TEKNOFEST

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

ROBOTAKSİ – BİNEK OTONOM ARAÇ YARIŞMASI

KRİTİK TASARIM RAPORU

TAKIM ADI: Team IMU AV

TAKIM ID:458926

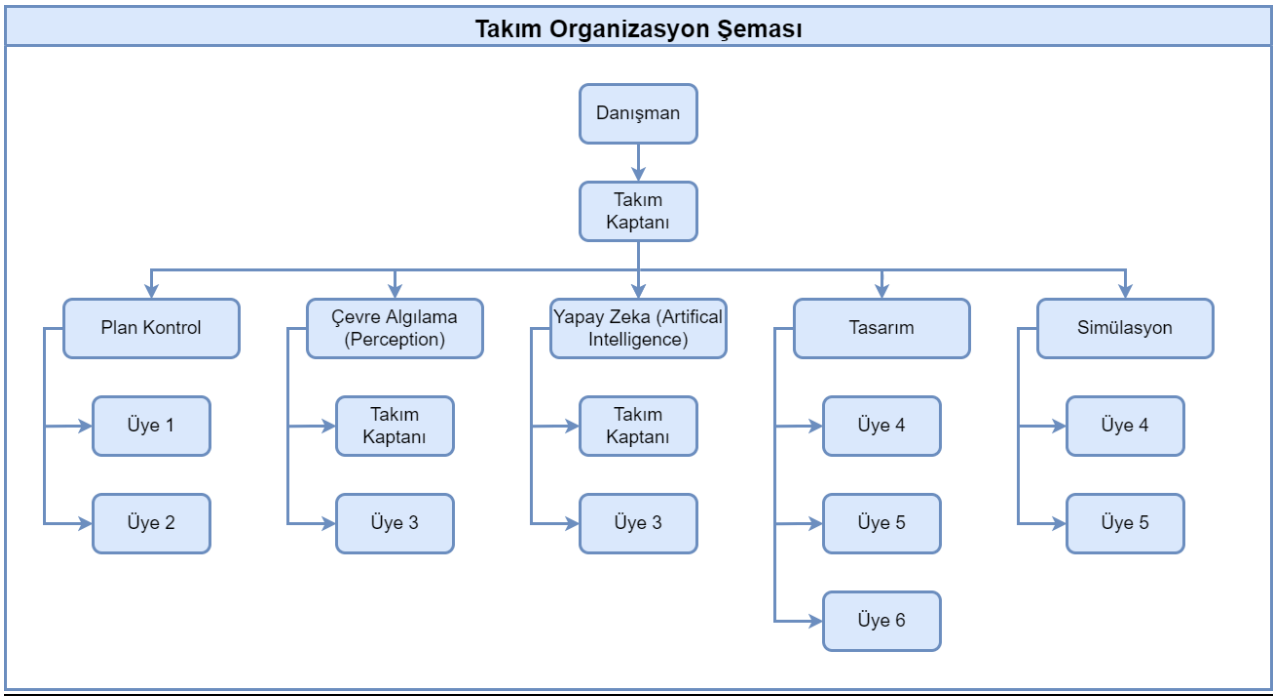
İÇERİK

1. Takım Organizasyonu 3
2. Ön Tasarım Raporu Değerlendirmesi 7
3. Araç Mekanik Özellikleri 9
4. Donanım Mimarisi 21
5. Yazılım Mimarisi 32
6. Özgün Bileşenler 54
7. Test 55
8. Referanslar 63

1. Takım Organizasyonu

Team IMU takımı, yaklaşık 4 yılı aşkın süredir otonom kabiliyete sahip hava, kara ve suüstü robotik sistemler üzerinde çeşitli çalışmalar yürütmektedir. Team IMU AV ekibi kara aracı ile ilgili çalışmaları yürütmektedir. Takım bu süreç boyunca elde edilen bilgi birikiminden istifade ederek Robotaksi-Binek Otonom Araç Yarışmasına katılıp, yarışma kapsamındaki görevleri başarılı bir şekilde gerçekleştirmeyi amaçlamaktadır.

Team IMU AV takımı, akademik danışman, takım kaptanı dahil olmak üzere toplamda 8 üyeden oluşmaktadır. Takım organizasyon şeması Şekil 1'deki gibidir.



Şekil 1. Takım Organizasyon Şeması

Takım danışmanı İstanbul Medeniyet Üniversitesi (İMÜ) Elektrik Elektronik Mühendisliğinde (EEM) tam zamanlı öğretim üyesidir. Doktorasını Boğaziçi Üniversitesi'nde çoklu-robot sistemleri üzerine yapmıştır. Doktora-sonrası araştırmalarını ise Minnesota Üniversitesi'nde Bilgisayar Bilimleri alanında, yüzen ve çok rotorlu uçan sistemler için hedef konumlandırma ve takip algoritmaları geliştirerek gerçek robotik sistemler üzerinde gerçeklenmesini yapmıştır. İMÜ'de Saha Robotiği Laboratuvarı'nı kurmuştur. Laboratuvar bünyesinde hava, su üstü ve kara robotları üzerinde çalışmalar gerçekleştirilmektedir.

Takım kaptanı, İMÜ EEM 4. Sınıf öğrencisidir. C/C++, Python, MATLAB ve Java yazılım dillerinin yanında ROS (Robot Operating System), Proteus devre çizim programını bilmektedir. Takımın koordinasyonundan sorumludur. Takım içerisinde ROS tabanlı çevre algılama yazılımlarının geliştirilmesi, mekanik tasarım ve elektrik-elektronik bileşenlerin entegrasyonu faaliyetlerinde rol alacaktır.

Takım üyesi 1, EEM bölümünde 4. sınıf öğrencisidir. C/C++, Python programlama dillerini bilmektedir. ROS ve Gazebo ortamları hakkında deneyim sahibidir. Takım içerisinde hareket planlama ve araç kontrol kısımlarından sorumludur.

Takım üyesi 2, EEM bölümü 4. sınıf öğrencisidir. C/C++, Python programla dillerini bilmektedir. ROS ve Gazebo ortamları hakkında deneyim sahibidir. Takım içerisinde hareket planlama ve araç kontrol kısımlarından sorumludur.

Takım üyesi 3, EEM bölümü 2.sınıf öğrencisidir. C ve Python programlama dillerini bilmektedir. Takım içerisinde yapay zeka ve çevre algılama kısımlarından sorumludur.

Takım üyesi 4, EEM bölümü yüksek lisans programında 2. sınıf öğrencisidir. C/C++, Python, Matlab-Simulink programlama dillerini, AutoCAD, Carla, Eagle, Proteus programlarını bilmektedir. Takım içerisinde otonom araç simülasyonu, hareket planlama ve araç kontrol kısımlarından sorumludur.

Takım üyesi 5, EEM bölümü doktora programında 1. sınıf öğrencisidir. C#, C/C++, Python, JavaScript dillerini bilmektedir. ROS ve Gazebo ortamları hakkında deneyim sahibidir. Takımda içerisinde simülasyon ve araç kontrol kısımlarından sorumludur.

Takım üyesi 6, EEM yüksek lisans 2. sınıf öğrencisidir. C/C++, Python programlama dillerinin yanı sıra ROS ve Gazebo gibi ortamlarda çalışmak için tecrübe sahibidir. Takımda elektrik, elektronik, elektromekanik ve mekanik revizyon çalışmalarında rol almaktadır.

Robotaksi yarışması kapsamında planlanan iş paketleri 5 ana kısımdan oluşmaktadır. İş paketlerinin listesi ve ilgili iş paketlerinden sorumlu takım üyeleri Tablo 1'de gösterilmiştir. İş paketlerinde yapılacaklara dair detaylar aşağıda verilmiştir.

İş Paketi 1 - Simülasyon: Otonom aracın bilgisayar ortamına aktarılmasıdır. Bilgisayar ortamında araç üzerinde testler yapılabilir. Araç üzerindeki testlerin bilgisayar ortamında gerçekleştirilmesi, gerçek hayatta oluşabilecek kazaların önüne geçmektedir. Aynı zamanda zaman ve maddi tasarruf sağlamaktadır. Simülasyon ortamı gerçek hayata ne kadar yakın hale getirilirse maliyet o kadar düşer. "Simülasyon" iş paketinde aracın ve çevrenin modellenmesi, sensörlerin simülasyon ortamına aktarılması, araç üzerine yazılan algoritmaların simülasyon ortamında denenmesi amaçlanmaktadır.

İş Paketi 2 - Hareket Planlama: Otonom aracın verilen hedefe ulaşabilmesi için rota oluşturulması gerekmektedir. Otonom araç rota boyunca engellerle karşılaşabilir. Engellerle karşılaştığı zaman da rotayı bulunduğu mevcut şartlara göre yeniden oluşturması gerekmektedir. "Hareket Planlama" iş paketi, rota oluşturma, şerit takibi, durağa girme ve duraktan ayrılma, park etme, engelle karşılaşınca yeni rota oluşturma gibi işlemlerin gerçekleştirilmesini hedeflemektedir.

İş Paketi 3 - Çevre Algılama: Aracın otonom olmasını için bulunduğu mevcut konumu ve çevresindeki nesneleri algılaması gerekmektedir. Algılama çeşitlilikleri olması ve farklı sensör türleri sebebiyle, otonom aracın çevresinden aldığı bilgileri anlamlandıracak alt sistemlerin olması gerekmektedir. "Çevre Algılama" iş paketi, trafik işaretlerinin ve trafik ışıklarının tespiti, yol üzerinde karşılaşılan engellerin algılanması, yol üstündeki şeritlerin bulunması, sürülebilir yolların tespiti gibi detayl alt faaliyetlerden oluşmaktadır.

İş Paketi 4 - Araç Kontrol: Otonom aracı kontrol edebilmek, sistem dinamiğini anlamak ve yazılan kodlarla uyumunu gerçekleştirmek amacıyla aracın matematiksel modelinin elde edilmesi amaçlanmaktadır. Elde edilen denklemlere ve yapılan hesaplamalara göre aracın kararlı (stabil) hareket edebilmesini sağlamak için gerekli kontrol parametrelerinin oluşturulması gerekmektedir.

“Araç Kontrol” iş paketi, otonom aracın uzaktan haberleşme sistemini, tekerlek ve direksiyon arasındaki bağlantıların gerçekleştirilmesi gibi durumları kapsamaktadır.

Tablo 1. İş Paketleri ve Görev Dağılım Tablosu

İŞ PAKETİ ADI	İŞ PAKETİ DETAYI	GÖREVLİLER
SİMÜLASYON	Literatür Taraması	Üye 4 Üye 5
	Parkurun, Aracın ve Alt Sistemlerin 3B Modellenmesi	
	Aracın, Parkurun ve Algoritmaların Gazebo Simülasyon Ortamına Aktarılması	
	Test süreci	
HAREKET PLANLAMA	Literatür Taraması	Takım Kaptanı Üye 1 Üye 2 Üye 3
	Şerit Takibi ve Dönüş Algoritmalarının Oluşturulması	
	Durak – Park Etme Algoritmalarının Oluşturulması	
	Ani Fren Algoritmasının Oluşturulması	
	Test ve Gerekli Düzeltmelerin Yapılması	
ÇEVRE ALGILAMA	Literatür Taraması	Takım Kaptanı Üye 3
	Trafik işaret ve Işıklarının Algılanması için Veri Seti Oluşturulması ve Etiketlenmesi	
	Nesne Algılama Model Eğitimi ve Test Süreci	
	2B Lazer Tarayıcı Sensör Verilerinden Engel Tespiti	
	Şerit Takip Algoritmasının Oluşturulması	
	Çevre Algılama Yöntemlerinin Simülasyon Ortamında Test Edilmesi ve Gerekli Düzeltmelerin Yapılması	
ARAÇ KONTROL	Literatür Taraması	Üye 1 Üye 2
	Aracın Kontrol Parametrelerinin Çıkarılması	
	Matematiksel Denklemlerin ve Kontrol Diyagramların Çıkarılması	
	Hız Kontrolü Sistemlerinin Oluşturulması	
	Tekerlek Açı Kontrol Sistemlerinin Oluşturulması	
	Uzaktan Haberleşme Sistemlerinin oluşturulması	
	Güvenlik Önlemlerinin Oluşturulması	
	Test ve Gerekli Düzeltmelerin Yapılması	
ELEKTRİK ELEKTRONİK, ELEKTROMEKANİK VE MEKANİK REVİZYON	Literatür Taraması	Üye 5 Üye 6
	Araç Yapısı ve Alt Sistemlerinin Belirlenmesi	
	Güç Yönetim Sistemlerinin Oluşturulması	
	Elektrik/Elektronik Devre Çizimlerinin Yapılması	
	Malzeme Tedariki	
	Elektromekanik Sistemlerin oluşturulması	
	Montaj ve Modifikasyon Çalışmaları	
	Saha Testleri ve Gerekli Düzeltmelerin yapılması	

İş Paketi 5 - Mekanik, Elektromekanik, Elektrik-Elektronik Aksam: Otonom aracın meydana gelebilmesi için, aracın fiziksel olarak oluşturulması gerekmektedir. "Mekanik, Elektromekanik, Elektrik-Elektronik Aksam" iş paketi, aracın mekanik, elektromekanik, elektrik-elektronik sistemlerinin tasarlanarak oluşturulmasını ve bu sistemlerin birbirleri içerisinde entegrasyonunu kapsamaktadır.

