(heading < 0) heading += 2*PI;
    if(heading > 2*PI) heading -= 2*PI;
    float mevcutaci = heading * 180/M_PI;
}

```

Manyometreden okunan değerler yukarıdaki kod satırında “atan2” fonksiyonu içerisine alınarak (x,y) koordinat sistemine göre başlangıç noktası (0,0) ve bitiş noktası “(event.magnetic.y, event.magnetic.x)” olan bir doğrunun yatay eksen ile yaptığı açı radyan cinsinden elde edilir ve bu açı değeri “heading” değişkeninin içine aktarılır. Açı değerinin elde edilmesinde aşağıdaki formülasyon kullanılır (Honeywell, 2023):

$$(X_{aciradyan} = X_{değer} + 2 * \pi) \quad (4)$$

$$(X_{aciradyan} = X_{değer} - 2 * \pi) \quad (5)$$

$$X_{aciderece} = X_{aciradyan} * (180/\pi) \quad (6)$$

Verilen formüllerde $X_{aciradyan}$ elde etmek istenilen açı değerinin radyan cisinden değerini, $X_{değer}$ değişkeni ise manyometreden okunan ham değeri temsil etmektedir.

Elde edilen veriler anlamlı birer ifade olmayacağından okunan değerlerin 0-360° aralığına normalize edilmesi gerekmektedir. Manyometreden okunan değerler arasından en küçük değer sıfır kabul edilerek “atan2(x,y)” fonksiyonu ile normalizasyonu sağlanmıştır. Manyometre değeri sıfırdan küçükse denklem (4), sıfırdan büyük ise denklem (5) kullanılarak radyan cinsinden açı değeri elde edilmiştir. Elde edilen değer radyan cinsinden olacağından denklem (6) kullanılarak radyan değeri dereceye çevrilmiş ve nihai açı değeri hesaplanmıştır (Honeywell, 2023). Sonuç olarak anlık açı değeri hesaplanarak 0-360° arasında aracın yönü belirlenmiştir. 0° batı, 90° kuzey, 180° doğu ve 270° güney yönünü göstermektedir.

4.3.3 Gps modülü ile konum belirleme

Araç konumu belirlenirken Arduino “TinyGPS.h” kütüphanesinin sunduğu özellikler kullanılarak enlem ve boylam bilgisi alınmıştır. Enlem ve boylam bilgisi elde edilirken kullanılan NMEA kelimeleri Çizelge 4.1’de verilmiştir.

Çizelge 4.1 Gps modülden okunan nmea kelimeleri

NMEA Kelimesi	Açıklama
GSV	Rakım, Güney Açısı, Bağlı Uydu Sayısı
RMC	Tarih, Saat, Enlem, Boylam, Yere Göre Hız
GSA	Bağlı Uyduların Numaraları
GGA	Saat, Enlem, Boylam, Bağlı Uydu Sayısı
GLL	Enlem, Boylam, Saat
VTG	Hız Bilgisi

GPS modülünden anlık konum bilgisi almak için kod satırları aşağıda verilmiştir.

```
void mevcutkonumbulma(){
    while(Serial1.available()){
        if(gps.encode(Serial1.read())){
            gps.f_get_position(&lat,&lon);
            mevcut_enlem=lat;
            mevcut_boylam=lon;
        }
    }
}
```

Kod satırlarında GPS yeterli sayıda uyduya bağlandığında “lat” ve “lon” değişkenleri içerisine koordinat değerlerini aktarması sağlanır. Devamında okunan enlem ve boylam değerleri “mevcut_enlem” ve “mevcut_boylam” değişkenleri içerisine aktarılarak aracının anlık konum bilgileri elde edilmiştir.

4.3.4 Hareket yönünün belirlenmesi

Araç yön açısı manyetometre ile belirlendikten sonra akıllı telefon üzerinde uygulama ile gönderilen hedef konuma doğru hareket etmesi için hareket yönünü değiştirmesi ve hedefe doğru istikamet açısını bulması gerekmektedir. Bu açıyı bulan kod satırları aşağıda verilmiştir.

```
void hedefacibulma(){
    float enlem_farki = abs(hedef_enlem – mevcut_enlem) * 111.1949;
```

```

float boylam_farki = 111.1949 * abs(hedef_boylam - mevcut_boylam) *
cos(radians((hedef_boylam + mevcut_boylam) / 2));
float mesafe = sqrt(pow(enlem_farki, 2) + pow(boylam_farki, 2));
double pi =3.14159;
double hedef_aci= (enlem_farki/boylam_farki)*(180/pi);
float kalibreli_mesafe=mesafe/1.12;
if(mevcut_enlem < hedef_enlem){
    if(mevcut_boylam < hedef_boylam)
        hedef_aci=hedef_aci;
    else
        hedef_aci=180-hedef_aci;
}
else
    if(mevcut_enlem > hedef_enlem){
        if(mevcut_boylam > hedef_boylam)
            hedef_aci=360-hedef_aci;
        else
            hedef_aci=180+hedef_aci;
    }
}

```

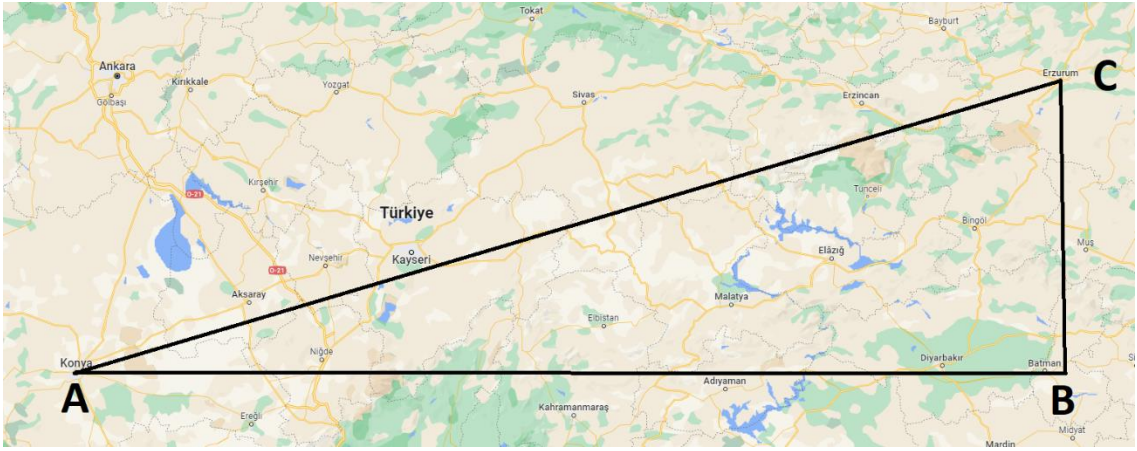
Yukarıda hedef açısının elde edilmesi için verilen kod satırlarında enlemler arasındaki farkın mutlak değeri alınarak her bir enlem arasındaki sabit fark olan 111.1949 km değeri ile çarparak enlemler arasındaki uzaklık km cinsinden bulunmuştur. Boylamlar arasındaki mesafenin yine mutlak değeri alınarak 111.1949 değeri ile çarpılmıştır. Fakat yer yüzeyinin küresel şeklinden dolayı her iki konumun boylamlarının toplamının yarısının radyan cinsinden kosinüsü alınarak işleme çarpım olarak eklenir ve boylamlar arasındaki mesafe km cinsinden bulunur. Bulunan bu enlemler arası mesafe ile boylamlar arası mesafe değerleri için Pisagor teorimi uygulanarak iki konum arası kuş bakışı mesafe elde edilir. Yer yüzeyinin şekli dikkate alınarak kuş bakışı mesafe 1.12 ile çarpılarak gerçek mesafe değeri elde edilir. Hedefin mevcut konuma göre durumu dikkate alınarak kartezyen koordinat sisteminde orijin noktası belirlenmiş ve bu noktaya göre hedefin hangi bölgede kaldığı belirlenerek hareket açısının kalibrasyonu aşağıda verilen formülasyonla yapılmıştır (Rahman, 2018):

$$d = \sqrt{(|E_h - E_a|)^2 + (|B_h - B_a|)^2} \quad (7)$$

$$E_f = |E_h - E_a| * 111.1949 \quad (8)$$

$$B_f = |B_h - B_a| * 111.1949 * \cos(B_h + B_a)/2 \quad (9)$$

Bu denklemlerde E_h hedef enlem değerini, E_a anlık enlem değerini, B_h hedef boylam değerini, B_a ise anlık boylam değerini temsil etmektedir. Şekil 4.5’de mevcut konum olan Konya ve hedef konum olan Erzurum arasında hareket açısının hesaplanması için çizilmiş hayali üçgen harita üzerinde görülmektedir.

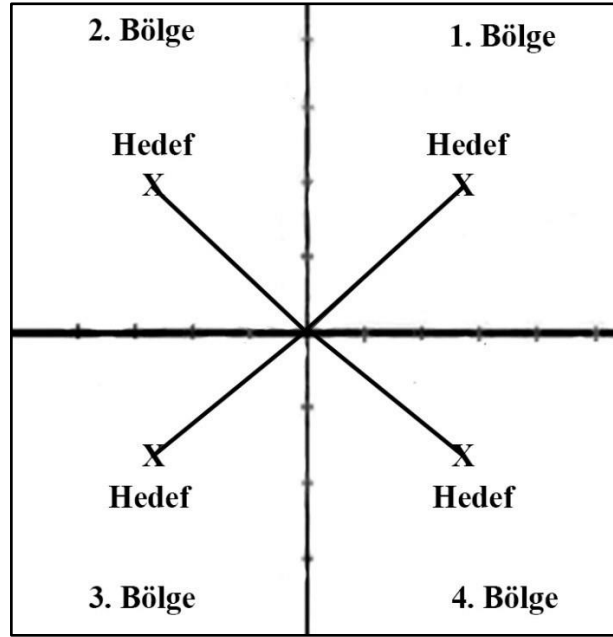


Şekil 4.5 Konya ile Erzurum arasında araç hareket yönünün belirlenmesi

Yön belirlenirken hedefin konumu ile aracın konumu arasında çizilen hayali çizgi yardımıyla hedefe doğru hareket yönü belirlenir. Hareket yönü belirlenirken kullanılan matematiksel denklem aşağıda verilmiştir:

$$\alpha = \tan^{-1} \frac{\text{Karşı Kenar}}{\text{Komşu Kenar}} \quad (10)$$

Bu durumda hareket açısı, aracın mevcut konumuna ve hedefin konuma göre değişmektedir. Araç koordinat sisteminde Şekil 4.13’de gösterildiği gibi orijine yerleştirildiğinde dört farklı bölge için hareket açısı hesaplanmıştır.



Şekil 4.6 Hedef konum ve araç konumuna göre değişen bölgeler

4.3.5 Hareket yönü kontrolü

Araç hareket açısı hesaplandığında belirlenen hareket açısına doğru hedefe ilerlenmesi sağlansa da olası bir engel ile karşılaşılması durumunda araç hedefe doğru olan yönden sapacaktır. Burada sapmanın büyüklüğü ve yönüne göre aracın yeniden hedefe yönlendirilmesi için oluşturulan kod bloğu aşağıda verilmiştir:

```
void istikametdenetleme(){
    if(mevcut_aci < (hedef_aci+10.0) && mevcut_aci > (hedef_aci-10.0))
    {
        servo.write(90);
        digitalWrite(uc1, HIGH);
        digitalWrite(uc2, LOW);
        analogWrite(hiz, 255);
    }
    else
    if(abs(mevcut_aci-hedef_aci) < 180 )
    {
        if(mevcut_aci > (hedef_aci+10.0) ) {
            servo.write(30);
```