İş paketlerine dair iş-zaman çizelgesi Tablo-2'de verilmiştir.

Tablo 2. İş-Zaman Çizelgesi

İş Paketleri	Süresi	Şubat	Mart	Nisan	Mayıs	Haziran	Temmuz
I. SİMÜLASYON	3 Ay						
Literatür taraması	1 Ay						
Parkurun, Aracın ve Alt Sistemlerin 3B Modellenmesi	1 Ay						
Aracın, Parkurun ve Algoritmaların Gazebo Simülasyon Ortamına Aktarılması	1 Ay						
Test süreci	1 Ay						
II. HAREKET PLANLAMA	6 Ay						
Literatür Taraması	1 Ay						
Şerit Takibi ve Dönüş Algoritmalarının Oluşturulması	3 Ay						
Durak-Park etme Algoritmalarının oluşturulması	3 Ay						
Ani Fren Algoritmasının Oluşturulması	3 Ay						
Test ve Gerekli Düzeltmelrin Yapılması	3 Ay						
III. ÇEVRE ALGILAMA	6 Ay						
Literatür Taraması	1 Ay						
Trafik İşaret ve Işıklarının Algılanması için Veri Seti Oluşturulması ve Etiketlenmesi	1 Ay						
Nesne Algılama Model Eğitimi ve Test Süreci	1 Ay						
Şerit Tespiti Algoritmalarının Oluşturulması	1 Ay						
2B Lazer Tarayıcı Sensör Verilerinden Engel Tespiti	1 Ay						
Çevre Algılama Yöntemlerinin Simülasyon Ortamında Test Edilmesi ve Gerekli Düzeltmelerin Y	2 Ay						
IV.ARAÇ KONTROL	6 Ay						
Literatür Taraması	1 Ay						
Aracın Kontrol Parametrelerinin Çıkarılması	2 Ay						
Kontrol Diyagramlarının ve Matematiksel Denklemlerin Çıkarılması	2 Ay						
Hız Kontrolü Sistemlerinin Oluşturulması	3 Ay						
Tekerlek Açı Kontrolü Sistemlerinin Oluşturulması	4 Ay						
Uzaktan Haberleşme Sistemlerinin Oluşturulması	3 Ay						
Güvenlik Önlemlerinin Oluşturulması	3 Ay						
Test ve Gerekli Düzeltmelerin Yapılması	3 Ay						
V. ELEKTRİK, ELEKTRONİK, ELEKTROMEKANİK ve MEKANİK REVİZYON	6 Ay						
Literatür Taraması	1 Ay						
Araç Yapısı ve Alt Sistemlerinin Belirlenmesi	2 Ay						
Güç Yönetim Sistemlerinin Oluşturulması	1 Ay						
Elektrik/Elektronik Devre Çizimlerinin Yapılması	2 Ay						
Malzeme Tedarik	3 Ay						
Elektromekanik Sistemlerin Oluşturulması	3 Ay						
Montaj ve Modifikasyon Çalışmaları	3 Ay						
Saha Testleri ve Gerekli Düzeltmelerin yapılması	3 Ay						

2. Ön Tasarım Raporu Değerlendirmesi

Bu bölümde ön tasarım raporu değerlendirmesi yapılarak, ön tasarım raporundan sonraki değişiklikler ve değişikliklerin sebepleri açıklanmıştır. Ön tasarımda planlanan bütçe ve son bütçe arasında karşılaştırma yapılarak tablolar halinde verilmiştir.

Tablo 3. Planlanan eski bütçe

MALZEME	TUTAR	OLASI RİSK VE ÖNLEMLERİ
UM7 Orientation Sensör	3.994,09 TL	Malzeme tedarik kısmında aksamalar veya gecikmeler yaşanması sonucu yedek malzeme veya alternatif ürün kullanma
Zed 2 Kamera	12.748,16 TL	
Nvidia Jetson Xavier NX	6.6000 TL	
Velodyne VLP-16 Lidar	58.000 TL	
Rplidar S1	16.416 TL	
KORMAS 24V Elektronik Devir Ayarlayıcı KORMAS 24V PWM Modülü	500 $	
Logitech C920 PRO HD WEB Kamerası	1.597 TL	

Ön tasarım raporunda araç üzerinde sinyal lambalarının eklenmesi planlanmış ancak bu sinyaller lambalarının nasıl ekleneceği, bağlantılarının nasıl yapılacağına değinilmemişti. Kritik tasarım raporunda sinyal lambalarının bağlantı şeması eklendi. Şekil 22'deki Harness diyagramında sinyal lambalarının bağlantı şeması gözükmektedir. Ayrıca sinyal lambalarının kontrolüne ilişkin detaylı bilgi Bölüm 3'teki donanım mimarisi kısmında bahsedilmiştir.

Ön tasarım raporunda aracın sadece cad çizimi bulunmaktaydı. Araç çizimini detaylandırmak amacıyla Lumion programı üzerinden render alınmış ve tasarım üzerine renklendirmeler gerçekleştirilmiştir. Şekil 13-14-15'de aracın ön tasarım raporundaki cad çizimleri gözükürken, Şekil 16-17-18-19'da aracın render alınmış hali gözükmektedir.

Tablo 4. Gerçekleşen bütçe

MALZEME	TUTAR	OLASI RİSK VE ÖNLEMLERİ
UM7 Orientation Sensör	3.994,09 TL	Tedarik edildi
Zed 2 Kamera	12.748,16 TL	Tedarik edildi
Nvidia jetson Xavier NX	6.6000 TL	Tedarik edildi
Velodyne VLP-16 Lidar	58.000 TL	Tedarik edilemedi Yerine alternatif ürün olarak Rplidar S1 kullanıldı
Rplidar S1	14.455 TL	Tedarik edildi
KORMAS 24V Elektronik Devir Ayarlayıcı KORMAS 24V PWM Modülü	500 $	Tedarik edildi
Logitech C920s PRO HD WEB Kamerası	1.597 TL	Tedarik edildi

Ön tasarım raporunda Harness diyagramına yer verilmemişti. Basit bağlantı şemaları bulunmakla beraber kapsamlı bir bağlantı şemasıda bulunmamaktaydı. Bu sebeple kritik tasarım raporunda bağlantı şemaları detaylandırılarak, Harnes diyagramı çizilmiştir. Bu sayede Harnes diyagramına bakılarak sistemin yapısı hakkında bilgi sahibi olunabilir. Şekil 22'de Harnes diyagramı gözükmektedir.

Kritik tasarım raporunda Harnes diyagramına ek olarak, elektronik bileşenlerin ve sensörlerin bağlantılarının yapıldığı fotoğraf eklenmiştir. Bu fotoğraf mevcut sistem üzerinden alınmıştır. Ön tasarım raporu sürecinden sonra sistem ayağa kaldırılmış ve bağlantılar kurularak elektronik bileşenlerin mevcut bağlantılarla çalışıp çalışmadığı test edilmiştir. Sistemin çalıştığı ortaya konmuştur. Şekil 20 ve Şekil 21'de resimler detaylı olarak ele alınmış ve açıklanmıştır.

Ön tasarım raporunda aracın mekanik özellikler başlığı altında bulunan, elektronik direksiyon (steer by wire), elektronik fren (brake by wire) ve elektronik gaz (pedal by wire) konuları detay verilmeden yazılmıştı. Kritik tasarım raporunda ise araç mekanik özellikleri kısmında detaylı

olarak ele alınmıştır. Ayrıca mevcut araç üzerinde nasıl çalıştıkları, donanımsal ve elektronik bağlantıları şemalarla gösterilmiştir. Şekil 2 ve Şekil 3'te bu şemalar gösterilmiştir.

Ürün kırılım yapısı ön tasarım raporunda bulunmamaktadır. Kritik tasarım raporunda detaylı ve derli toplu bir gösterim elde etmek amacıyla ürün kırılım yapısı eklenmiştir.

Ön tasarım raporunda güvenlik önlemlerinden bahsedilirken, kişisel önlemlerine değinilmemişti. Kritik tasarım raporu yazılırken bu durum gözden geçirildi. Kişisel güvenlik önlemleri kısmı eklendi. Ayrıca güvenlik önlemleri detaylandırıldı.

Batarya yönetim sistemine ön tasarım raporunda değinilmemişti. Kritik tasarım raporunda donanım mimarisi altında, 4.4. başlık olarak batarya yönetim sistemine değinilmektedir. Nasıl gerçekleştirildiğinden bahsedilmektedir.

Ön tasarım raporunda bulunan iş zaman çizelgesinde güncelleme gerçekleştirilmiştir. Mevcut duruma bakılarak, zamansal olarak planlama farklılıkları oluşmuştur. Kritik tasarım raporunda bu durumlar incelenerek revize edilmiştir. Kritik tasarım raporunda yeni iş zaman çizelgesine yer verilmiştir.

Ön tasarım raporundaki hareket planlama kısmı detaylandırılmıştır. Yeni şemalar eklenerek çalışma prensibinin anlaşılabilirliği arttırılmıştır. Ayrıca çalışma prensibi mevcut durum göz önüne alınarak detaylı bir şekilde ele alınmıştır. Kritik tasarım raporunun yazılım mimarisi başlığında bahsedilmiştir.