```

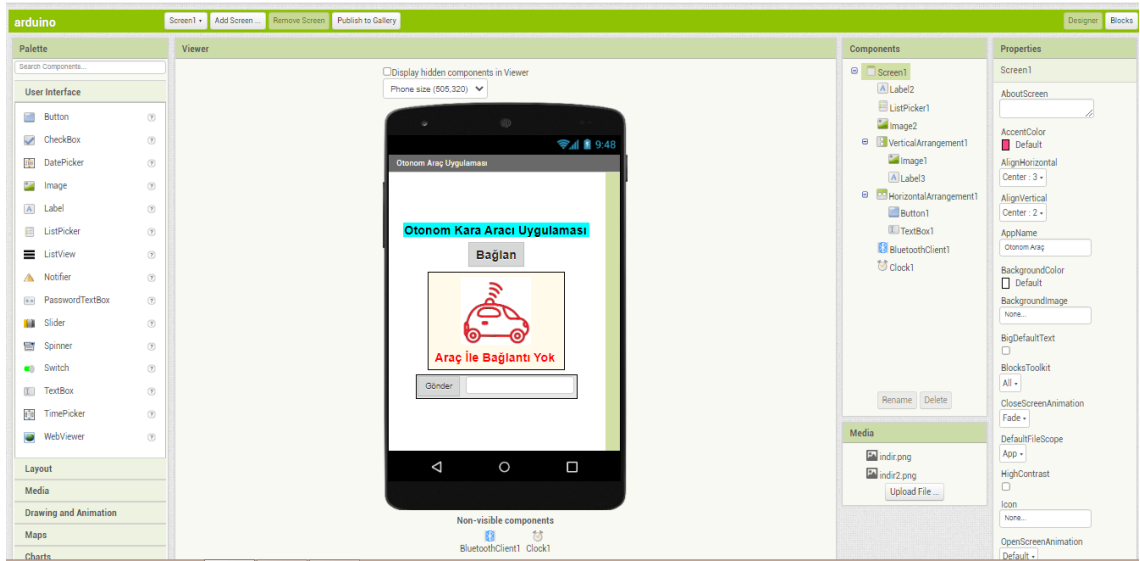
        delay(250);
    }
    else if( mevcut_aci < (hedef_aci-10.0)) {
        servo.write(150);
        delay(250);
    }
}
else
{
    if(mevcut_aci > (hedef_aci+10.0) ) {
        servo.write(150);
        delay(250);
    }
    else if( mevcut_aci < (hedef_aci-10.0)) {
        servo.write(30);
        delay(250);
    }
}
engel_denetle();
}

```

Oluşturulan kod satırlarında ilk şart ifadesi içerisinde mevcut açı ile hedef açı arasında sapma payı 10 derece bırakılır. Hedef açısı ile mevcut açı arasındaki fark belirlenip eğer bu aralık içerisinde ise direksiyon açısı düz olarak ayarlanıp araç ileri istikamette hareketine devam etmesi sağlanır. İki açı arasındaki farkın mutlak değeri 180'den küçük ise ikinci şart ifadesi çalıştırılır. Mevcut açı istikamet açısından en az 10 derece daha büyük ise direksiyon açısı sağa doğru çevrilir. Mevcut açı istikamet açısından en az 10 derece daha küçük ise direksiyon açısı sola doğru çevrilir. Diğer bir şart ifadesinde ise iki açı arasındaki farkın mutlak değeri 180'den fazla ise hedef açısına yaklaşmak için hesaplanan fark 10 dereceden fazla ise direksiyon açısı sola, 10 dereceden az ise direksiyon açısı sağa doğru yönlendirilir. Programın sonunda engel denetleme bloğu çalıştırılarak hesaplanan hareket açısına doğru ilerlerken yolda herhangi bir engelin olup olmadığı kontrol edilir.

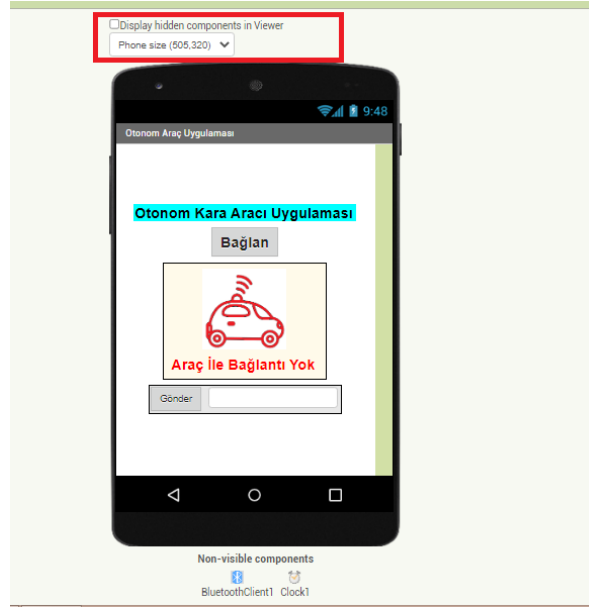
4.3.6 Araç ile haberleşme için geliştirilen Android uygulaması

Bu çalışmada araç ile uzaktan haberleşme amacıyla kullanılmak üzere akıllı telefon üzerinde çalışacak Android uygulaması geliştirilmiştir. Geliştirilen uygulamanın amacı, araca hedef koordinat bilgisini kablosuz olarak göndermektir. Bunun için araç üzerine HC-06 Bluetooth modül yerleştirilmiş ve MIT App Inventor ile Android işletim sistemine sahip cep telefonu için arayüz geliştirilmiştir. MIT App Inventor, Google tarafından ortaya koyulan ve sonrasında Massachusetts Institute of Technology tarafından geliştirilen, özgür bir uygulama geliştirme aracıdır. Android işletim sistemi için blok kodlama yöntemiyle uygulama geliştirmesine olanak sağlar. Yapboz yapısı ve sürükle-bırak mekanizması sayesinde MIT App Inventor ile kolaylıkla uygulama geliştirilmesi mümkündür (Vikipedi, 2023). Tez çalışmasında araca hedef konum bilgisi göndermek için kullanılmak üzere geliştirilen Android uygulama Şekil 4.7’de gösterilmiştir.



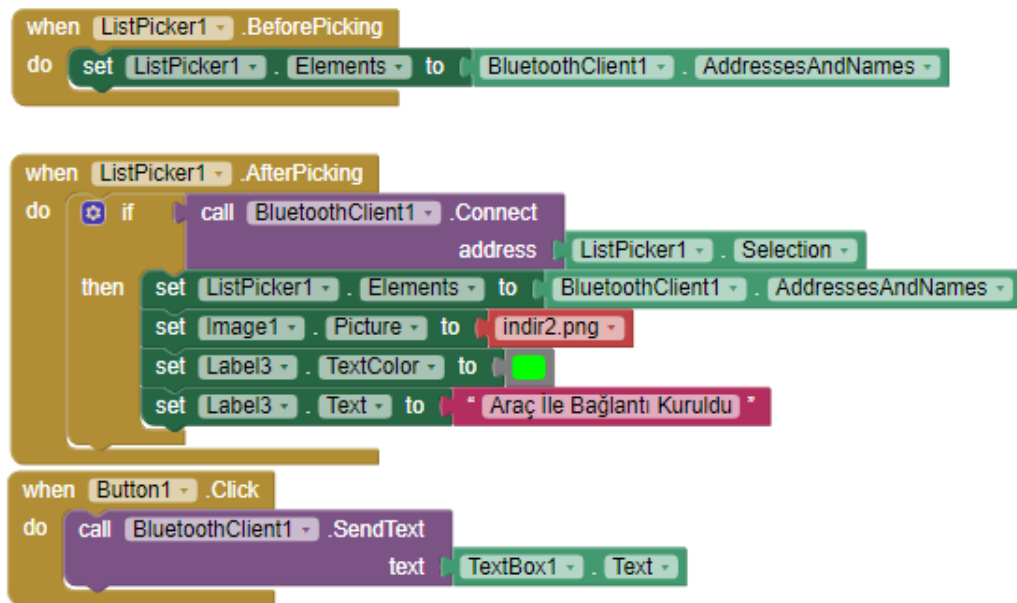
Şekil 4.7 App Inventer ile uygulama arayüzü geliştirme

Uygulamanın tasarlanması için proje dosyası hazırlanıp uygulamanın hangi cihaz üzerinden kullanılacağı seçilir (telefon, tablet, PC vb.) ve ardından telefonun ekran boyutu görüldüğü belirlenir. Şekil 4.8’de gösterildiği gibi “Label” “Textbox” “Button” araçları form üzerine yerleştirilerek uygulama arayüzü oluşturulmuştur. Ardından “Connectivity” menüsünden “BluetoothClient” aracı projeye dahil edilmiştir. Arayüz tasarımının ardından “Blocs” sekmesinden uygulamaya dahil edilen araçların nasıl çalışacağı belirlenmiştir.



Şekil 4.8 Mobil uygulama için cihaz ve çözünürlük seçimi ekranı

Mobil uygulamanın amacına uygun olarak çalışması için geliştirilen yapboz fonksiyon blokları Şekil 4.9’da gösterilmiştir. İlk blokta uygulama açıldığında önceden eşleştirme yapılmış Bluetooth cihazlarının listelenmesi sağlanmıştır. İkinci blokta ise açılan listede seçilen bluetooth modülü ile bağlantı sağlanıp ardından kırmızı olan araç ikonu yeşile döneceği sonra yeşil renkte “Araç ile Bağlantı Kuruldu” yazısı ekranda görüntülenmesi sağlanmaktadır. Üçüncü blokta ise “Gönder” butonuna her basıldığında “TextBox1” içinde yazılı ifadenin bağlı cihaza gönderilmesi sağlanmaktadır.



Şekil 4.9 Mobil uygulama için yapboz fonksiyon blokları

Aracın akıllı telefon uygulamasından gelen konum bilgisini işleyip hedefe yönlendirilmesi için hazırlanan kod bloğu aşağıda verilmiştir:

```
void bluetooth(){
    if (Serial.available() > 0) {
        veri = Serial.read();
        if (veri == 50){
            hedef_enlem=38.424332; hedef_boylam=27.147501;} //İzmir
        else if (veri == 49){
            hedefenlem=39.934287, hedefboylam=32.890664;} //Ankara
        else if (veri == 51) {
            hedefenlem=36.906211, hedefboylam=30.713970;} // Antalya
        else if (veri == 52) {
            hedefenlem=39.902807, hedefboylam=41.268494;} //Erzurum
        else if (veri == 48)
            analogWrite(hiz, 0);
        }
    }
}
```

Haberleşme için geliştirilen kod bloğunda akıllı telefon için geliştirilen uygulama ile kullanıcı tarafından girilen koordinat bilgisi “veri” değişkeninin içine aktarılır. Şart komutlarıyla aracın hedefe göre durumuna bağlı olarak “hedef_enlem” ve “hedef_boylam” değişkenleri içerisine hedefin coğrafi konum bilgileri aktarılmış olur.

4.3.7 Bulanık mantık ile dönüş açısı belirleme

Araç kontrolünde, aracın hareket edip etmediği lojik olarak 0 veya 1 şeklinde ifade edilebilir. Fakat gerek duyulduğunda aracın yön ve hız bilgisi de sorgulanabilir. Kesin olmayan, yoruma açık verilerin kontrol uygulamalarında daha esnek şekilde işlenebilmesi için Lotfi A. Zadeh tarafından 1965 yılında bulanık mantığın temelleri atılmıştır (Zadeh, 1965). Bulanık mantık temelleri atıldıktan on yıl sonra ticari uygulamalarda kullanılmaya başlanmıştır (Altaş, 1999). Günümüzde belirli bir matematiksel modele sahip olmayan sistemlerin kontrolünde yaygın olarak bulanık mantık kullanılmaktadır (Ünsal ve ark., t.y.).

Bulanık mantık ilk kullanım alanlarından olan buhar makinesinden birçok fabrikada kullanılan SCADA sistemlerine kadar pek çok alanda yaygın olarak kullanılmaktadır. Bulanık mantıkta istenilen kesin veri girişinin ardından daha önceden belirlenen bir kural tablosu içerisinde sözel bazı şartlar konularak bulanıklaştırıcı ile belirli bir çıkarım sağlanır ve bulanık mantık işleyicisi sayesinde veriler işlenerek kesin olmayan çıkış bilgisine dönüştürülür. Bu çıkarım yöntemlerinden en yaygın olan yöntemler Mamdani ve Sugeno yöntemleridir.

Bu tez çalışmasında Mamdani çıkarım yöntemi kullanılmaktadır. Bu yöntemde bulanıklaştırıcı çıkışında elde edilen kesin olmayan veriler kural tabanında işlenerek durulaştırmaya tabi tutulur. Durulaştırma işlemi için çeşitli yöntemler bulunmaktadır. Bunlardan bazıları; ağırlık merkezi, ağırlıklı ortalama, alan merkezi, en büyüklerin en küçüğü ve en büyüğü, en büyüklerin ortalaması durulaştırma yöntemleridir. Bu tez çalışmasında alan merkezi yöntemi kullanılarak kesin çıkış verisi elde edilmiştir.

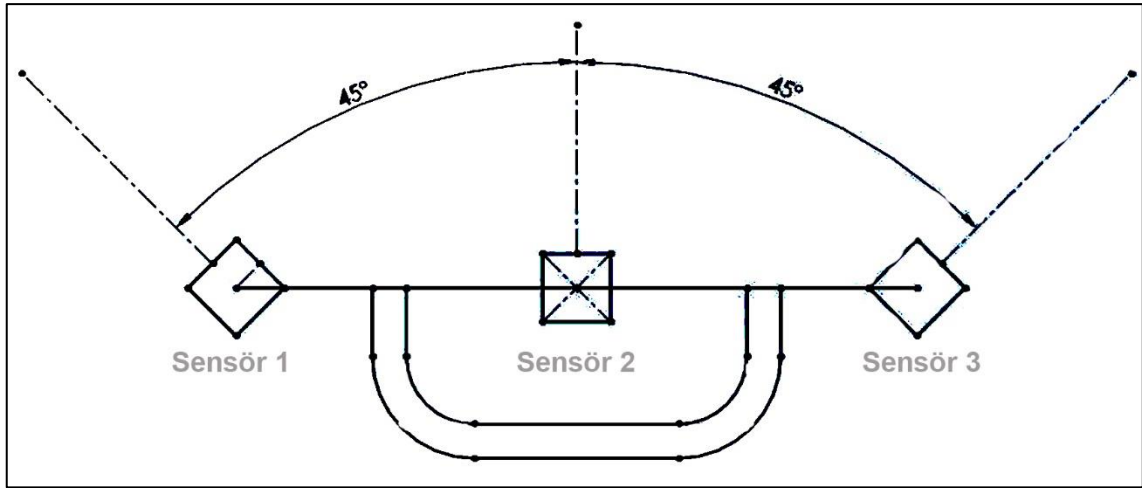
Bulanık mantık kontrolcüsünün avantajları kontrol edilen sistemin matematiksel bir modelin olmasına ihtiyaç duyulmaması, doğrusal olmayan matematiksel modellerin çalışmasına izin verilmesi, karmaşık kontrol sistemlerine basit çözümler getirmesi olarak sayılabilir. Bunun yanında kural tablosunun oluşturulabilmesi için uzman bilgisine ihtiyaç duyulması gibi dezavantajlar da mevcuttur (Kıyak ve Kahvecioğlu, 2003).

4.3.7.1 Engelden kaçma algoritması

Araç saha üzerinde GPS verisi kullanarak kendisine verilen hedefe doğru ilerlerken ultrasonik mesafe sensörleri ile önüne çıkabilecek engelleri tarar. Bu sensörlerden aldığı veriler engel varlığına işaret ederse, önüne çıkan engelleri aşması gerekir. Şekil 4.11'de gösterildiği gibi araç üzerinde yerleştirilmiş ultrasonik sensörlerden elde edilen mesafe ölçüm değerleri bulanık kontrolör için giriş olarak kullanılmıştır. Bulanık kontrolör çıkışında ise direksiyonu kontrol eden servo motor için açı değeri elde edilmektedir.

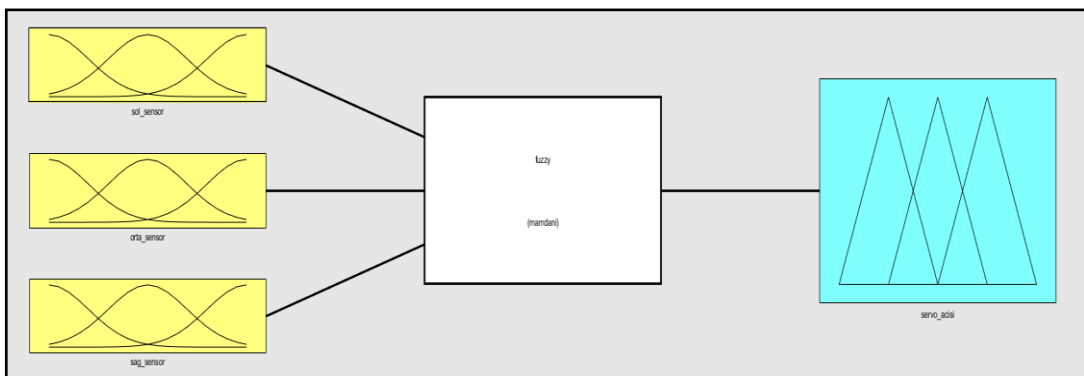
4.3.7.2 Bulanıklaştırma yöntemi

Bulanık kontrolör girişinde ultrasonik sensör mesafe ölçüm değerlerinin sözel ifadelerle dönüştürülerek bulanıklaştırılması gerekmektedir. Bu aşamada insan zekâsı ve uzman bilgisi kullanılarak bulanıklaştırma yapılır.

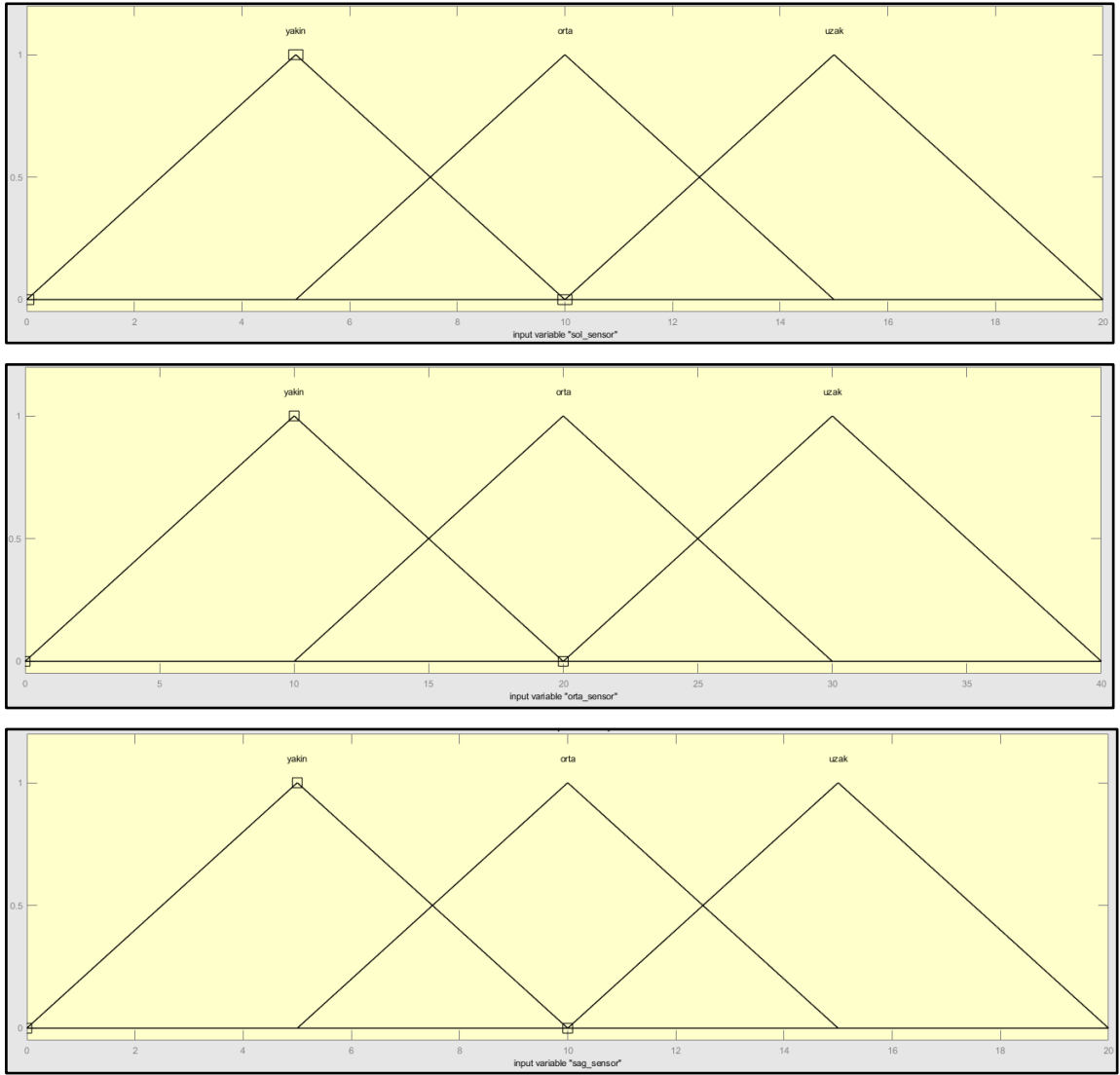


Şekil 4.9 Otonom aracın ön kısmında yer alan ultrasonik sensörler

Şekil 4.12’de araç kontrol uygulamasında kullanılan ve MATLAB bulanık mantık aracı kullanılarak oluşturulan kontrolör blok diyagramı gösterilmiştir. Mesafe sensörleri araç hareket yönünde engel tespit ettiğinde bulanık kontrolöre engellerin uzaklığı ile ilgili veri gönderimi yapar. Yapılan çalışmada aracın engellerden kaçarken ideale yakın bir rota izlemesi amaçlanmış olup, hedefe doğru hareket etmeye engel olmayacak kadar uzaktaki engeller için bulanık kontrolöre veri girişi yapılmamıştır. Aracın sol, orta ve sağ tarafında bulunan ultrasonik mesafe sensörlerinden alınan ölçüm değerleri giriş üyelik fonksiyonlarına bağlı olarak bulanıklaştırılır. Şekil 4.13’de mesafe sensörleri için kullanılan bulanık mantık giriş üyelik fonksiyonları gösterilmiştir.

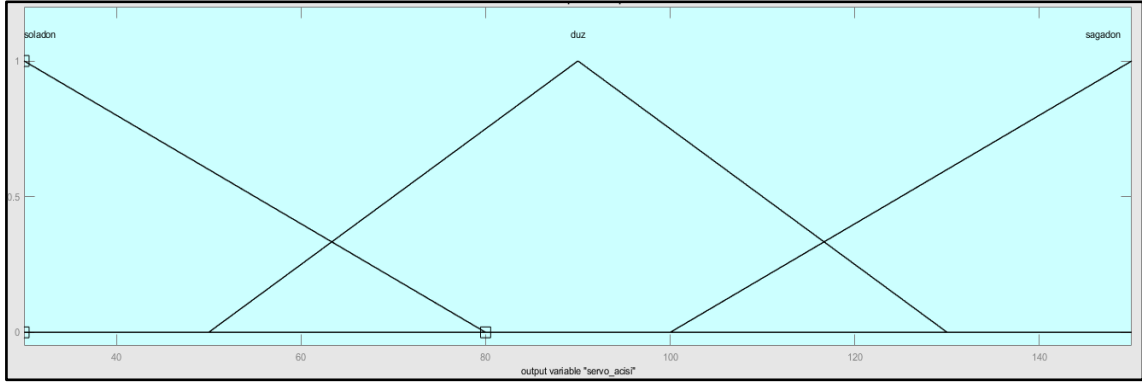


Şekil 4.10 Çalışmada kullanılan bulanık mantık algoritması



Şekil 4.11 Bulanık mantık giriş üyelik fonksiyonları

Aracın sağındaki ve solunda engellerin aşılması, önündeki engellere göre daha kolay olacağından sağ ve sol sensörler için mesafe ölçüm aralığı 0-20 cm, orta sensör için ise giriş 0-40 cm olarak seçilmiştir. Giriş değerlerinin bulanıklaştırılması için her biri üçgen fonksiyonu ile tanımlanan üç farklı üyelik fonksiyonu (yakın, orta, uzak) tanımlanmıştır. Bulanıklaştırılan mesafe ölçüm değerleri, Mamdani yöntemi kullanılarak oluşturulan kural tablosuna bağlı işlenmektedir. Bulanık kontrol uygulamasında durulaştırma için kullanılan üyelik fonksiyonları Şekil 4.14’de gösterilmiştir. Aracın engelden kaçışı için yön tayininde “soladon, duz, sagadon” şeklinde üç farklı çıkış tanımlanmıştır. Dönüş açısı 30 ile 150 derece arasında elde edilerek aracın engelden kaçış yönü belirlenmektedir. Kullanılan servo motor 0 ile 180 derece arasında çalışabiliyor olsa da aracın mekanik yapısından kaynaklanan sınırlama nedeniyle aralık daraltılmıştır.



Şekil 4.12 Bulanık mantık çıkış üyelik fonksiyonları

4.3.7.3 Kural tabanı

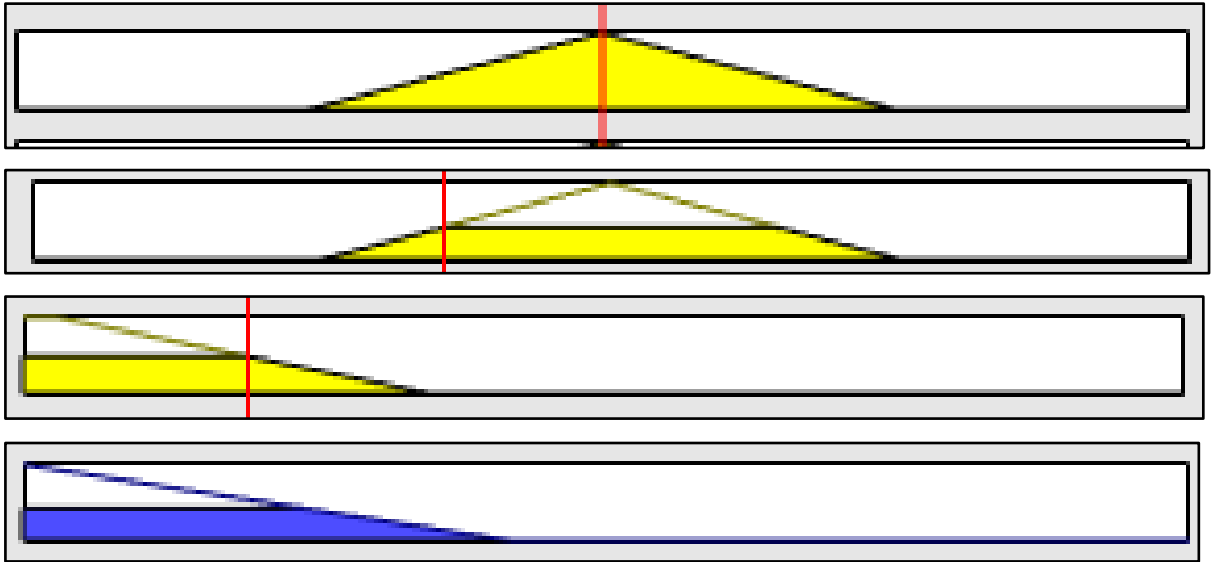
Kural tabanı insan tecrübesine dayanarak oluşturulan bulanık mantığın karar mekanizmasını temsil eder. Bulanık kontrol uygulamasında karşılaşılabilecek tüm durumlara göre ihtiyaç duyulacak kurallar oluşturulur. Giriş üyelik fonksiyonlarının sayısı, kural tabanındaki kural sayısını belirlemektedir. Sensör ölçüm değerlerine bağlı olarak, kural tablosunda birden fazla kurula uyan senaryolar ortaya çıkabilmektedir. Bulanık kontrolün doğası gereği, tüm durumlar için uzman tecrübesi veya insan zekasına uygun olarak ideal kurallar tanımlanarak çıkış elde edilmektedir. Çizelge 4.2'de gerçekleştirilen otonom araç kontrolü uygulamasında kullanılan kural tabanı verilmiştir.

Çizelge 4.2 Bulanık mantık kural tabanı

SENSOR 1	SENSOR 2	SENSOR 3	DİREKSİYON	SENSOR 1	SENSOR 2	SENSOR 3	DİREKSİYON
YAKIN	YAKIN	YAKIN	SAGADON	ORTA	ORTA	UZAK	SAGADON
YAKIN	YAKIN	ORTA	SAGADON	ORTA	UZAK	YAKIN	SOLADON
YAKIN	YAKIN	UZAK	SAGADON	ORTA	UZAK	ORTA	DUZ
YAKIN	ORTA	YAKIN	DUZ	ORTA	UZAK	UZAK	DUZ
YAKIN	ORTA	ORTA	SAGADON	UZAK	YAKIN	YAKIN	SOLADON
YAKIN	ORTA	UZAK	SAGADON	UZAK	YAKIN	ORTA	SOLADON
YAKIN	UZAK	YAKIN	DUZ	UZAK	YAKIN	UZAK	SOLADON
YAKIN	UZAK	ORTA	DUZ	UZAK	ORTA	YAKIN	SOLADON
YAKIN	UZAK	UZAK	SAGADON	UZAK	ORTA	ORTA	SOLADON
ORTA	YAKIN	YAKIN	SOLADON	UZAK	ORTA	UZAK	SOLADON
ORTA	YAKIN	ORTA	SOLADON	UZAK	UZAK	YAKIN	SOLADON
ORTA	YAKIN	UZAK	SAGADON	UZAK	UZAK	ORTA	DUZ
ORTA	ORTA	YAKIN	SOLADON	UZAK	UZAK	UZAK	DUZ
ORTA	ORTA	ORTA	DUZ				

Kural tabanı kullanımı şu şekilde açıklanabilir: Sensörlerden alınan mesafe ölçümlerinin bulanıklaştırılması sağlanır ve bunlara karşılık gelen girişler için kurallar

çalıştırılır. Her kural için girişlerin üyelik fonksiyonları belirlenir, engele olan mesafeye karşılık gelen üyelik değerleri elde edilir. Elde edilen üyelik değerleri arasında en küçük olan üyelik değeri çıkış olarak belirlenen üyelik fonksiyonuna iletilir. Mamdani tarafından geliştirilen maksimum-minimum (max-min) çıkarım yöntemine bağlı olarak çıkışta ilgili üyelik fonksiyonuna karşılık gelen üyelik değeri elde edilir. Şekilde 4.15’de sağ, orta ve sol sensörlerden alınan mesafe bilgisi için dönüş açısının bulanık mantık ile belirlenmesi grafiksel olarak gösterilmiştir. Sarı renk ile gösterilen kısımlar giriş üyelik fonksiyonlarını temsil etmektedir. Görüldüğü üzere en küçük üyelik değeri, çıkış üyelik değeri olarak alınmaktadır. Mavi renk ile gösterilen çıkış üyelik fonksiyonunda, üyelik değerine bağlı olarak altta kalan alan elde edilmiştir. Mesafe sensörlerinden alınan ölçüm değerlerine bağlı olarak kural tabanında uygun olan her kural için bu şekilde alanlar elde edilir ve çıkış değerini belirlemek için çıkarım yapılır.



Şekil 4.13 Üyelik fonksiyonlarının görsel karşılıkları

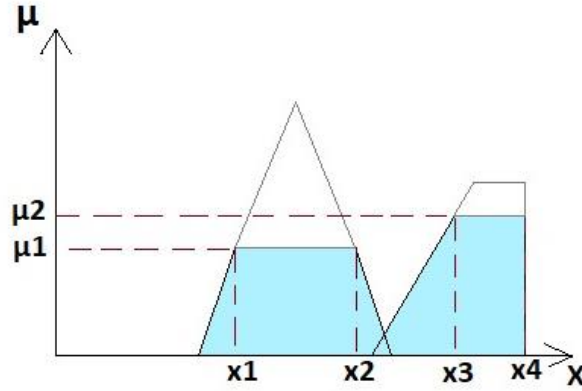
4.3.7.4 Durulaştırma

Bulanık kontrolör çıkışında elde edilen bulanık değerlerin durulaştırılıp kesin değerlere dönüştürülmesi gerekmektedir. Durulama işlemi ile gerçekleştirilen otonom araç kontrol uygulamasında mikrokontrolör tarafından kullanılacak hareket açısı değeri elde edilir. Yapılan uygulamada durulaştırma işlemi için alan merkezi yöntemi kullanılmıştır. Bu yöntemde yamuk ya da üçgen şeklinde çıkış üyelik fonksiyonu için üyelik değerine karşılık gelen tüm çıkışların ortalaması alınır. Elde edilen ortalama değer ile üyelik değeri

çarpılır ve her çıkış için elde edilen değerler toplanır. Toplam sonucunda elde edilen değerlerin kullanılan üyelik değerlerinin toplamına bölünmesi ile durulaştırma gerçekleştirilir. Durulaştırma için kullanılan matematiksel formül aşağıda verilmiştir:

$$\frac{\sum_{i=1}^n x_{ort_i} * \mu_i}{\sum_{i=1}^n \mu_i} \quad (11)$$

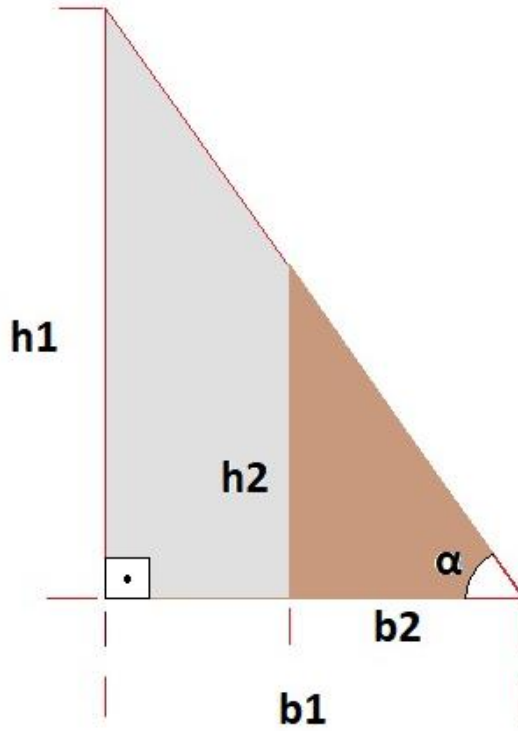
Burada $x_{ort} = \frac{x_1 - x_2}{2}$ değeri, üyelik değerine karşılık gelen x değerlerinin ortalamasını, μ ise üyelik değerini temsil etmektedir. Alan merkezi yöntemi ile durulaştırma işlemi Şekil 4.16'da gösterilmiştir. Durulaştırma sonucunda elde edilen açı değeri direksiyon kontrolü için servo motora iletilerek engelden kaçış sağlanır.



Şekil 4.14 Alan merkezi durulaştırma yöntemi uygulaması

4.3.7.5 Bulanık mantık için kullanılan geometrik yöntemler ve kodlanması

Bulanık mantık üyelik fonksiyonlarında üyelik değeri ve çıkış değerlerini elde etmek için geometrik kurallardan yararlanılmaktadır. Bu çalışmada kullanılan üyelik fonksiyonları üçgen ve yamuk şeklinde olduğu için dik üçgende benzerlik yöntemi kullanılarak üyelik değeri hesaplamaları yapılmıştır.



Şekil 4.15 Üçgende benzerlik yöntemi

Üçgende benzerlik kuralı aşağıda verilmiştir:

$$\frac{h_1}{b_1} = \frac{h_2}{b_2} \quad (12)$$

Burada h değerleri yüksekliği temsil ederken b değerleri taban uzunluğunu temsil etmektedir. Arduino için kod geliştirme yapılırken üçgende benzerlik yöntemi kullanılmıştır. Bu şekilde sözel ifadeler kod satırlarına aktarılmıştır. Buna ilişkin bir kod örneği aşağıda verilmiştir:

```

if((0.0<sol_sensor&&sol_sensor<10.0)&& (0.0<orta_sensor&&orta_sensor<20.0)&&
(0.0<sag_sensor&&sag_sensor<10.0))
{
    if(0.0<sol_sensor&&sol_sensor<10.0){
        if(sol_sensor<5.0){
            solu=(sol_sensor-0.0)/(5.0-0.0);
        }
        else if(sol_sensor>5.0){

```

```

        solu=(10.0-sol_sensor)/(10.0-5.0);
        }
    else{solu=0.99;
    }
}

```

Kod satırlarında görüleceği üzere her mesafe sensörü için giriş üyelik fonksiyonunu belirlemek üzere kodlama yapılmıştır. Mesafe sensöründen gelen değere bağlı olarak ilgili kod satırları çalışmakta ve üyelik değeri elde edilmektedir. Yamuk ya da üçgen şeklinde olan üyelik fonksiyonlarında, üyelik değerlerini elde etmek için dik üçgenler tanımlanmış ve bu dik üçgenlerin yüksekliği mevcut olan ya da bulunması gereken üyelik değeri olurken taban uzunluğu olarak yatay eksendeki birim değerler olarak ele alınmıştır. Böylece engele olan mesafeye karşılık gelen üyelik değerlerini hesaplayabilmek için benzerlik eşitliği kod satırlarına dönüştürülmüştür. Elde edilen üyelik değerleri sırasıyla; sol sensör üyelik değeri, sağ sensör üyelik değeri ve orta sensör üyelik değeri olarak “solu, sagu, ortau” şeklinde tanımlanmıştır.

Her sensör için elde edilen üyelik değerlerinden en küçük olanın, kurala bağlı olarak ilgili üyelik fonksiyonuna iletilmesi gerekmektedir. Buna sağlayabilmek için küçük olanı seçen bir karşılaştırma denetimi hazırlanmıştır. Bu denetim sonucunda elde edilen değer, ilgili çıkış üyelik fonksiyonuna iletilerek sistemin bulanık çıkarım yapması sağlanmıştır. Durulaştırma işlemi için oluşturulan kodlar aşağıda verilmiştir:

```

float durulama(float alanmerkezi,float toplamuyelik){
    aci=alanmerkezi/toplamuyelik;
    float alanmerkeziw=0.0;
    float toplamu=0.0;
}
float soladon(float uno){
    deger=80.0-(50.0*uno);
    ortdeger=(deger+30.0)/2.0;
    carpim= ortdeger*uno;
    alanmerkeziw=carpim+ alanmerkeziw;
    toplamu=uno+toplamu;
}

```

```

float sagadon(float uno){
    deger=(50.0*uno)+100.0;
    ortdeger=(deger+150.0)/2.0;
    carpim= ortdeger*uno;
    alanmerkeziw=carpim+ alanmerkeziw;
    toplamu=uno+toplamu;
}
float duz(float uno){
    deger=(40.0*uno)+50.0;
    float deger2=130-(40*uno);
    ortdeger=(deger+deger2)/2.0;
    carpim= ortdeger*uno;
    alanmerkeziw=carpim+ alanmerkeziw;
    toplamu=uno+toplamu;
}

```

Alan merkezi durulaştırma yöntemi göre, her kural için, üyelik değerine denk gelen çıkış değerlerinin ortalaması alınır ve üyelik değeriyle çarpılır. Devamında ilgili tüm kurallar için bulunan bu değerler toplanır ve toplam üyelik değerlerine bölünür. Oluşturulan fonksiyonlarda ortalama değer hesaplanabilmesi için, üyelik değeri için ilk ve son değerlere ihtiyaç vardır. Üçgende benzerlik kullanılarak bu değerler hesaplanır. Daha sonra ilk ve son değer kullanılarak ortalama değer elde edilir. Hesaplanan ortalama değer ile ilgili üyelik değeri çarpılır ve bu çarpımların toplandığı değişkene atanır. Durulaştırma işleminde o anki girişlere uygun olan tüm kuralların çıkış verileri toplanarak ve ilgili üyelik değerlerinin toplamına bölünür. Sonuçta elde edilen değer “aci” değişkenine atanır. Böylece kesin bir değer elde edilerek durulaştırma tamamlanmış olur.

5 ARAŞTIRMA SONUÇLARI VE TARTIŞMA

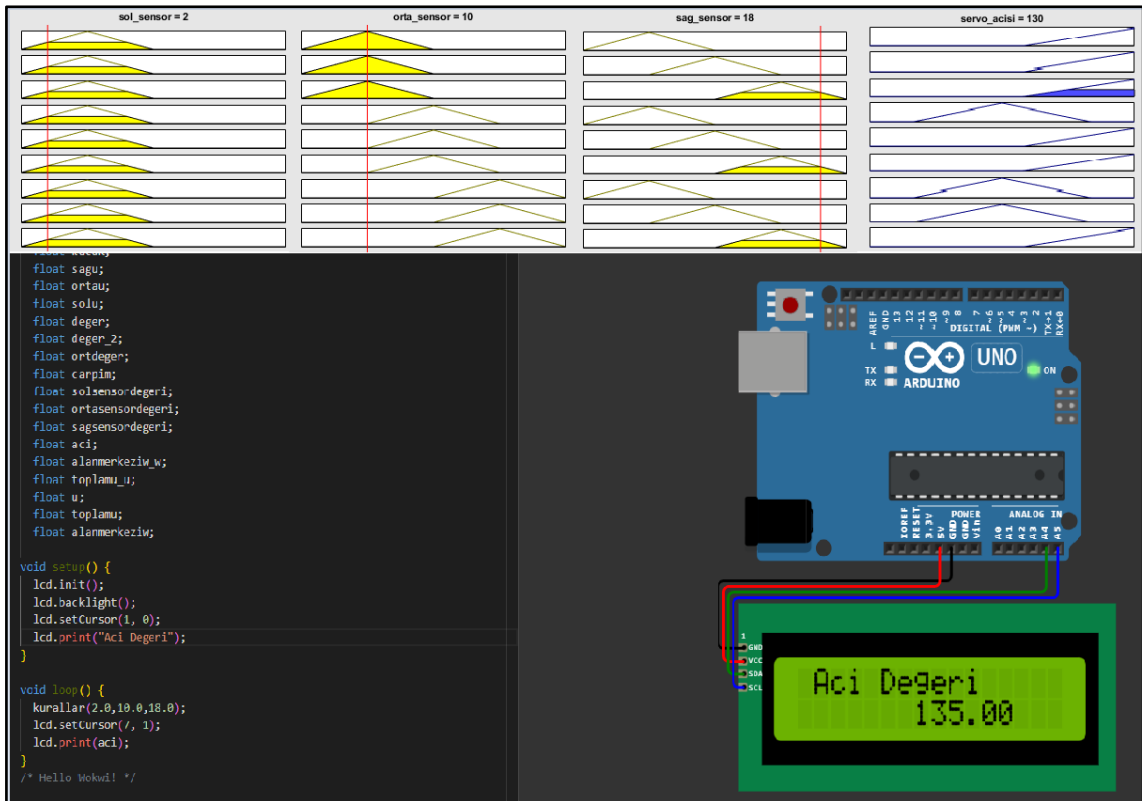
5.1 Bulanık Kontrol Uygulaması Test Sonuçları

Otonom sürüşü mümkün kılan bulanık kontrolör için MATLAB ile elde edilen çıkış değerleri, hazırlanan Arduino kodları ile elde değerlerle karşılaştırılmış ve böylece sistemin doğru çalıştığı doğrulanmıştır. Yapılan testlerde beş farklı giriş durumu oluşturulmuş ve her biri için elde edilen çıkış değerleri karşılaştırılmıştır. Test amacıyla giriş olarak alınan mesafe ölçüm değerleri Çizelge 5.1’de gösterilmiştir. Çizelgedeki gösterilen değerler cm cinsinden, engellere olan mesafeyi göstermektedir.

Çizelge 5.1 Bulanık mantık test değerleri

Test No	Sol Sensör Değeri (cm)	Orta Sensör Değeri (cm)	Sağ Sensör Değeri (cm)
1	2.0	10.0	18.0
2	17.0	35.0	3.0
3	10.0	8.0	16.0
4	8.0	28.0	6.0
5	11.0	13.0	17.0

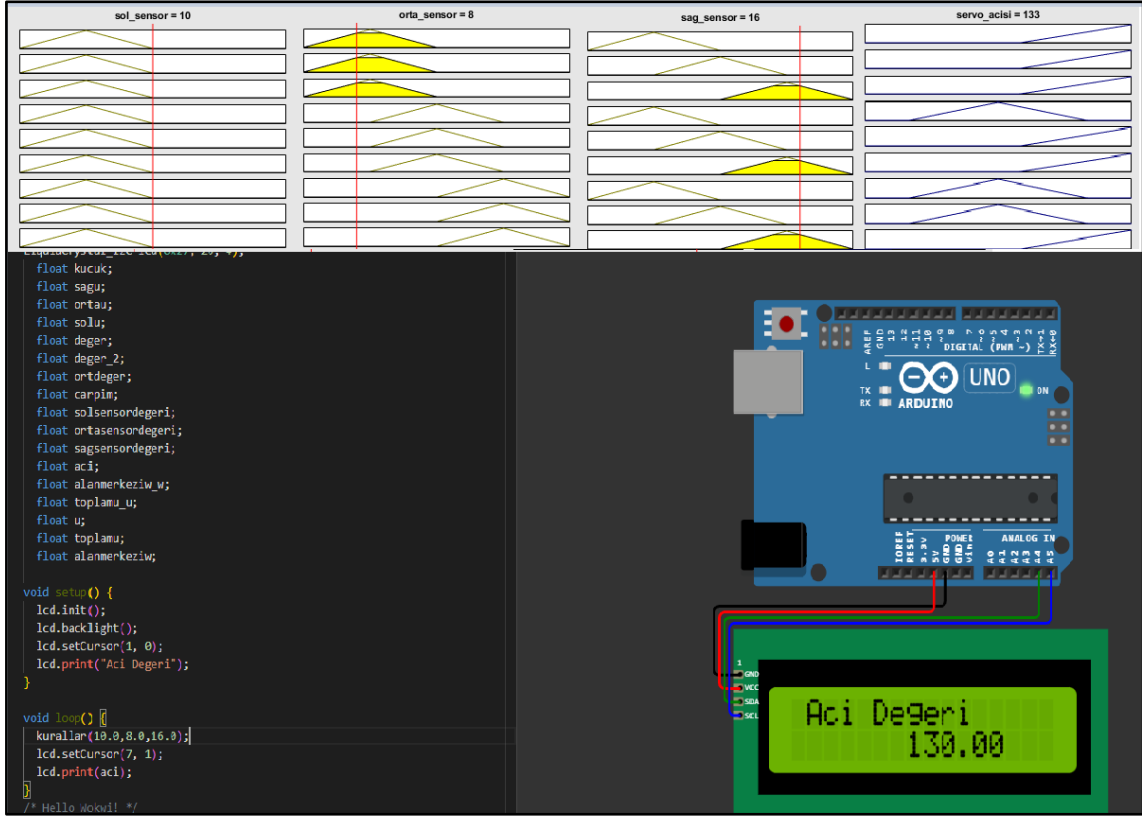
Tanımlanan beş farklı durum için MATLAB programı ve Arduino emülatör ile yapılan karşılaştırmalı testlere ilişkin elde edilen sonuçlar sırasıyla Şekil 5.1, Şekil 5.2, Şekil 5.3, Şekil 5.4 ve Şekil 5.5’de gösterilmiştir. Arduino için hazırlanan kod satırlarında matematiksel hesaplamayı kolaylaştıracak ifadeler kullanılması sebebiyle elde edilen sonuçlar arasında farklılıklar olduğu gözlenmiştir. Şekil 5.1’de aracın sağ tarafında daha fazla boşluk bulmasından dolayı sağa doğru hareketlendiği görülmektedir. Matlab ile 130.0° çıkış elde edilirken, Arduino ile 135.0° çıkış elde edilmiştir. Şekil 5.2’de aracın sola doğru yöneldiği görülmektedir. Böylece çok yakında olan sağ taraftaki engelden kaçış sağlanmaktadır. MATLAB ile 49.1° çıkış elde edilirken, Arduino ile 42.50° çıkış elde edilmiştir. Şekil 5.3’de aracın yine sağa doğru yöneldiği görülmektedir. Burada aracın önünde daha yakın bir engel olduğu düşünüldüğünde bu engelin aşılabilmesi için aracın sağa ya da sola manevra yapması mümkündür. Bulanık kontrolör çıkışında sağa manevra yapılması gerektiği ortaya çıkmıştır. MATLAB ile çıkış olarak 133.0° açı hesaplanırken, Arduino ile 130.0° çıkış elde edilmiştir.



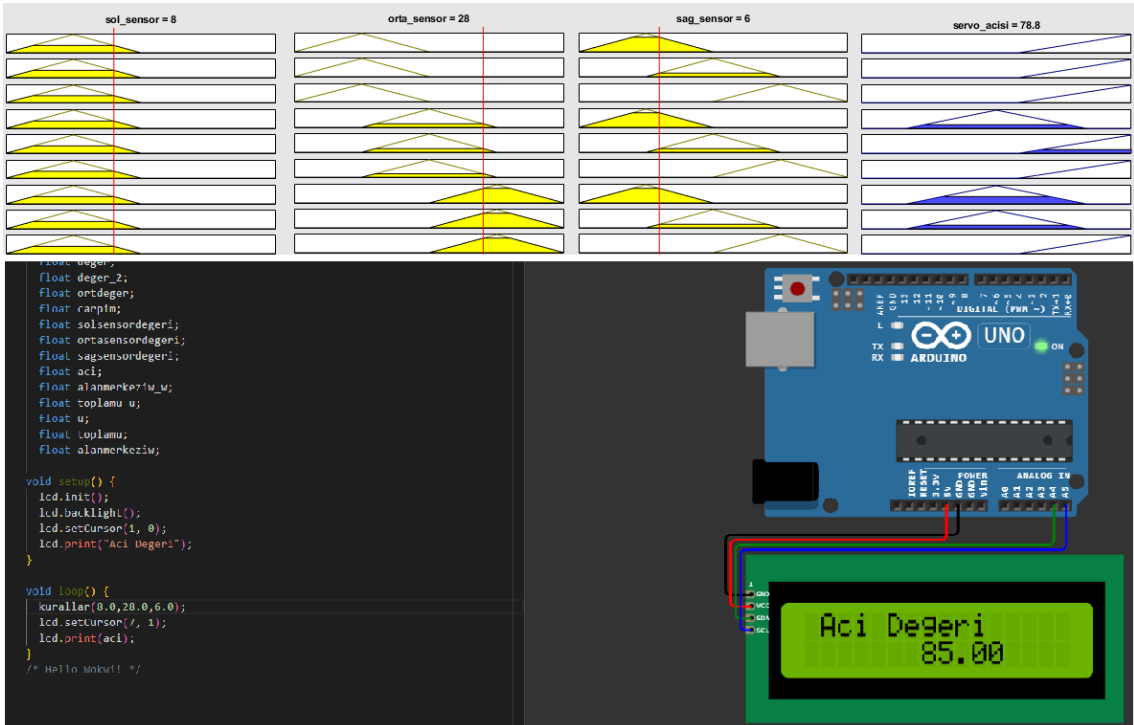
Şekil 5.1 Benzetim 1 sonuçları



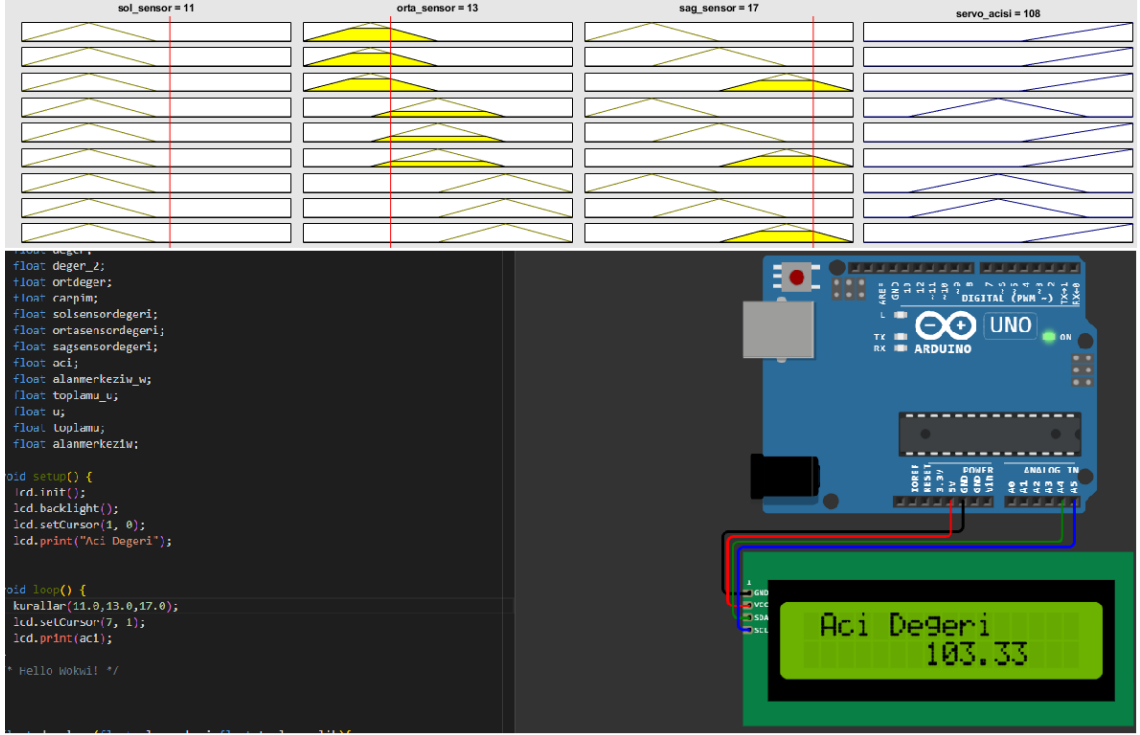
Şekil 5.2 Benzetim 2 sonuçları



Şekil 5.3 Benzetim 3 sonuçları



Şekil 5.4 Benzetim 4 sonuçları



Şekil 5.5 Benzetim 5 sonuçları

Şekil 5.4’de aracın düz bir hareket izlediği görülmektedir. Bununla birlikte sağdaki engelin, soldaki engele göre daha yakın olması sebebiyle sola doğru küçük bir dönüş açısı ile hareketi söz konusudur. MATLAB ile çıkış olarak 78.8° hesaplanırken, Arduino ile 85.0° çıkış hesaplanmıştır.

Son olarak Şekil 5.5’de tüm sensörler için engellere olan mesafelerin birbirlerine çok yakın olduğu görülmektedir. Bu durumda aracın, en uzak olan engele doğru hareket ettiği görülmektedir. Pratikte aracın en uzak engele doğru yönelip, diğer engellerden kurtulabilmesi beklenir. Eğer aracın engeller arasında sıkışacağı bir durum ortaya çıkarsa, geriye hareket ederek kurtulması sağlanmıştır. MATLAB ile elde edilen 108° çıkış değerine karşılık, Arduino ile 103.33° çıkış elde edilmiştir.

Elde edilen sonuçların birlikte değerlendirilmesi için Çizelge 5.2 hazırlanmıştır. Bu çerçevede Arduino ortamında hazırlanan bulanık kontrol kodlarının aracın engelden kaçış yönelimleri doğru olarak gerçekleştirdiği anlaşılmaktadır. MATLAB kullanılarak elde edilen değerler ile Arduino çıkış değerleri arasındaki farklılıkların %10’un altında kaldığı gözlenmektedir. MATLAB uygulamasında durulaştırma yöntemi olarak ağırlık merkezi yöntemi tercih edilmişken, Arduino ortamında, kodlama kolaylığından dolayı alan merkezi yöntemi tercih edilmiştir. Araç dönüş açısı için yapılan testlerde ortaya çıkan sapmaya sebep olarak MATLAB uygulamasında kullanılan durulaştırma yöntemi

ile Arduino ortamında kullanılan durulaştırma yönteminin farklılığı gösterilebilir. Yüksek hassasiyet gerektirmeyen uygulamalar için elde edilen dönüş açısı değerlerinin kabul edilebilir olduğu söylenebilir.

Çizelge 5.2 Arduino benzetim sonuçları

Deney No	Sol Sensör Değeri (cm)	Orta Sensör Değeri (cm)	Sağ Sensör Değeri (cm)	Matlab Çıkış Değeri (derece)	Arduino Kodu Çıkış Değeri (derece)	Çıkıştaki Sapma (%)	Yönelim
1	2.0	10.0	18.0	130.0	135.0	4.16	Sağa
2	17.0	35.0	3.0	49.1	42.5	5.5	Sola
3	10.0	8.0	16.0	133.0	130.0	2.5	Sağa
4	8.0	28.0	6.0	78.8	85.0	5.16	Düz
5	11.0	13.0	17.0	108.0	103.33	3.89	Sağa

5.2 Saha Testleri

Prototip tasarımı yapılan aracın sahada davranışlarını görebilmek adına, otonom kara aracı dört farklı senaryoda test edilmiştir.

Birinci senaryoda araç üzerinde alınan GPS konum bilgisinin doğruluğu incelenmiştir. Aracın, verilen hedef enlem ve boylama gitmesi istenmiştir. Bu saha testinde araç önünde herhangi bir engel yoktur.

İkinci senaryoda araç hedefe doğru ilerlerken karşılaşılabileceği farklı engeller karşısında nasıl bir davranış sergileyeceği izlenmiştir. Burada engel olarak önce iki silindirik blok yan yana ve aralarında boşluk olacak şekilde kullanılmıştır. Devamında tek silindirik ve diğerlerine göre biraz daha büyük boyutlarda olan engel aracın geçiş yolunun tam ortası sayılabilecek noktaya yerleştirilmiştir. Son olarak hedefe yakın bir noktaya düz bir engel konulmuştur.