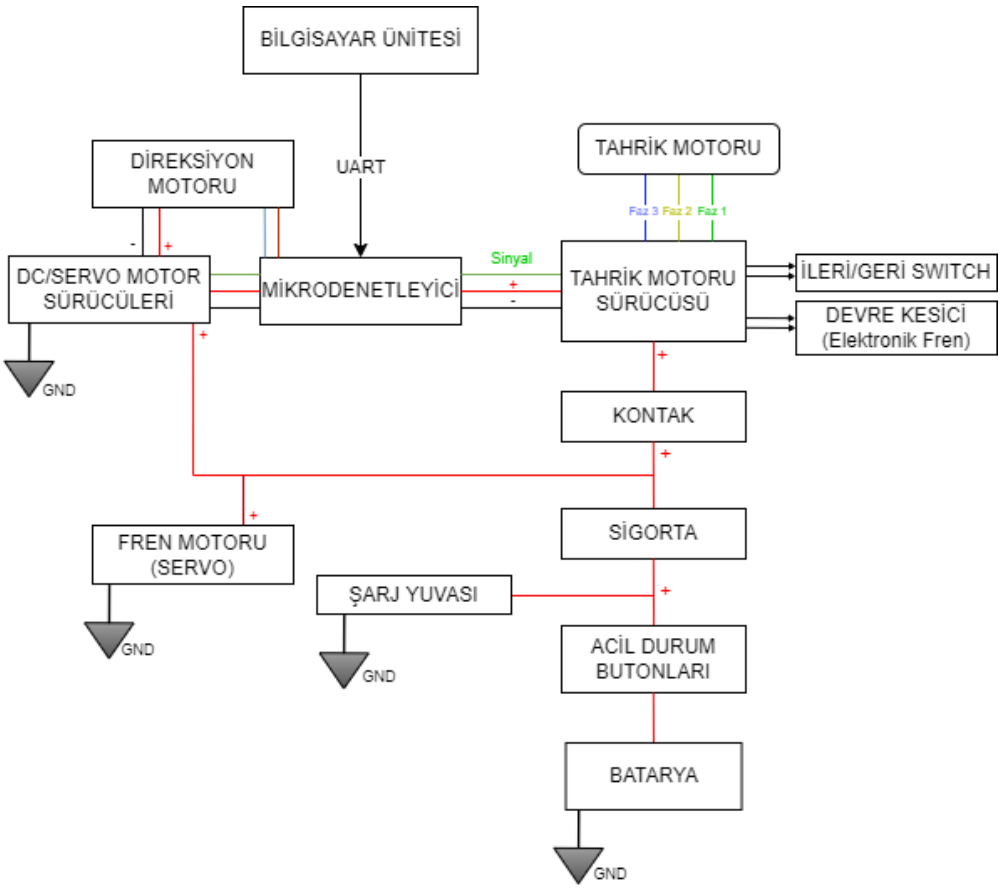
3. Araç Mekanik Özellikleri

Bu bölümde aracın mekaniksel yapısı ele alınmıştır. Aracın dış şasesi elektronik direksiyon, elektronik fren ve elektronik gaz pedalından bahsedilmektedir. Aracın cad çizimleri ve render alınmış halleri de bu başlık altında ele alınmıştır.

3.1. Elektronik Direksiyon (Steer-by-Wire)

Yarışma için tasarladığımız aracın istenilen görevleri (şeritte hareket etme, trafik ışık ve işaretlerine göre çeşitli eylemleri gerçekleştirme vb.) yerine getirebilmesi için elektronik altyapı oluşturuldu. Bu altyapı kullanılarak gaz, fren ve direksiyon sistemlerinin bilgisayar üzerinden kontrolü sağlandı. Aracımızın kablolu sürüşe uygun hale getirmek için izlediğimiz adımları bu başlık altında inceleyeceğiz.

Aracın direksiyonunun döndürülmesi, direksiyon miline dişli ve kasnak üzerinden bağlanan step motorun, step motor sürücüsü üzerinden sürülmesi ile sağlanmaktadır. Düşük seviye kontrolü ise Arduino Uno ile sağlanmaktadır. Şekil 2'te görüldüğü üzere DC Servo motor, motor sürücüye üzerinden mikrodenetleyici tarafından kontrol edilmektedir. Gerekli kontrol sinyalleri bilgisayar ünitesi tarafından gönderilmektedir.



Şekil 2. Motor Sürücü, Motor ve Akü için Elektrik Şeması

3.2. Elektronik Fren (Brake-by-Wire)

Elektronik fren (brake-by-wire), fren sisteminde hidrolik ve mekanik sistemler gibi geleneksel bileşenler yerine elektronik sensör ve aktüatörlerin kullanıldığı bir teknolojidir. Geleneksel fren sistemleriyle birlikte kullanılabileceği gibi bağımsız olarak dizayn edilebilir.

Proje kapsamında kullanılan aracın fren sistemi, fren kolunun sıkılması ile aracın ön tekerlerinde yer alan disklere basınç uygulayarak aracın durmasını veya yavaşlamasını sağlayan hidrolik fren sistemine sahiptir. Bu sisteme ek olarak fren kolu sıkılmaya başladığında motora giden elektrik gücünü kesen bir bağlantı da bulunmaktadır. Fren koluna bağlı olan bu tetikleme, bir düğme anahtar gibi çalışmaktadır. Fren kolunun sıkılması durumunda anahtar kısa devre olarak motor sürücüsüne lojik 1 sinyali gönderilmesiyle motor sürücü motora giden gücü kesmektedir.

Araç üzerinde mevcut bulunan fren sistemini, elektronik fren sistemi haline getirmek için bazı elektronik ekipmanlar fren sistemine eklenmiştir. Bu ekipmanlar; araca konforlu ve güvenli fren yaptıracak balata baskı kuvvetini sağlamak için gereken fren kolunun sıkma derecesini hesaplayacak bir bilgisayar, hesaplanan sıkma değerini fren kolu üzerinde gerçeklemeyi kontrol edecek bir mikrodenetleyici ve fren kolunu sıkma işlemini gerçekleştirecek servo motor ve servo motor sürücüsüdür.

Elektronik fren sisteminin gerçeklenmesinde bilgisayar olarak proje genelinde halihazırda kullanılan Jetson Xavier kullanılmıştır. Sisteme kolay entegre edilip kontrolü gerçekleştirebilecek yeterlilikte olduğundan mikrodenetleyici olarak düşük bütçeli Arduino Uno tercih edilmiştir. Fren kolunu tamamen sıkıştırmak için gereken tork 8 kg.cm olarak ölçülmüştür. Bu nedenle bu sıkıştırmanın rahat yapılabilmesi için 25kg/cm torka sahip LD-20MG Servo Motor kullanılmasına karar verilmiştir. Servo motorun düşük seviye kontrolü Adafruit 16 kanal 12 bit PWM çıkışlı kart üzerinden Arduino Uno ile sağlanmıştır.

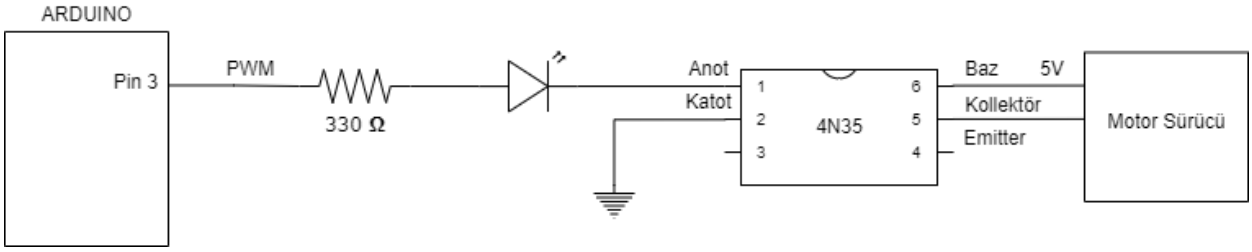
Bu sistemin oluşturulabilmesi için belirlenen servo motor, motora bağlı bir makara ve yeterli uzunlukta çelik tel yardımıyla fren kolunu sıkabilecek bir mekanizma kurulmuştur. Mekanizmada, fren kolunun bağlı olduğu gidona bir servo motor yerleştirilmiş ve servo motorun miline bir makara takılmıştır. Fren koluna ve makaraya bir çelik tel gergin şekilde bağlanmıştır. Servo motorun farklı yönlere dönüşü ile fren kolu istenen değerlerde sıkılıp gevşetilebilmektedir. Bu sayede araç için gerekli fren torku sağlanabilmektedir.

3.3. Elektronik Gaz (Pedal-by-Wire)

Elektronik gaz pedalı için kullanılan malzemeler, Arduino UNO, 4N35 optokuplör, led, motor sürücü ve motordan oluşmaktadır. Arduinonun pin 3'ünden analog sinyal (PWM) alınmaktadır. Arduinodan gelen analog sinyale göre motorun hızı belirlenmektedir. Araç motoru, arka tekerleklere bağlıdır. Aracın elektrik motoru üzerinde bulunan sürücüsü vasıtası ile PWM sinyali kullanılarak kontrol edilmektedir. Araçta ayrıca iki kademe mekanik tork dönüştürücü dişli sistemi bulunmaktadır. Bu sayede araç yüksek ve düşük tork ayarlarında kullanılabilir.

Optokuplör led ile beraber kullanılır. Kullanım amacı düşük seviyeli voltajla yüksek seviyeli voltaj değerlerini kontrol etmektir. Bir nevi elektronik cihazları korumak için yalıtım olarak düşünülebilir.

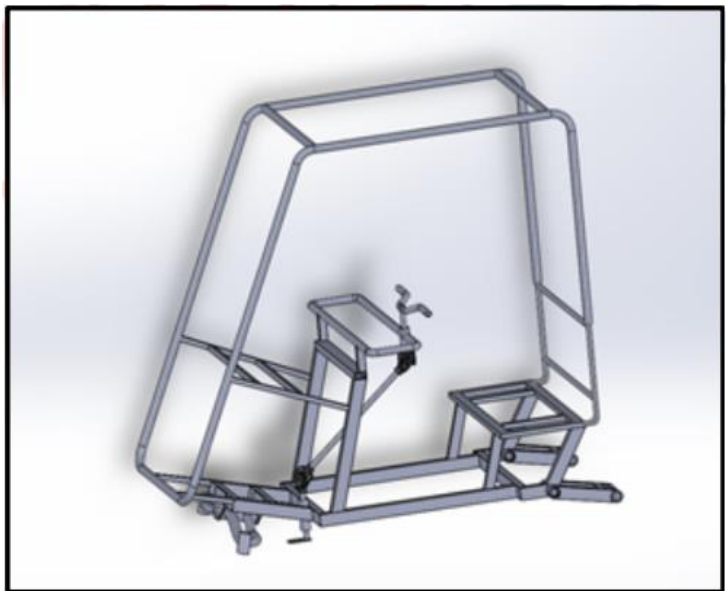
Arduinonun 3 numaralı PWM pinin çıkışı 330 ohm'luk dirence bağlıdır. Direncin diğer ucu ise lede bağlıdır. Led yandığı zaman optokuplör aktif olmaktadır. Bu sayede motor sürücüye gerekli voltaj değerleri gönderilebilir. Motor sürücü ise motoru sürmektedir. Bu sayede motor çalışır. Yukarıda anlatılan devre Şekil 3'te gözükmektedir.



Şekil.3 Elektronik gaz pedalı devresi

3.4. Araç Mekaniği

Aracın şasesi şekil 4'te görüldüğü üzere tasarlanmış genel olarak çelik iskelet üzerine inşa edilmiştir.