Üçüncü senaryoda aracın belirli bir süre dar bir yolda ilerlemesi ve sonucunda hedefe doğru yönelmesi istenmiştir. Bu durumda, kavisli ve yeterince dar bir yola araç konulmuştur. Etraftaki engeller bir bakıma refüj görevi görmektedir. Aracın bu yolu hiç çarpmadan tamamlaması istenmektedir. Araç yolu tamamlayıp engelsiz ortama çıktığında hedefe ulaşması beklenmektedir.

Dördüncü ve son olarak aracın sıkıştığı durumlarda nasıl tepki vereceği incelenmek istenmiştir. Bu durumda u şeklinde bir engel tasarlanıp, araç doğrudan bu

engelle karşılaşacak şekilde engel yerleştirilmiştir. Aracın engeli aşmak için geriye doğru hareket edip yeniden manevra yaparak engelden kurtulması ve hedefe doğru ilerlemesi beklenmektedir. Yapılan testler için seçilen konumlar ve hedefe olan mesafe bilgisi Çizelge 5.3’de verilmiştir. GPS için 2 metre yarıçapında hassasiyet ile konum bilgisi alındığı göz önünde tutularak hedef için 80 metre mesafe belirlenmiştir.

Çizelge 5.3 Saha testi konum değerleri

Başlangıç Enlem	Başlangıç Boylam	Hedef Enlem	Hedef Boylam	Mesafe
37°57'55.9"N	32°48'01.8"E	37°57'55.0"N	32°48'04.5"E	≈ 80 m

5.2.1 Senaryo 1 saha testi çalışması

Araç bu senaryoda üç kez test edilmiştir. Başlangıçta araç yönü kuzeye doğru konumlandırılmış ve doğudaki bir konuma gitmesi istenmiştir. Hedef konum ve başlangıç konumları sabit tutulmuştur.



Şekil 5.6 Senaryo 1 için otonom sürüş kontrolü test 1 davranışı

Şekil 5.6’de kaydedilen birinci test için aracın kuş bakışı hareketi gösterilmiştir. İzlenen yol üzerinde yer alan noktalar belirli zamanlarda alınan konum bilgileridir. Test esnasında araç kuzey yönüne doğru ilerleyip hedefe doğru bir dönüş yapmıştır. Araç hedefe doğru ilerlerken belirli aralıklarda sağ ve sol yönlerde hareket ettiği görülmektedir. Son noktada araç hedefe tam olarak ulaşmadan durmuştur. Sonuç olarak araç hedef konuma doğru gidebilme eğilimi gösterdiği gözlemlenmiştir. Hedefe tam olarak ulaşamasa da kat edilen mesafe göz önüne alındığında hedefe yeterince yaklaştığı gözlenmektedir.

Şekil 5.7’de kaydedilen ikinci test için aracın kuş bakışı hareketi gösterilmiştir. Araç yönelimi incelendiğinde daha düz bir hareket yapıp devamında dönme eğilimine girdiği görülmektedir. Bununla beraber araç, hedefe yönelmiş ve hedefe yaklaşmada hatasını telafi etmiştir. Sonuçta aracın beklenen şekilde hedefe ulaşabildiği gözlenmiştir.



Şekil 5.7 Senaryo 1 için otonom sürüş kontrolü test 2 davranışı

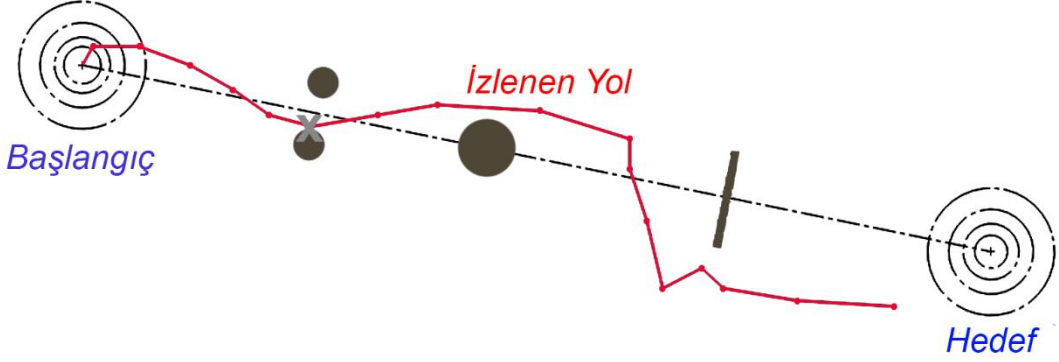
Şekil 5.8’de kaydedilen üçüncü test için aracın kuş bakışı hareketi gösterilmiştir. Aracın hareketi incelendiğinde başlangıçta hedef konuma doğru daha hızlı yöneldiği ve diğer iki teste göre başlangıç davranışını daha iyi yaptığı görülmektedir. Araç hedefe doğru ilerlerken belirgin bir sapma göstermese de hedefe yaklaştığı anda küçük bir sapma meydana gelmiştir. Sonuç olarak aracı hedefin arkasında kaldığı görülmüştür. Kat edilen mesafeye kıyasla ortaya çıkan hata oldukça küçük olup yapılan test çalışması ile otonom sürüş hedefine ulaşılması sağlanmıştır.



Şekil 5.8 Senaryo 1 için otonom sürüş kontrolü test 3 davranışı

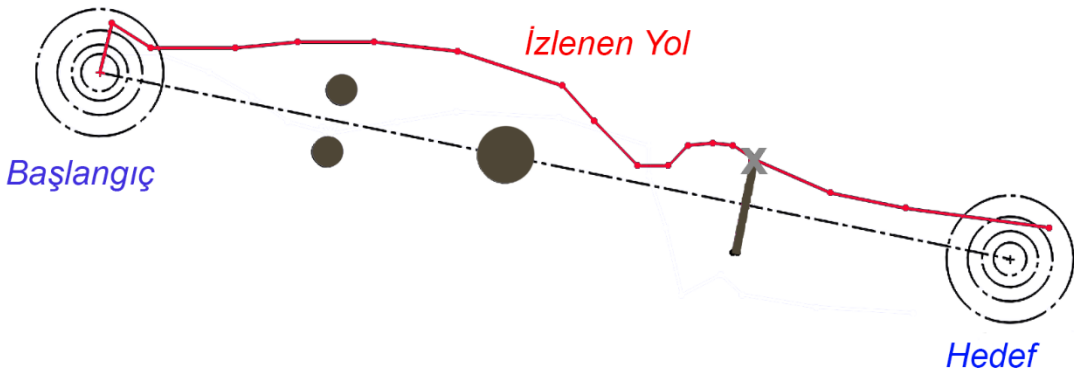
5.2.2 Senaryo 2 saha testi çalışması

Araç bu senaryoda yine üç kez test edilmiştir. Test esnasında aracın önüne çıkan engellere karşı hedefe ulaşmak için kaçış davranışları gözlemlenmiştir. Aracın hedefe ulaşması yanında engellerden kaçış uygulamasında başarısı bu test ile sınanmıştır.



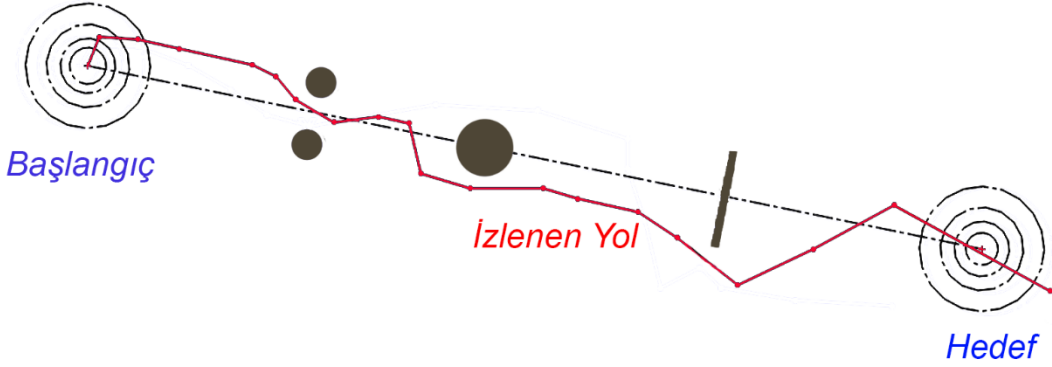
Şekil 5.9 Senaryo 2 için otonom sürüş kontrolü test 1 davranışı

Şekil 5.9’da kaydedilen birinci test için aracın kuş bakışı hareketi gösterilmiştir. Araç hedefe doğru ilerlerken iki silindirik engel ile karşılaştığında, sağ tarafta kalan engeli içeriden çarparak sol tarafa doğru hareketini sürdürmüştür. İkinci büyük silindirik engelden kaçış gerçekleştirilmiş olup sonraki geniş yüzeyli engeli gelmeden önce yine başarılı olarak kaçış yaparak hedefe doğru yönelmiştir.



Şekil 5.10 Senaryo 2 için otonom sürüş kontrolü test 2 davranışı

Şekil 5.10’da kaydedilen ikinci test için aracın kuş bakışı hareketi gösterilmiştir. Araç hareketi incelendiğinde yazılımsal olarak verilen 10° tolerans değerine sadık kalacak şekilde hedefe doğru hareketini sağlamıştır. Senaryo 2 için aracın önüne konulan 2 adet küçük ve 1 adet büyük silindirik engellerin solundan geçerek bu engelleri aşmıştır. Harekete devam ederken istikamet açısı tolerans değerinin dışına çıkmış ve yeniden istikamet açısına dönmek için sağa doğru manevra yapmış ve tekrar hedefe doğru yeniden harekete geçmiştir. Bir sonraki engeli algıladığında sola doğru yüksek direksiyon açısı ile manevra yapmıştır ve araç sağ tarafını engele sürterek engeli aşmıştır. Son olarak araç başka engelle karşılaşmayıp hedefine ulaşmıştır.

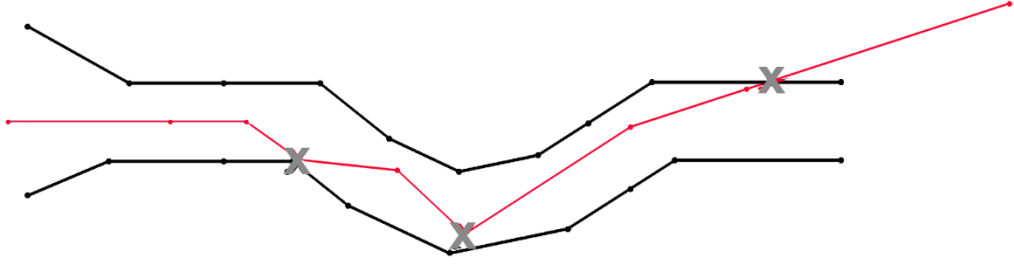


Şekil 5.11 Senaryo 2 için otonom sürüş kontrolü test 3 davranışı

Şekil 5.11’de kaydedilen üçüncü test için aracın kuşbakışı hareketi gösterilmiştir. Araç hedefe doğru ilerlerken 2 adet yan yana duran silindirik engelden solda olanı algılamış ve engelin sağına doğru hareket etmiştir. Bu manevra esnasında sağ taraftaki silindirik engeli de algılayan araç iki engelin ortasından geçerek bu engelleri aşmıştır. Ardından tam karşısına konumlanmış büyük silindirik engeli sağa doğru manevra yaparak aşmıştır. Sonra geniş yüzeyli engeli de aşmış fakat bu engeli aşma esnasında hedef açısı tolerans değerinin dışına çıkmış ve tekrar hedef açısını bulup hedefe ulaşmıştır.

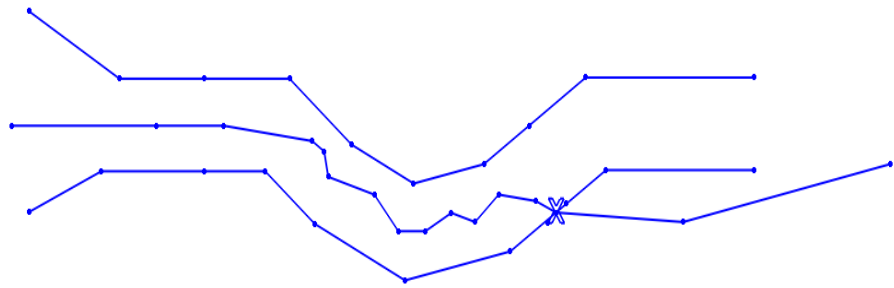
5.2.3 Senaryo 3 saha testi çalışması

Bu senaryoda ilk iki teste göre nispeten daha dar bir yolda ilerleyen araç davranışları gözlenmiştir. Araç bu senaryoda iki kez test edilmiştir.



Şekil 5.12 Senaryo 3 için otonom sürüş kontrolü test 1 davranışı

Şekil 5.12’de kaydedilen birinci test için aracın kuşbakışı hareketi gösterilmiştir. Araç hedefe doğru harekete başladıktan sonra düz bir yol izlemiş ve devamında önündeki engelle yeterince yaklaştığında kaçış gerçekleştirmiştir. Bu kaçıştan sonra araç parkurun sağ bariyerine oldukça yaklaştığı, araç engeli algılamış fakat bariyerlere sürterek engeli aşabilmiştir. Bu manevranın ardından yeniden hedefe doğru hareketi sağlamıştır. Yolun sağa doğru virajını algılayan araç yönünü sağa doğru çevirmiş fakat yine yolun şeklinden dolayı sağ taraftaki bariyerlere çarpıp yeniden hedefe doğru yönelmiştir. Parkurun sonuna yaklaşan araç sol bariyere çarpıp hedefe doğru hareket ettiği gözlemlenmiştir.



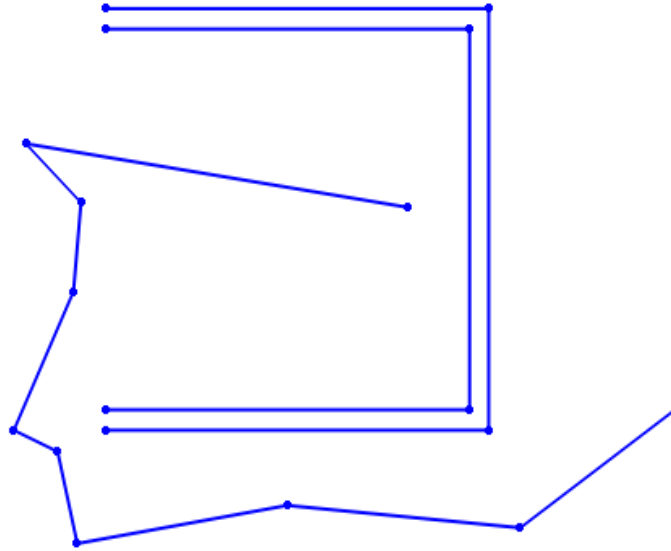
Şekil 5.13 Senaryo 3 için otonom sürüş kontrolü test 2 davranışı

Şekil 5.13’de kaydedilen ikinci test için aracın kuşbakışı hareketi gösterilmiştir. Araç hedefe doğru düz bir çizgide hareket etmiştir. Parkurun sağa doğru tasarlanmış