Şekil 4. Aracın Şase Kısmının Solidworks Modeli

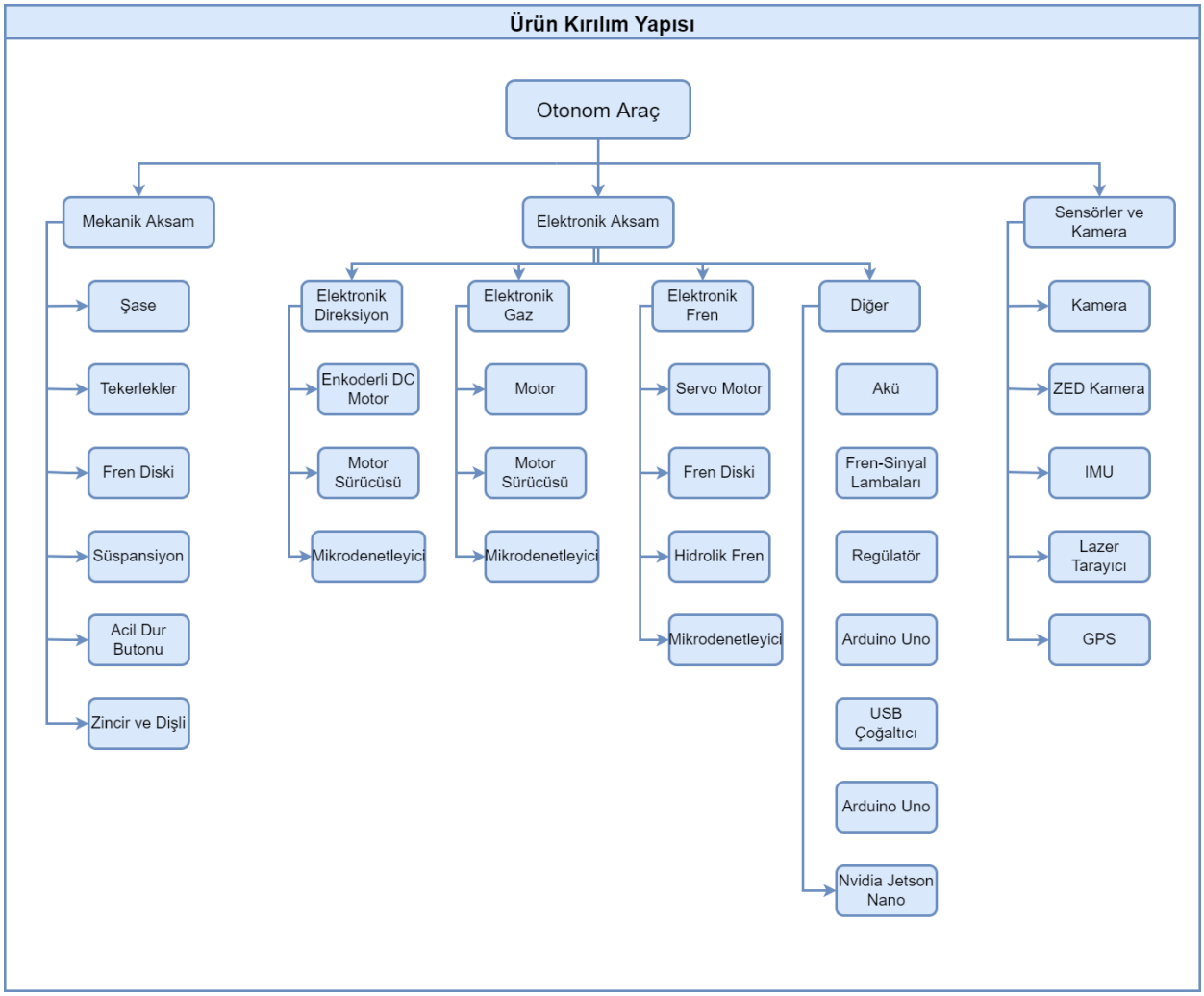
Araç sürüş konforunu arttırmak için Şekil 5'te görüldüğü üzere süspansiyon sistemi kullanılmıştır.

Şekil 5. Aracın süspansiyon kısmının görüntüsü

Aracın sürüşünü gerçekleştirebilmesi için tekerlekler uygun boyutta seçilmiştir.

Şekil 6. Aracın tekerlek kısmının görüntüsü

Ürün kırılım yapısı Şekil 7'de görülmektedir.



Şekil 7. Sistemin Ürün Kırılım Yapısı

3.5. Alt Sistemler ve Temel Bileşenler Arası Fiziksel Arayüzler

Yarışmada istenen görevleri yerine getirecek şekilde tasarlanan aracın alt sistemleri ve bu sistemlerde kullanılan bileşenler Tablo 5'de görülmektedir. Bölüm kapsamında araç üzerindeki alt sistemler ve temel bileşenler arasındaki fiziksel arayüzler sunulmaktadır.

Tablo 5. Alt sistemler

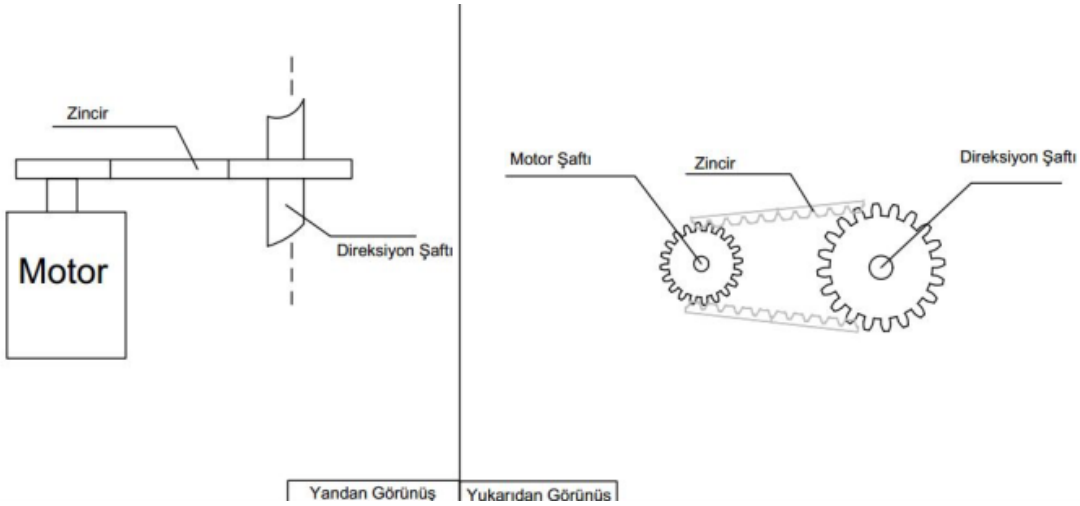
ALT SİSTEMLER	MALZEME LİSTESİ	MALZEME BİLGİLERİ
Elektronik Fren Sistemi	Servo Motor - Motor Sürücü	2 N.m Torka sahip servo motor (LD-20MG Servo Motor) ve motoru sürmeye uygun motor sürücü (Adafruit PCA9685 Servo Motor Sürücü)
	Sıkma Teli	Sarmal 0.3 mm Çelik Halat
	Mikrodenetleyici	Arduino Mega 2560 R3
Elektronik Direksiyon Sistemi	Enkoderli DC motor	KORMAS 24V, 190W, ø67, Enkoderli 1/30 Redüktörlü DC Motor
	DC Motor sürücü	KORMAS 24V Elektronik Devir Ayarlayıcı KORMAS 24V PWM Modülü
	Mikrodenetleyici	Arduino Mega 2560 R3
	Dişli seti - zinciri	18T ve 11T Ruble metal alaşım dişli ve dişli zinciri
Elektronik Gaz Sistemi	Mikrodenetleyici	Arduino Mega 2560 R3
	Motor Sürücü	KUNRAY 2000W - 60V - 35A Motor Sürücü
	Motor	KUNRAY 2000W Electric Brushless DC Motor

3.5.1. Step Motor ile Direksiyon Kontrolü

Projede üzerinde çalışılan aracın yön kontrolü bir gidon ile yapılmaktadır. Gidon Şekil 8'de görülen metal silindirik bir profile bağlanmıştır. Gidonun çevrilmesi ile silindirik profil dönmektedir. Silindirin dönmesi sonucunda mafsallar ve diğer mekanik parçalar ile sol ön tekerleğin dönmesi sağlamaktadır. Sol ön tekerlek ile sağ ön tekerlek arasına metal bir profil bağlıdır. Bu sayede sol teker döndüğünde sağ tekerlek de dönmektedir.

Şekil 8. Araç gidonuna bağlı metal silindir ve teker bağlantıları.

Tekerlekler eksen üzerinde 40 derece sola ve 40 derece sağa dönebilmektedir. Bu dönüşler için gidonun eksende 40 derece sola ve 40 derece sağa döndürülmesi gerekmektedir.



Şekil 9. Direksiyon Motoru - Direksiyon Şaftı Bağlantısı

Direksiyonun kontrol edilebilmesi için Şekil 9'da görülen gidonun bağlı olduğu silindirik metal profili döndürecek bir motor mekanizması tasarlanmıştır. Gidonun takılı olduğu metal profili araç dururken sağa ve sola döndürmek için gereken maksimum tork 24 kg.cm olarak ölçülmüştür. Bu nedenle tekerlerin rahat dönebilmesi ve dönüşün kontrolü için enkoder içeren 30 kg.cm torklu DC motor kullanılmıştır (KORMAS 24V, 190W, ø67, Enkoderli 1/30 Redüktörlü DC Motor).

3.5.2. Gaz Kolu ve Motor Sürücü ile DC Motor Kontrolü

Projede üzerinde çalışılan araç üç farklı güç modu ile çalışmaktadır. Bu güç modları; minimum güç, orta güç ve maksimum güç aralıklarıdır. Herhangi bir güç modunda, gaz kontrolü gaz kolu ile gerçekleştirilebilmektedir. Gaz kolu, kablo ile aracın arka kısmında bulunan motor sürücüye bağlanmaktadır. Bununla birlikte gaz koluna bağlı 5 farklı kablo güç modu kontrolünü sağlamaktadır. Güç modu kontrolcüsünün ve gaz kolunun görünümü Şekil 10'daki gibidir. Güç modu butonuna basıldığında kablolardan gönderilen lojik değer 1-0 olarak değişmekte ve bu sayede güç modu motor sürücüsü tarafından algılanmaktadır. Gaz kolu bir potansiyometre işlevinde çalışmaktadır. Motor sürücüsünden gaz koluna 5V besleme gerilimi gelmektedir. Gaz kolu çevrildiğinde çıkıştaki voltaj değeri 0V ile 4.6V arasında değişmektedir. Gaz kolunun çıkış

tekrar motor sürücüsüne bağlanmaktadır. Voltaj değişimine göre motor sürücüsü aracın hareketini sağlayan, arka tekerleklere bağlı motora gönderilen voltajı değiştirerek aracın hız kontrolünü sağlamaktadır.

Şekil 10. Güç modu kontrolcüsü ve gaz kolu

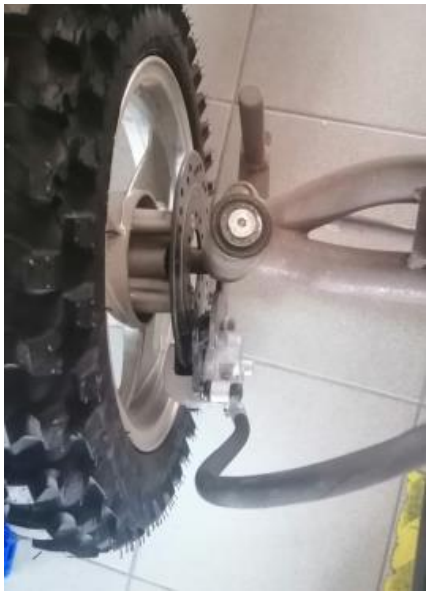
Gaz kontrolünün sağlamak için gaz kolu ile motor sürücü arasındaki sistem senaryosunu taklit eden bir mikrodenetleyici mekanizması tasarlanmıştır. Araç dururken gaz kolu çıkışından motor sürücüsüne 0 V göider. Gaz kolu maksimum derecede çevrildiğinde ise 4,6 V gönderilmiş olur. Tasarlanan sistemde mikrodenetleyicinin analog pinlerinden motor sürücüsünün gaz kolu pinine bağlantı bulunmaktadır. Bilgisayar ünitesinden mikrodenetleyiciye gelen hız verilerine göre motor sürücüsüne bağlı mikrodenetleyicinin analog pinlerinden motor sürücüsüne 0-4,6 V aralığında bir voltaj değeri verilerek araç hızı kontrol edilmektedir.

3.5.3. Fren Kolu ve Servo Motor ile Fren Kontrolü

Projede üzerinde çalışılan aracın fren mekanizması, fren koluna bağlı hidromekanik - elektronik tabanlı bir fren mekanizmasıdır. Fren kontrolünü sağlamak için fren kolunun sıkılmasını sağlayacak bir servo motor mekanizması geliştirilmiştir. Fren kolunun servo motor eklenerek otonom kontrol edilebildiği hali Şekil 11'deki gibidir.

Şekil 11. Servo motor ile kontrol edilen fren kolu

Fren kolunun sıkılması ile Şekil 12'de görülen aracın ön tekerleklerine bağlı disklere basınç uygulanarak aracın yavaşlaması veya durması sağlanmaktadır.