virajını algılayan araç sağa doğru manevra yaparak engelden kaçmıştır. Bu engelden kaçış esnasında hedef açısını kaybettiğinden yeniden hedef açısına doğru manevra yapmış fakat yeniden engel algılayan araç tekrar sağa manevra yapmıştır. Bu manevra döngüsü birkaç kez tekrar ederek zigzag hareketi yaptığı gözlemlenmiştir. Bu tekrar eden hareketler esnasında bariyerlerin zayıf noktasına dik bir açıyla çarpan araç engeli ezip hedef açısını yeniden bularak hedefe doğru hareket ettiği gözlemlenmiştir. Bu iki testin sonuçlarına göre aracın boyutlarına göre oldukça dar bir parkurda engelleri algılayıp manevra kabiliyeti sınırları içerisinde kalan durumlarda engeli aşmış fakat fiziki olarak manevra kabiliyetinin dışında kalan durumlarda engele çarpmış yada engeli ezdiği gözlemlenmiştir.

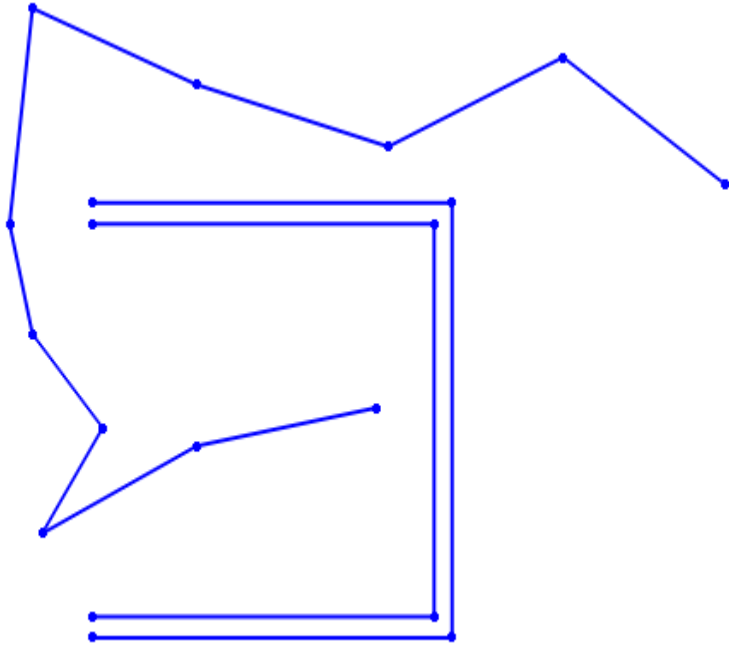
5.2.4 Senaryo 4 test çalışması

Bu senaryoda araç iki kez tez edilmiştir. Test esnasında ileri yönde manevra ile engelin aşılmasının mümkün olmadığı “U” formunda bir engelle karşılaştığında nasıl bir davranış sergileyeceği sınanmıştır. Test esnasında araç doğrudan engele sıkıştığı pozisyona konumlandırılmıştır.



Şekil 5.14 Senaryo 4 için otonom sürüş kontrolü test 1 davranışı

Şekil 5.14’de kaydedilen birinci test için aracın kuşbakışı hareketi gösterilmiştir. Araç üç tarafında manevra kabiliyetinin altında engel olduğunu algılamış ve araç geri yönde hareket etmeye başlamıştır. Araç herhangi yönde engelin ortadan kalktığı bir anda aracın hareketini durdurup hedef açısını hesaplayıp sağa doğru manevra yaparak ileri yönde engeli soluna alacak şekilde hareketine başladığı gözlemlenmiştir. Hedefe doğru hafif sola doğru dönüşler yapan araç engeli yeniden algılayıp sağa doğru hafif manevra ile engeli aşıp hedefine doğru hareket ettiği gözlemlenmiştir.



Şekil 5.15 Senaryo 4 için otonom sürüş kontrolü test 2 davranışı

Şekil 5.15’de kaydedilen ikince test için aracın kuşbakışı hareketi gösterilmiştir. Bu test esnasında araç yönü engele göre hafif sola bakacak şekilde engelin içine konumlandırılmıştır. Araç sıkışık pozisyonda olduğu algılamış ve geri yönde harekete başladığı gözlemlenmiştir. Teste başlamadan önceki konumlandırma şekline göre sağa doğru daha çok yaklaşmış ve solunda engel olmadığı bir anda aracı durdurup hedef açısını hesaplayarak sola manevra yapmış ve ileri yönde hareketine başlamıştır. Bir süre hedefe doğru hareket eden araç sağında engel olduğunu algılamış ve araç hafif sola manevra yaparak engeli sağına almış ve ileri yönde hareketine devam etmiştir. Sağında engelin olmadığı durumda hedef açısını yeniden hesaplayarak sağa doğru dönüşünü

sağlayıp hedefe doğru hareket etmiştir. Yeniden önüne engel çıktığını algılayan araç sola doğru manevra yaparak engeli tamamen aşmış ve hedefine doğru hareketine devam etmiştir.

5.2.5 Değerlendirme

Yapılan tüm saha testleri dikkate alındığında otonom sürüş kontrolü gerçekleştirilen kara aracı; GPS ile konum bilgisi elde etme, mesafe sensörleri ile engel tespiti ve bulanık mantık ile engelden kaçış yetenekleri sayesinde hedefe ulaşma sürecini kabul edilebilir seviyede hata olmakla birlikte başarılı olarak tamamlamıştır. Hedefe olan mesafe göz önünde bulundurulursa; özellikle engelden kaçış uygulamasında ortaya çıkan sapmaların sistem başarımını önemli ölçüde etkilemeyeceği anlaşılmıştır. Yapılan bu tez çalışmasında düşük maliyetli ultrasonik mesafe sensörleri kullanılarak otonom araç sürüş kontrolüne yönelik bulanık mantık tabanlı basit bir uygulama ortaya koyulmuştur. Radar ve LiDAR gibi ortam algılamaya yönelik farklı sensörlerin tek başına veya birlikte kullanılması durumunda otonom sürüş kontrolünde daha etkin ve verimli uygulamaların hayata geçirilmesi mümkündür. Özellikle ultrasonik sensör yanında Radar ve LiDAR sensör kullanılması halinde, gerçekleştirilebilecek sensör füzyonu tekniği uygulaması ile otonom sürüş uygulamasını daha ileri boyutlara taşımak mümkündür.

6 SONUÇ VE ÖNERİLER

Bu tez çalışmasında üzerinde ultrasonik mesafe sensörleri, GPS modülü, Bluetooth modülü, Arduino Mega kontrol kartı bulunan küçük ölçekte bir kara aracı prototip olarak tasarlanmış, engelleri aşarak cep telefonu ile gönderilen hedef konuma ulaşması için otonom sürüş ve bulanık mantık tabanlı engelden kaçış uygulaması gerçekleştirilmiştir. Oluşturulan yazılım kullanılarak hedefe giden yolda değişik konumlarda değişik engellerin var olduğu dört farklı senaryoda otonom sürüş için gerekli saha testleri gerçekleştirilmiştir.

Yapılan testler sonucunda kara aracının uzaktan cep telefonu ile gönderilen hedef konuma otonom sürüş ile yaklaştığı ve önüne çıkan engelleri aşmak üzere engelin konumuna göre ihtiyaç duyulacak manevraları gerçekleştirdiği görülmüştür. Saha testlerinde karşılaşılan zorluklar şöyle listelenebilir:

1. Tahrik motorunun sürülmesinde kullanılan L298N tipi sürücü modülü tahrik için kullanılan DC motoru başarı ile sürmüştür. Fakat aracın ağırlığı arttığında tahrik motorundan çekilen akım da arttığından kullanılan DC motor sürücü modülünün kısmen yetersiz kaldığı anlaşılmıştır.
2. Direksiyon sisteminde kullanılan sg90 model servo motorun her ne kadar katalog bilgilerine bakıldığında prototip aracı yönlendirebilecek torka sahip görünse de özellikle prototip aracın lastiklerinin kauçuk olması ve bundan dolayı zemin ile arasında aşırı sürtünme sebebiyle ihtiyaç duyulan akım değeri sağlanamamıştır. Bu problemin çözümü için harici besleme kaynağı kullanılarak sorun çözülmüştür.
3. Yön belirleme için kullanılan HCM5883L manyetometre ile x ekseninde alınan değerlerin oldukça değişken olduğu gözlenmiştir. X ekseninde alınan değerler -300 ile +600 aralığında olup “map” komutu ile servo motorun çalışma aralığı olan 0-360° arasına getirilmiştir. Normal koşullarda manyetometre sorunsuz çalışmakta olup araç etrafında yüksek manyetik alana sahip (miknatis vb.) malzeme bulunduğu alanda alınan değerlerde değişkenlik sebebiyle araç yönlendirme algoritması kararsız çalışmaktadır.
4. Konum belirleme için kullanılan GPS modülü dış ortamda hava koşulları elverişli iken 2 metre doğrulukla çalışmaktadır. Ayrıca veri transfer hızı düşük olduğundan konum bilgisine dayalı karar verme süreci oldukça uzundur. Bu sebepten aracı engele çarpma ihtimali artmaktadır. Yapılan çalışmada engelden kaçış imkânı sağlamak için aracın hızı sınırlandırılmıştır. Otonom sürüş çalışması yapılan küçük ölçekli araç için

daha yüksek maliyetli ve daha yüksek hassasiyette GPS modülü kullanımı mümkündür.