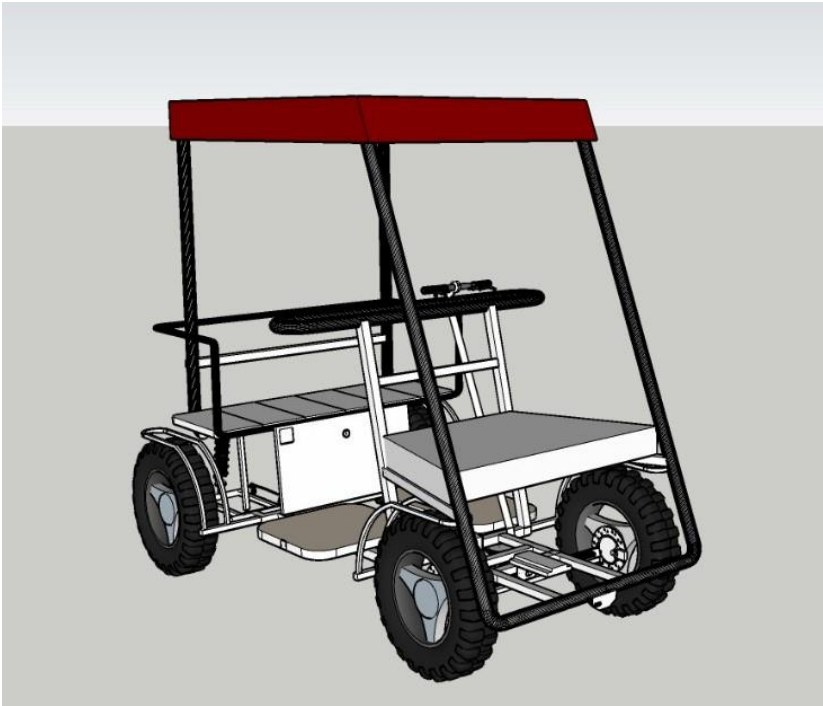


Şekil 12. Tekerleklere bağlı diskler ve fren balataları

Aynı zamanda bu fren kolundan araç hareketini sağlayan motorun sürücüsüne bir bağlantı vardır. Bu bağlantı sayesinde fren kolu sıkılmaya başlandığında motora giden elektrik kesilmektedir. Fren koluna bağlı olan bu tetik, bir düğme anahtarı gibi çalışmaktadır. Anahtar, motor sürücüsü üzerinden lojik 1 (5V) sinyali ile beslenmekte ve çıkışı motor sürücüsüne geri dönmektedir. Fren kolunun sıkılması halinde anahtar kısa devre olarak motor sürücüsüne lojik 1 sinyali gönderilmesiyle motor sürücüsü motora giden gücü kesmektedir.

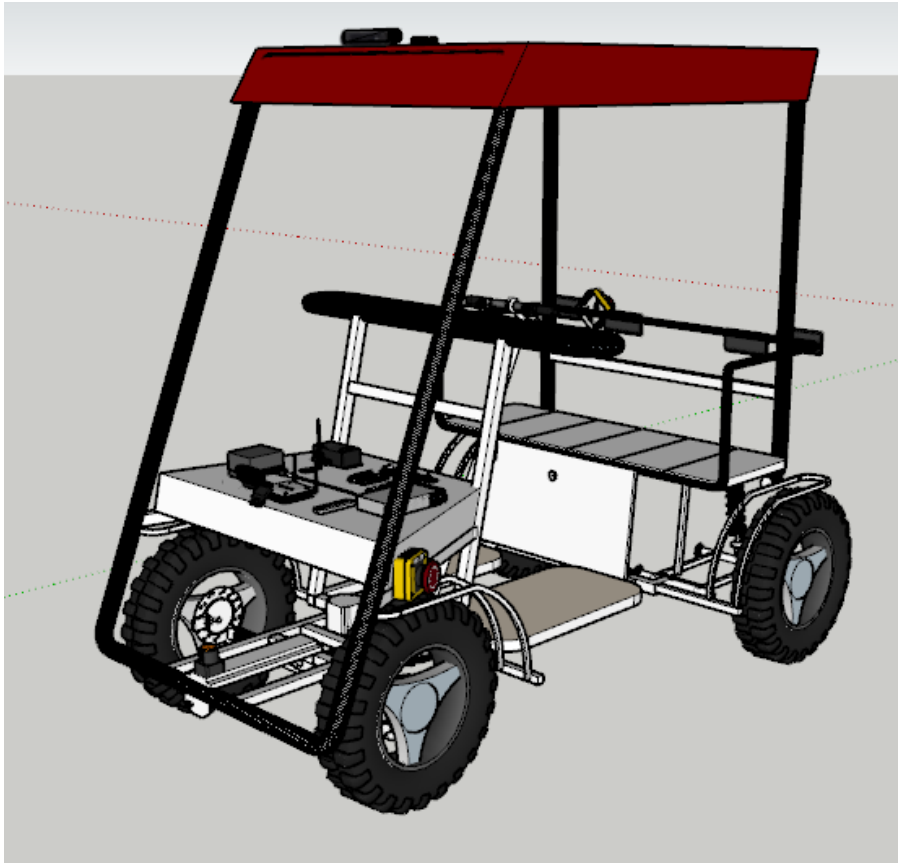
3.5.4. Araç Mekanik Tasarımı

Aracın kasa kısmı, direksiyon sistemi, süspansiyon sistemi, tekerlekleri, aküsü, elektronik freni, motorları, sigortası bnb bisiklet firması tarafından hibe edilmiştir.



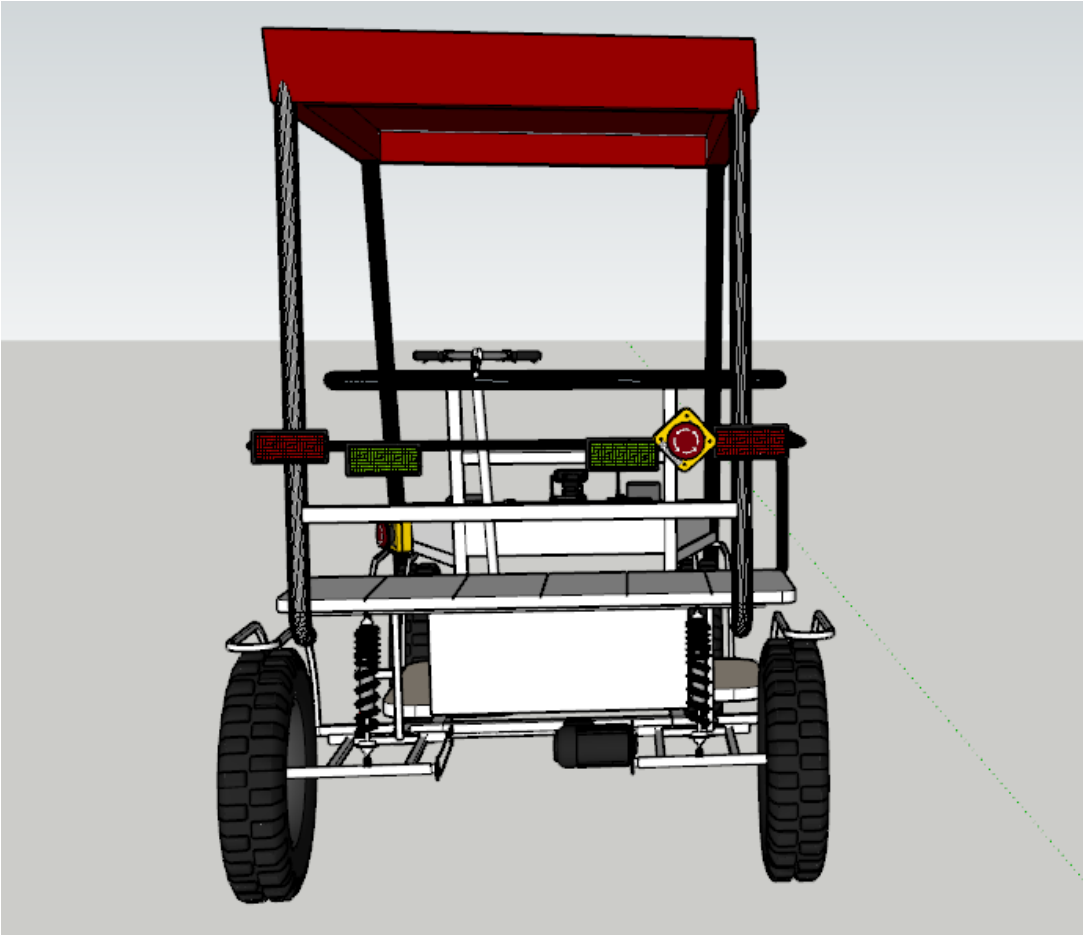
Şekil 13. Aracın genel CAD tasarımı

Aracın tasarımı Sketchup programında gerçekleştirilmiştir. Şekil 13'te görüldüğü gibi, tasarımın ilk halinde herhangi bir sensör ve ekipman bulunmamaktadır. İlk çizimde aracın mekaniksel yapısı dikkate alınmıştır. Bu sayede aracın tekerlekleri, direksiyonu zincirleri, motoru, akü kutusu, elektronik ekipmanların konulacağı yer çizilmiştir.



Şekil 14. Aracın Sensörler ve Elektronik Ekipmanlar Eklenmiş CAD Tasarımı

Aracın mekaniksel tasarımı bittikten sonra, gene Sketchup programı kullanılarak, mevcut aracın üzerindeki elektronik komponentler, sensörler ve acil durdurma butonları birebir olarak yerleştirilmiştir. Şekil 14'te yerleştirilen parçalar gözükmektedir. GPS ve ZED kamera aracın üst tente kısmına yerleştirilirken, lazer tarayıcı alt kasanın önüne konmuştur. Ayrıca şerit takibi için kullanılacak kamerada elektronik komponentlerin bulunduğu tablaya yerleştirilmiştir. Elektronik komponent olarak, wifi cihazı, regülatör, Arduinolar, Nvidia Jetson Nano, usb çoğaltıcı gibi araçta hali hazırda kullanılan tüm ekipmanlar eklenmiştir.



Şekil 15. Aracın Sensörler ve Elektronik Ekipmanlar Eklenmiş Tasarımı (Arka)

Şekil 15'te araç tasarımının arka kısımdan görüntüsü gözükmektedir. Fren ve sinyal lambalarının eklendiği gözükmektedir. Ayrıca acil durdurma butonu da aracın arka kısmında bulunmaktadır. Aynı şekilde tasarımda da gösterilmiştir.

Şekil 16. Aracın tasarımının Lumion programından render alınmış hali

Araç Sketchup programında çizildikten sonra, render alma işlemi uygulanmıştır. Bunun için Lumion programı kullanılmıştır. Şekil 16'da aracın render alınmış hali gözükmektedir. Otonom hareketini temsilen üzerinde insanların oturduğu da gösterilmiştir.

Şekil 17. Lumion programından render alınan aracın önden görünümü

Araç render alındıktan sonra önden görünümü Şekil 17'de gösterilmiştir.

Şekil 18. Lumion programından render alınan aracın ekipmanlara odaklanmış görünümü

Render alınmış araç üzerindeki sensörler ve elektronik bileşenlerin detaylı gösterimi Şekil 18'de ifade edilmiştir. Bu sayede fiziksel araç üzerindeki elektronik bileşenler aracın render alınmış

halinde de görülebilir.