5. Engel algılamak için kullanılan HC-SR04 serisi ultrasonik mesafe sensörü küçük ölçekli araç ile gerçekleştirilen tez çalışması için yeterli hassasiyete sahiptir. Ultrasonik mesafe sensörü kullanılarak sadece yakında engel var olup olmadığını algılayabilirken, kamera veya LiDAR sensörler kullanılarak hedefe doğru hareket ederken karşılaşılan engelin fiziksel boyutu ve geometrik şekli de algılanabilir.
6. Arduino kullanımı her ne kadar gerekli olan kütüphane ve örnek uygulama kodlarının yaygın olması sebebiyle uygulama geliştirme süreçlerini kısaltsa da ARM tabanlı yüksek performanslı mikrokontrolör kullanılarak bulanık mantık tabanlı engelden kaçış algoritmalarının daha hızlı çalıştırılması mümkün kılınabilir.

Tez kapsamında yapılan prototip tasarım, algoritma geliştirme çalışması ve saha testleri ile sivil ve askeri uygulamalarda kullanılan gerçek boyutlarda kara araçları için otonom sürüşe yönelik bir model oluşturulmaya çalışılmıştır. Gelecekte yapılacak çalışmalarda kullanılan donanım bileşenleri ve algoritmaların performansının artırılmasıyla daha büyük boyutlarda kara araçlarının otonom sürüşüne yönelik uygulamalar geliştirilmesi hedeflenmektedir.

7 KAYNAKLAR

- Ahn, K. J., & Kang, Y. (2019). A particle filter localization method using 2d laser sensor measurements and road features for autonomous vehicle. *Journal of Advanced Transportation*, 2019. <https://doi.org/10.1155/2019/3680181>
- Aki, K., & Dirik, A. E. (2020). Derin Öğrenme Tabanlı Ve Pıd Kontrol Tabanlı Sürücüsüz Araç Sistemleri. *Mühendislik Bilimleri ve Tasarım Dergisi*, 8(5), 306-316. <https://doi.org/10.21923/jesd.829598>
- Altaş, İ. (1999). Elektromekanik-3e. İçinde *Bilesim yayıncılık A.Ş* (C. 62).
- Arduino, M. (2023). *Arduino® MEGA 2560 Rev3*.
- Armagan, E., & Kumbasar, T. (2018). A fuzzy logic based intelligent autonomous vehicle control system design in the TORCS game environment. *Pamukkale University Journal of Engineering Sciences*, 24(8), 1435-1442. <https://doi.org/10.5505/pajes.2018.77910>
- Bayram, A., Almalı, M. N., & Al-Naqshbandı, F. M. (2023). Path following of an unmanned ground vehicle with GPS feedback using model predictive control method. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 38(1), 345-355. <https://doi.org/10.17341/gazimmfd.1024463>
- Beşdok, E., & Özçelik, A. E. (2009). Kalman Filtreleme Yöntemiyle Otonom Hava Araç Navigasyonunda Gps/Ins Entegrasyonu. İçinde *Kadastro Mühendisleri Odası* (C. 12).
- Bingöl, M. S., Kaymak, Ç., & Uçar, A. (2019). Derin Öğrenme Kullanarak Otonom Araçların İnsan Sürüşünden Öğrenmesi. İçinde *Fırat Üniversitesi Müh. Bil. Dergisi* (C. 31, Sayı 1).
- Çetinkaya, A. (2017). *Otonom Bir Robotun Bulanık Kontrolör Yaklaşımı İle Konum Kontrolü*.
- Elec, F. (2017). *Ultrasonic Ranging Module HC-SR04*. www.ElecFreaks.com
- Faisal, A., Kamruzzaman, M., Yigitcanlar, T., & Currie, G. (2019). Understanding autonomous vehicles. *Journal of Transport and Land Use*, 12(1), 45-72. <https://doi.org/10.2307/26911258>
- Fan, Y. C., Yelamandala, C. M., Chen, T. W., & Huang, C. J. (2021). Real-Time Object Detection for LiDAR Based on LS-R-YOLOv4 Neural Network. *Journal of Sensors*, 2021. <https://doi.org/10.1155/2021/5576262>
- Geleceği Yazarlar Ekibi. (2020). *Bluetooth ile İletişim*. <https://gelecegiyazarlar.turkcell.com.tr/konu/egitim/arduino-201/bluetooth-ile-iletisim>.

- Gul, F., Rahiman, W., & Nazli Alhady, S. S. (2019). A comprehensive study for robot navigation techniques. İçinde *Cogent Engineering* (C. 6, Sayı 1). Cogent OA. <https://doi.org/10.1080/23311916.2019.1632046>
- Gürtaş, S. (2018). *Otonom Araç Sürüş Destek Sistemleri Ve Yapay Zeka Uygulamaları*.
Hassani, I., Maalej, I., & Rekik, C. (2018). Robot Path Planning with Avoiding Obstacles in Known Environment Using Free Segments and Turning Points Algorithm. *Mathematical Problems in Engineering*, 2018. <https://doi.org/10.1155/2018/2163278>
- Honeywell. (2023). *3-Axis Digital Compass IC HMC5883L*. www.honeywell.com
- Hülako, H., & Kapucu, S. (2018). Düşük Maliyetli GPS Tabanlı Otonom Bir İnsansız Kara Aracının Tasarımı ve Yapılması. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*. <https://doi.org/10.29109/gujsc.421665>
- Jo, K., Lee, M., Lim, W., & Sunwoo, M. (2019). Hybrid Local Route Generation Combining Perception and a Precise Map for Autonomous Cars. *IEEE Access*, 7, 120128-120140. <https://doi.org/10.1109/ACCESS.2019.2937555>
- Jo, K., & Sunwoo, M. (2014). Generation of a precise roadway map for autonomous cars. *IEEE Transactions on Intelligent Transportation Systems*, 15(3), 925-937. <https://doi.org/10.1109/TITS.2013.2291395>
- Karaali, C., & Yıldırım, Ö. (2020). *Global Konum Belirleme Sistemi (Gps)*.
- Kıyak, E., & Kahvecioğlu, A. (2003). *Bulanık Mantık Ve Uçuş Kontrol Problemine Uygulanması Arş. Grv. Emre Kıyak Yrd. Doç. Dr. Ayşe Kahvecioğlu*.
- Li, T., Zhang, F., Gao, X., Xu, H., & Ji, S. (2023). The Control System Design Of A Omni-Directional Mobile Logistics Sorting Vehicle Based On Stm32. *U.P.B. Sci. Bull., Series D*, 82, 2020.
- Martinez, M., Roitberg, A., Koester, D., Schauerte, B., & Stiefelhagen, R. (2017). *Using Technology Developed for Autonomous Cars to Help Navigate Blind People*. <http://cvhci.anthropomatik.kit.edu/>
- Nennioğlu, A. K., & Köroğlu, T. (2018). *Otonom Araçlarda Hareket Planlaması* (C. 1, Sayı 2).
- O. Hachour. (2008). *Path planning of Autonomous Mobile robot*.
- Royo, S., & Ballesta-Garcia, M. (2019). An overview of lidar imaging systems for autonomous vehicles. *Applied Sciences (Switzerland)*, 9(19). <https://doi.org/10.3390/app9194093>
- o
Sariff, N., & Buniyamin, N. (2006). *An Overview of Autonomous Mobile Robot Path Planning Algorithms*.

- Silva de Lima, R., Izidoro, S., Gomes de Almeida, W., Bernardes, G., & Silva de Lima, B. (2019, Aralık 4). *Autonomous Vehicle Sensoring Using Low Level Can-Bus And Stm32 Microcontroller*. <https://doi.org/10.26678/abcm.cobem2019.cob2019-0664>
- Şanlı, E. (2018). *Yapay Sinir Ağı Kontrollü Otonom Rc Araç Uygulaması*.
- Towe, P. (2017). *SG90 9 g Micro Servo*.
- Ünsal, S., Alişkan, İ., & Mühendisliği Bölümü Bülent Ecevit Üniversitesi, E.-E. (t.y.). *Mamdani ve Takagi-Sugeno Çıkarım Yöntemlerine Sahip Bulanık Mantık Denetleyicilerin Özgün Yazılım ve Araç Kutusu Performans Analizi Performance Analysis of Fuzzy Logic Controllers Having Mamdani and Takagi-Sugeno Inference Methods By Using Unique Software and Toolbox*.
- Veerachai Malyavej, Peerapong Torteeka, & Siripong Wongkharn. (2009). *2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology : ECTI-CON 2009 : May 6th-9th, 2009, Ambassador City Jomtien, Pattaya, Chonburi, Thailand*.
- Vikipedi. (2023). *App Inventor*. https://tr.wikipedia.org/wiki/App_Inventor.
- Zadeh, L. A. (1965). Fuzzy S e t s *. İçinde *INFOR~ATIO~ AND CONTROL* (C. 8).