Şekil 19. Lumion programından render alınan aracın arkadan görünümü

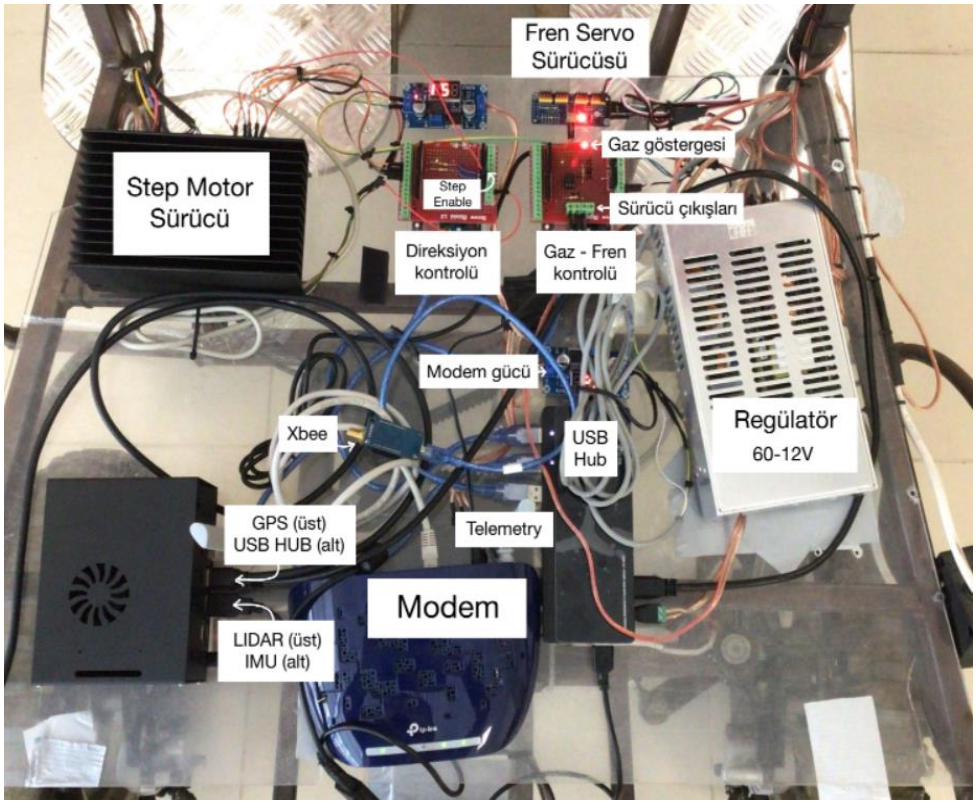
Aracın render alındıktan sonra arkasından görünümü Şekil 19'da gösterilmiştir. Burada acil durdurma butonu, sinyal lambası, fren lambası ve aracın mekaniksel yapısı gözükmektedir.

4. Donanım Mimarisi

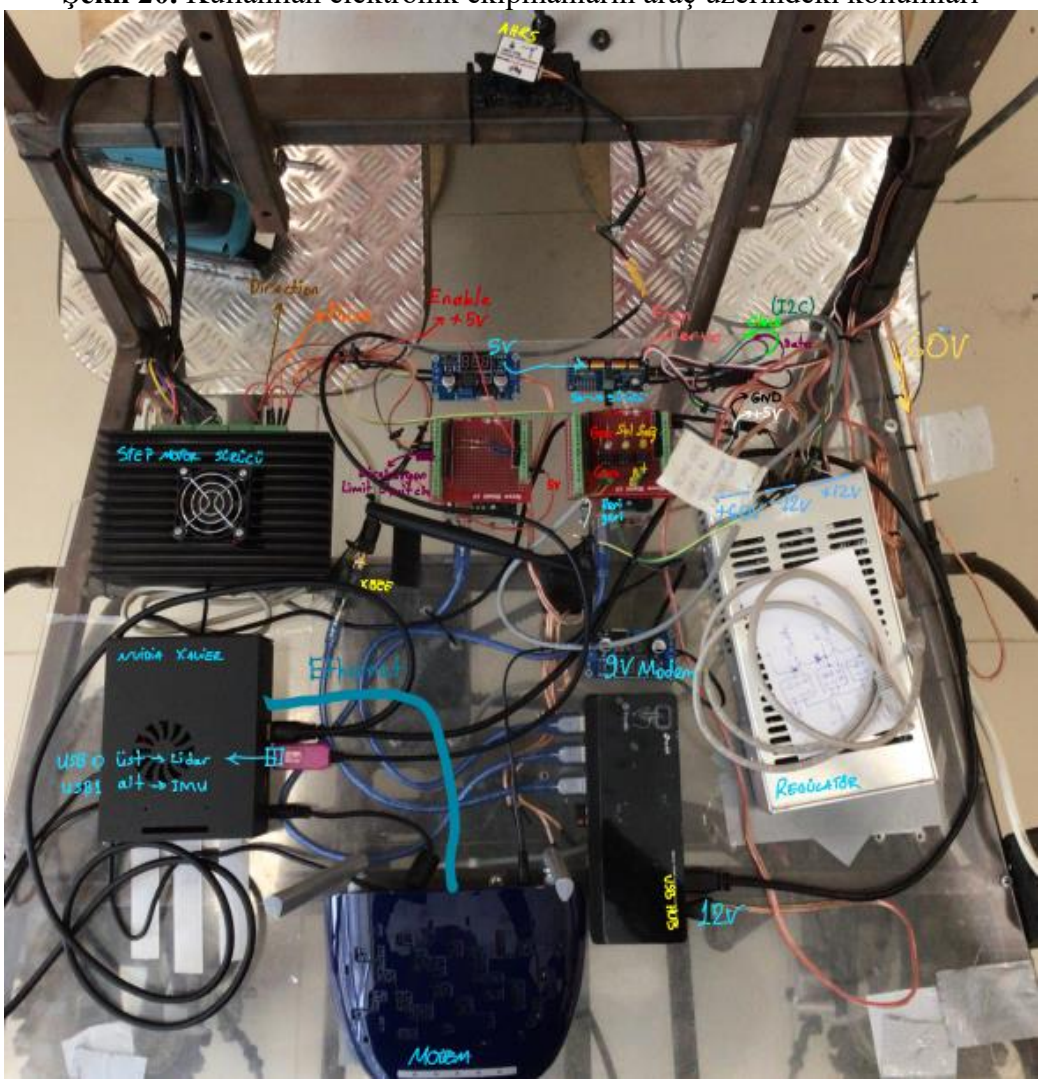
Bu bölümde aracın kontrolcüsü, yapay zeka bilgisayarı, sensörleri, kablosuz haberleşme sistemi, batarya yönetim sistemi ve donanımsal güvenlik önlemleri gibi donanım bileşenlerinin detayları hakkında bilgi aşağıda verilmiştir.

Kullanılan tüm sensörler ve sistemler için bağlantı şeması, elektriksel, kablo bağlantıları gösteren Harness Diyagramını bu bölümde gösterilmiştir.

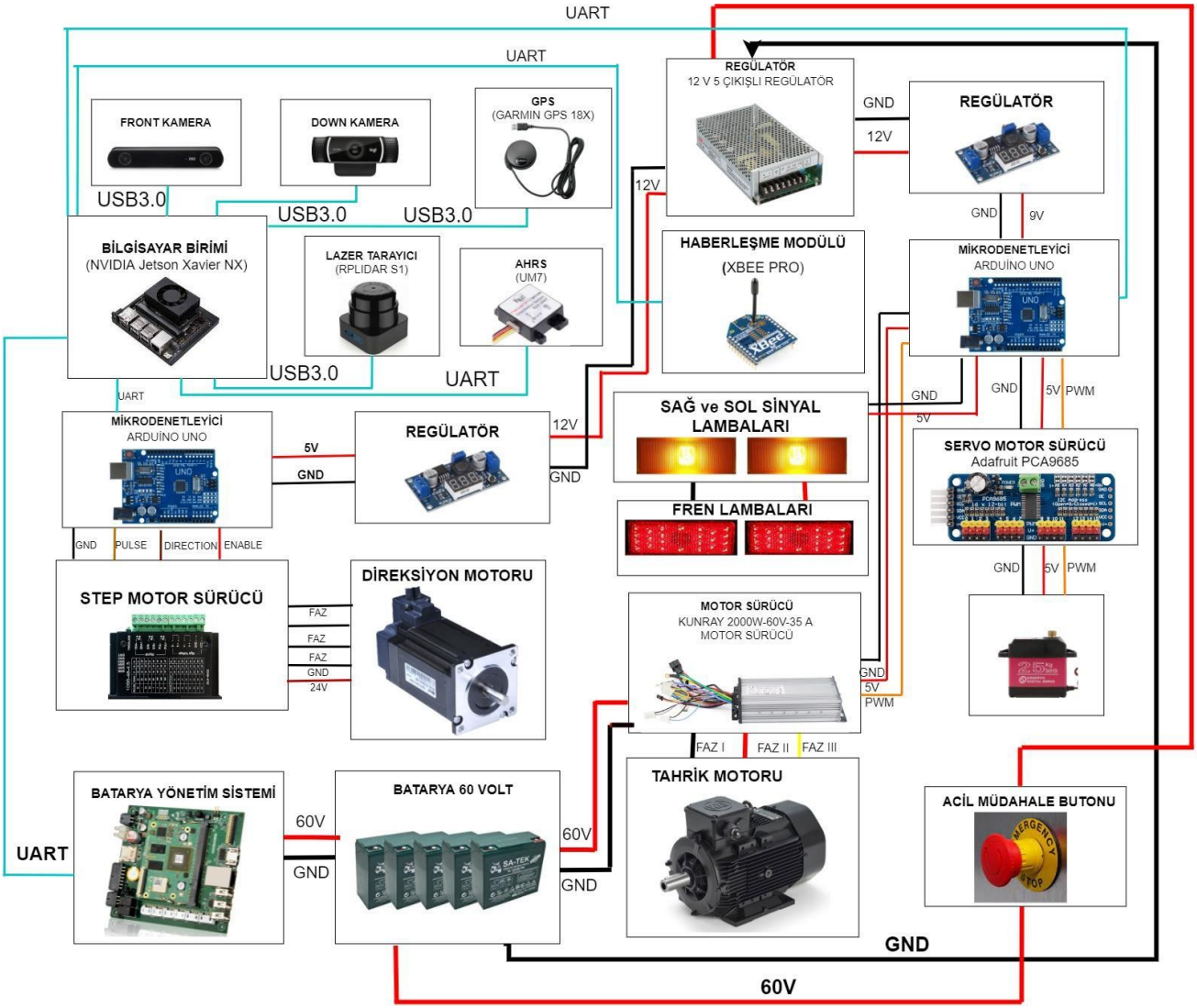
Harness Diyagramı, araçta kullanılan donanım ve bileşenler temsil edecek şekilde soyut veya basitleştirilmiş şekilde şekiller bu bölümde verilmiştir.



Şekil 20. Kullanılan elektronik ekipmanların araç üzerindeki konumları



Şekil 21. Kullanılan elektronik ekipmanların araç üzerindeki konumları



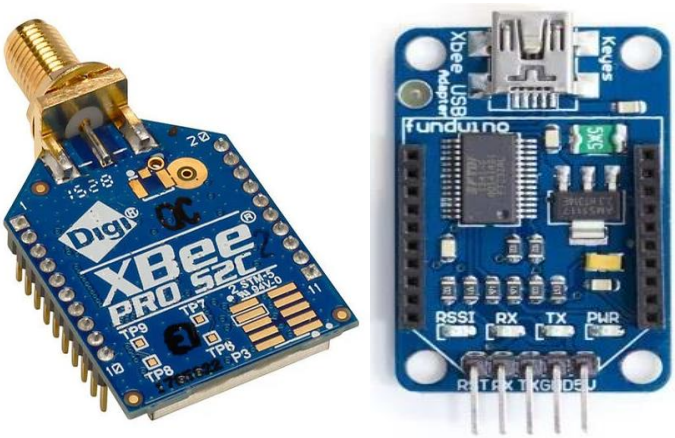
Şekil 22. Sistemin Harness Diyagramı

4.1. Limit Anahtarları

Aracın ön tekerleklerinin dönüş açısı yaklaşık ±40 derece ile sınırlıdır. Bu durumda, direksiyonu kontrol eden step motor sürücüsünü bu sınırlar içerisinde çalıştırmak gerektiğinden, sağ ve sol olmak üzere iki adet limit anahtarı aracın ön aksına eklenmiştir. Bu sayede direksiyon kontrolü sırasında, direksiyonun mekanik limitinden dolayı meydana gelebilecek arızaların önüne geçilmiştir.

4.2. Xbee

2 adet XBee Pro S2C modülü, XBee Explorer USB kartıyla birlikte telemetri haberleşmesinde kullanılmıştır. Bu haberleşme 9600Kbps hızındadır. Aracın hız, konum gibi bilgileri bu hat üzerinden arayüzde gösterilmek üzere aktarılmaktadır. Ayrıca aracın başlatma ve acil durdurma komutları da bu hat üzerinden iletilir.



Şekil 23. Sağda Xbee Pro S2C ve solda USB ile bağlanmayı sağlayan XBee Explorer USB

4.3. Modem

Bir adet modem araca eklenmiştir. Bu modemin oluşturduğu Wifi ağı sayesinde, araca kablosuz şekilde bağlanılarak, terminal aracılığıyla gerekli komutları çalıştırılabilmekte ve dosya aktarımı sağlanmaktadır.

Şekil 24. TP-Link TD-W9960 300Mbps

4.4. Batarya Yönetim sistemi

Bataryaların kontrol edilmesi ve yönetilebilmesi amacıyla kullanılan batarya yönetim sisteminden bahsedilmektedir. Batarya yönetim sistemi temel olarak dört adet işlev gerçekleştirmektedir. Bunlar, gerilim ve sıcaklık ölçme, doluluk oranı hesaplama, koruma ve güvenlik önlemleridir.

4.4.1. Gerilim, Sıcaklık Ölçme

Aracın gerilimi, üzerindeki Arduino Uno ile, telemetri üzerinden takip edilebilmektedir.

4.4.2 Doluluk Oranı Hesaplama

Aracın akü gerilimi ölçülüp telemetri aracılığıyla arayüzden izlenebilmektedir. Bu sayede ne kadar çalışma süresi kaldığı kestirilebilir.

4.4.3. Koruma

Araçta iki adet acil durum butonu ve bir adet sigorta bulunmaktadır. Bu şekilde yüksek akım çekme veya aracın kontrolden çıkması durumları için önlem alınmıştır.

4.4.4. Güvenlik Önlemleri

Aracın kameralar ile bağlantılarının kopması durumunda, aracın kontrol yazılımları kendilerini kapatacak ve aracın gazını kesecek şekilde ayarlanmıştır.

4.5. Yapay Zeka Bilgisayarı

Aracımızda otonom sürüş için kullanılacak olan bilgisayarın, nesne tespiti ve şerit takibi algoritmalarını doğru şekilde çalıştırabilmesi, gelen veriyi doğru işleyip bunu gerçek zamanlı olarak yürütebilmesi gerekmektedir. Bu doğrultuda aracımızda nvidia jetson xavier NX bilgisayarı kullanılmaya karar verilmiştir. Sağladığı 21 TOPS (saniye başına teorik işlem) yapay zeka performansı ile modern yapay sinir ağlarını paralel olarak çalıştırabilmektedir. CPU, GPU özellikleri ve bulundurduğu derin öğrenme hızlandırıcı motoru sayesinde yüksek çözünürlüklü sensörlerden elde edilen verileri eş zamanlı olarak işleyebilmektedir. Sahip olduğu güç performansı sayesinde otonom aracımızın çevre birimlerine ve sönserlerine daha fazla güç ayırmamızı sağlamaktadır. Nvidia Jetson Xavier NX teknik özellikleri şekil 25'te verilmiştir.

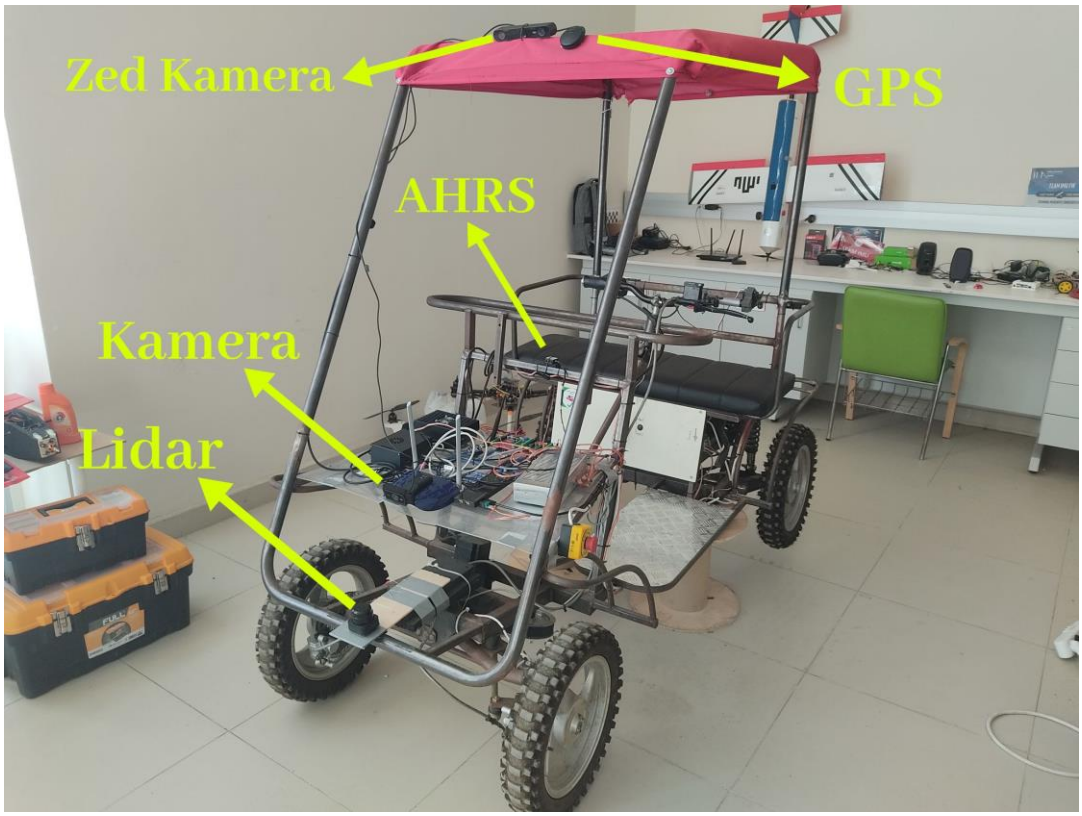
Şekil 25. Nvidia Jetson Xavier NX Bilgisayarı. Yapay Zekanın ve bütün sürüş programlarının çalıştığı ana bilgisayardır.[1]

Yapay Zeka Performansı	21 TOPS
GPU	48 Tensör Çekirdeğiyle Donatılmış 384 Çekirdekli NVIDIA Volta™ GPU
CPU	6 Çekirdekli NVIDIA Carmel ARM® v8.2 64 bit CPU 6 MB L2 + 4 MB L3
Bellek	16 GB 128 bit LPDDR4x 59,7 GB/sn
Depolama	16 GB eMMC 5.1
Güç	10 W | 15 W | 20 W
PCIe	1 x1 (PCIe Gen3) + 1 x4 (PCIe Gen4), toplam 144 GT/sn*
CSI Kamera	6 kameraya kadar (sanal kanallar aracılığıyla 24) 14 şerit (3x4 veya 6x2 veya 3x4 + 1x2 veya 5x2 + 1x4) MIPI CSI-2 D-PHY 1.2 (30 Gb/sn'ye kadar)
Video Enkodlama	2x 4K60 | 4x 4K30 | 10x 1080p60 | 22x 1080p30 (H.265) 2x 4K60 | 4x 4K30 | 10x 1080p60 | 20x 1080p30 (H.264)
Ekran	2 çoklu mod DP 1.4/eDP 1.4/HDMI 2.0
DL Hızlandırıcı	2 adet NVDLA
Görüş Hızlandırıcı	2 adet PVA
Ağ	10/100/1000 BASE-T Ethernet

Şekil 26. Nvidia Jetson Xavier NX Teknik Özellikleri[2]

4.6. Sensörler

Aracın otonom olarak hareket kabiliyeti kazanması için çevresini algılaması ve anlamlandırması gerekmektedir. Çevre algılama ve anlamlandırması yapabilmek için sensörlerden gelen veriler kullanılır. Bu veriler ham olabileceği gibi işlenmiş verilerde olabilir. Yarışma kapsamında oluşturulan otonom araçta kullanılan sensörler Şekil 27'de görüldüğü üzere kamera, lazer tarayıcı, AHRS ve GPS'tir.



Şekil 27. Sensörlerin Araç Üzerindeki Yerleşimi

4.6.1. Kamera

Otonom aracın çevresini algılayabilmesi, insanın görmesi kadar önemlidir. Çevre algılamayı sağlayan sensörlerin en temeli kameradır. Kamera sayesinde çevresindeki nesneleri algılayabilecektir. Kameralar sayesinde trafik lambaları, trafik işaretleri, şeritler, bariyerler gibi trafik düzenini sağlayan işaretler tespit edilebilir. Bu sayede gerekli algoritmalar çalıştırılarak çevre algılama ve anlamlandırma gerçekleştirilmiş olur.

Yarışmanın koşulları gereği, araçlar dış ortamda sürüleceği için kamera seçim özelliklerine dikkat edilmiştir. Kameranın verimli çalışma sıcaklık aralığı -10 C ile +125 C arasındadır. Dış ortamda yeterli düzeyde bir sıcaklık aralığıdır. Kameranın diyafram açıklığı f/1.8'dir. Bu sayede 120 derecelik görüş alanı elde edilir. Kameranın boyutu 175 x 30 x 33 mm, ağırlığı ise 150 gramdır. Araç üstüne yerleştirilmek için uygundur. Ubuntu işletim sistemini desteklemekte, ROS ve Gazebo paketleri bulunmaktadır. Kameranın içerisinde dahili olarak, ivmeölçer, jiroskop, barometre, manyometre ve sıcaklık sensörü bulunmaktadır. İçerisinde bulundurduğu sensörler sayesinde basit düzeyde konum takibi ve hız kontrolü yapabilmektedir. Bu da aracın hareketinde ciddi bir sapma olduğunda test amaçlı kullanılabilir. Ayrıca kameranın en önemli özelliği stereo

olmasıdırı. Kamera derinlik algılayabilmektedir. Araç kapsamında maddi yetersizlikten dolayı LIDAR cihazı bulunmadığı için derinlik algılamadan faydalanılmaktadır. Derinlik algılama sayesinde tabelalar da daha verimli şekilde algılanabilmektedir. ZED 2 kamera ile 20 metreye kadar 3D görüntü elde edilebilirken, 40 metreye kadar 2D görüntü elde edilebilir. Kameranın seçiminde mesafe aralığının uzunluğu da etkili olmaktadır. Ayrıca [3].

Şekil 28. ZED 2 Kamera [4]

Otonom araçta bir adet ZED 2 kamerası kullanılacaktır. Kamera tabelaların algılanması için karşıya doğru bakacak şekilde konumlandırılacaktır.

Otonom araç, algılanan çevre verilerinden yola çıkarak sürüşünü yapmaktadır. Çevre algılama ne kadar doğru algılanırsa sürüş o kadar sağlıklı gerçekleşmektedir. Şerit takibi için kullanacağımız kamera ise Logitech C920 PRO HD WEB olarak seçilmiştir. Seçilen kamera modülü şekil 29'da gözükmektedir.

Kamera 1080p/30 FPS - 720p/30 FPS maksimum çözünürlük sağlamakla birlikte diyagonal görüş alanı (dFoV): 78°'tir. Loş ortamlardan elde edilen görüntülerde sahip olduğu RightLight 2 teknoloji sayesinde daha aydınlık görüntüler elde etmemizi sağlamaktadır. Görüntülerde net renkler yakalamamızı sağlamakla birlikte otomatik odak özelliği bulunmaktadır. Keskin ve net görüntüler elde etmemize olanak sağladığı için özellikle bu model tercih edilmiştir.

Şekil 29. Logitech C920 PRO HD WEB Kamerası [5]

kamera şeritlerin daha iyi tespit edilmesi için yola bakacak şekilde konumlandırılacaktır. Bu sayede şerit takibi başarılı bir şekilde gerçekleştirilecektir.

4.6.2. Lazer Tarayıcı

Lazer Tarayıcı, otonom aracın çevresini algılamada kullandığı sensörlerden birisidir. Lazer tarayıcı sayesinde, sürüş esnasında otonom aracın karşısına çıkacak olan engeller ve yayalar tespit edilebilir. Ayrıca durak ve park manevralarının gerçekleştirilebilmesi için gerekli olan, direk ile

kendi arasındaki mesafeyi tespit edebilmektedir. Geometrik denklemler kullanılarak, direklerin x, y koordinatları da tespit edilebilir. Bu sayede manevraların düzgün gerçekleşmesi sağlanır.

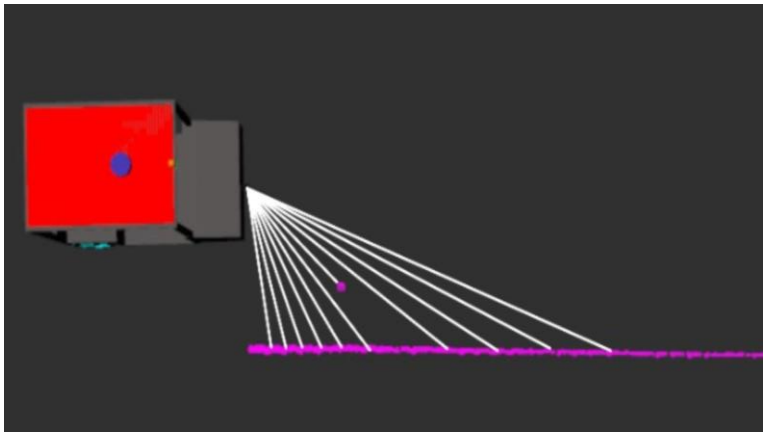
Otonom araçta kullanılması planlanan lazer tarayıcı RPLIDAR S1'dir. RPLIDAR S1 lazer tarayıcı iki boyutludur. Maddi destek sağlandığı durumda 3 boyutlu LIDAR kullanılması planlanmaktadır. RPLIDAR S1'in en önemli özelliği 360 derece tarama yapabilmesidir. RPLIDAR S1'in menzili 40 metredir. Dış ortam için geliştirilmiştir. Lazer tarayıcı kendinden muhafazalıdır. Bu muhafaza toz, su gibi dış etkenlerden korur. Bu sayede dış ortamlarda kullanılabilmektedir. Açısal çözünürlüğü 0.391 derecedir. Yani 360 derecede 920 adet ölçüm almaktadır. Örneklem alma sayısı ise saniyede 9200'dür. Bu da saniyede 10 kez ölçüm alması demektir. Saniyede ölçüm alma sayısı otonom araç için gerçekleştirilen algoritmalara uygundur. Bu RPLIDAR S1'in ROS ve Gazebo paketleri bulunmaktadır. Bu da simülasyon ortamında deneme imkanı sunduğu için RPLIDAR S1'in tercih edilmesi için önemli bir etkendir.

Şekil 30. Lazer tarayıcı - RPLIDAR S1 [6]

Şekil 30'da otonom araçta kullanılan RPLIDAR S1 lazer tarayıcısı gözükmektedir. Şekil 30'da gözüken lazer tarayıcının boyutu, 55.5 x 55.5 x 51 mm, ağırlığı ise 105 g'dir. Otonom aracın üzerine koymak için uygun boyut ve ağırlığa sahiptir.

Yazılan algoritmada lazer tarayıcının görevi levhaları ve bariyerleri algılamaktadır. Levha ve bariyerlerin arasında oluşan geometriden yararlanarak park ve durak manevraları gerçekleştirilmektedir.

Şekil 31, Gazebo 11'de aracın rviz görüntüsüdür. Şekil 31'de beyaz çizgiler sonradan eklenmişlerdir. Beyaz çizgiler uzaklığı temsil etmektedir. Mor noktalar ise Lazer Tarayıcı'nın algıladığı iki boyutlu verilerdir. Bu veriler parkur için, levhalar, bariyerler ve engellerden oluşmaktadır.



Şekil 31. Lazer Tarayıcı Görüntüsü (Mesafe Eklenmiş)

Lazer tarayıcı levhaları tespit ederken, bariyerlerle levha arasında oluşan anlamlı farktan yararlanmaktadır. Şekil 31'de anlamlı fark gözükmektedir. Mesafe verileri sürekli artarken levha ile karşılaştığında mesafede azalış olmaktadır. Levhadan sonra tekrardan mesafenin artışı devam etmektedir. Aracın bulunduğu konuma ve doğrultuya göre anlamlı fark değişmektedir. Analizleri yapılarak, aracın doğrultusuna göre anlamlı mesafe değerleri ortaya çıkarılmıştır.

4.6.3. GPS

Araçta yüksek hassasiyetli GPS olan, Garmin GPS 18x OEM kullanılacaktır. Garmin GPS sensörünün çapı 6 cm'dir. Veri aktarımı USB üzerinden gerçekleşmektedir. Garmin GPS sensörünün ROS paketi bulunmaktadır. GPS sensörünün kullanım amacı, uydular üzerinden GNSS konum bilgisine erişmektir. Aracın konum bilgisine GPS sensörü sayesinde ulaşılacaktır. Garmin GPS'nin ROS paketi bulunmaktadır [7].

Şekil 32. Garmin GPS 18x OEM

4.6.4 AHRS (Attitude and Heading Reference System)

AHRS sensörü IMU (Inertial Measurement Unit) sensörünün gelişmişidir. IMU sensöründe sadece sensör verileri bulunmaktadır. IMU sensörü içerisinde temel olarak iki sensör barındırır. Bunlar jiroskop ve ivmeölçerdir. AHRS sensörü içinde IMU bileşenlerini barındırmakla beraber, bu bileşenlerden gelen verileri işlemektedir. AHRS sensörü kuaterniyonlar, euler açıları ve ya vektörler gibi farklı verileri işleyerek veren, konum ve yön hesaplayan on-board sensördür.

Otonom araçta kullanılan sensör, Şekil 8'de gösterilmiş olan UM7 Orientation Sensordür. UM7 içerisinde, üç eksenli ivmeölçer, hız jiroskopu ve manyometre sensörleri vardır. Standart IMU'lara göre içerisinde manyometre kullanılması tercih sebebidir. Bu sensörlerden gelen verileri Genişletilmiş Kalman Filtresi kullanarak birleştirir [8]. Kalman filtresi kullanılması sebebiyle doğruluk oranı artar. Sensör seçiminde, Genişletilmiş Kalman Filtresi olması önemli ölçüttür. 6 serbestlik derecesine sahiptir. Otonom araç için yeterli düzeydedir. UM7 Orientation Sensör'ün ROS ve Gazebo paketi bulunmaktadır [9]. Bu sayede simülasyon ortamında deneme imkanı sağlar.

Şekil 33. UM7 Orientation Sensor [10]

4.6.5. Sinyal ve Fren Lambaları

Araç üzerinde iki adet sinyal lambası ve iki adet fren lambası bulunmaktadır. İki adet fren lambası birbirine bağlıdır. Fren lambaları beslemeyi doğrudan aküden 12V olarak almaktadır. Frene basıldığında, mandalın altındaki anahtar hem motorun sürücüsünü durdurmakta hem de fren lambalarının yanmasını sağlamaktadır. Bunun sebebi, herhangi bir yazılımsal sıkıntı olsa bile, fren lambalarının güvenlik açısından çalışmasının sağlanmasıdır. Sağ ve sol sinyal lambalarının düşük seviye kontrolleri ise Arduino Uno'da kullanılmak üzere tasarlanan devre (Şekil.3 Pedal-by-wire devresinin benzeri) aracılığıyla kontrol edilmektedir. Güçlerini ise 12V olarak 60V/12V DC/DC dönüştürücüden almaktadır.

Şekil 34. Sağ sinyal lambası

Şekil 35. Sol sinyal lambası

Şekil 36. Sağ, sol ve dörtlü sinyal lambası

4.7. Güvenlik Önlemleri

Proje kapsamında çevresel unsurlar ve araç içi sistemlerin oluşturabileceği birçok farklı tehlike durumu bulunmaktadır. Tehlikeli durumları ortadan kaldırmak için aracın çeşitli güvenlik önlemlerine sahip olması gerekir. Bu nedenle araçta mekanik, elektrik, elektronik ve yazılımsal güvenlik önlemleri alınacaktır. Oluşturulacak araçta bulunacak güvenlik sistemleri bu başlık altında incelenecektir.

4.7.1. Yazılım Güvenlik Önlemleri

Uygulanması planlanan yazılım güvenlik önlemleri şu şekilde sıralanabilir:

● Yabancı bir unsura belirli bir mesafeden fazla yaklaşılması durumunda ani fren işlemi gerçekleştirilecektir.

● Aracın durum kontrolünün uzaktan yapılacağı panelde ani fren yapıp ardından elektriği kesecek acil durum müdahale sistemi bulunacaktır.

● Şerit tespitinin yapılamaması durumunda aracın yavaşlaması / ani fren ile durması sağlanacaktır.

● Sensörlerden veri gelmeme durumunda sistem durdurulacaktır.

● Aracın üzerindeki lazer tarayıcıdan alınan veriler sürekli kontrol edilecektir. Lazer tarayıcı tarafından, araca belirli bir mesafeden fazla yaklaşma tespit edildiği durumda bilgi gönderecektir. Bu bilgi geldiği anda ani fren işlemi gerçekleşecektir.

4.7.2. Elektronik Güvenlik Önlemleri

Uygulanması planlanan elektronik güvenlik önlemleri şu şekilde sıralanabilir:

● Önde, arkada, sağda ve solda elektriği kesecek acil durum butonları bulunacaktır.

● Batarya parametrelerinin ekstrem olması halinde güvenli çalışma sağlayacak bir Batarya Yönetim Sistemi oluşturulacaktır. Bu sistem sayesinde aracın bataryasının güvenli çalışma parametreleri (batarya sıcaklığı, bataryadan çıkan akım, bataryadan çıkan gerilim. vb.) içinde olup olmadığı anlık olarak kontrol edilebilecektir. Eğer ekstrem durumlar olursa, batarya kontrol altına alınacaktır. Araç kendini kapatacaktır.

● Araç 5 tane 12V akü, toplamda 60V gerilim ile çalışmaktadır. Aracın aküsü ile motor sürücü devresi arasında yüksek akımda aracın akülerinin bağlantısını kesmek üzere bir 60 amper bıçak sigorta bulunmaktadır.

4.7.3. Mekanik ve Elektrik Güvenlik Önlemleri

Uygulanması planlanan mekanik ve elektrik güvenlik önlemleri şu şekilde sıralanabilir:

● Batarya kolay erişilebilecek bir bölgede, yalıtkan ve ısıya dayanıklı batarya kutusu içerisinde bulunacaktır.

● Kablolar kapalı bölgelerden geçirilecektir, çıplak kablo kullanılmayacaktır.

● Kablo bağlantıları bantlar veya makaronlar ile kapatılacak ve kablo demetleri plastik kelepçeler ile toplanacaktır.

● Batarya bağlantıları bataryadan gelecek akıma dayanıklı ve dışarıya izole konnektörler ile sağlanacaktır.

● Direksiyon dönüşünün gereğinden fazla olmasını engellemek üzere limit anahtarı (limit switch) kullanılacaktır.

4.7.4 Kişisel Güvenlik Önlemleri

Yarışmacı için dikkat edilmesi gereken güvenlik önlemleri:

● Araca olan mesafa korunmalı, çok yaklaşılmamalıdır.
● Araç çalıştırılmadan önce tüm kontroller yapılmalı kaçak vb herhangi bir durum olup olmadığı kontrol edilmelidir.
● Anahtar kontrol edilerek aracın geri yada ileri yöne hareket edeceği kontrol edilmelidir.

5. Yazılım Mimarisi

Yazızılımlar C++ ve Python dilleri kullanılarak ROS tabanlı çalışacak şekilde node'lar halinde yazılmıştır. İşletim sistemi olarak da Ubuntu 20.04 kullanılmıştır.

5.1. Şerit Tespit Algoritması

Şerit tespit (line detection) algoritması, araçta bulunan ve 45 derece açı ile aşağıya doğru bakacak şekilde konumlandırılmış kameradan görüntüyü girdi olarak almaktadır. OpenCv kullanılarak, kameradan gelen anlık görüntüler üzerinden şerit tespiti yapılmaktadır. OpenCv kütüphanesinin tercih edilme sebebi, bünyesinde görüntü işlemek için birçok hazır metot bulundurmasıdır. Şerit tespit algoritmasının amacı, şeritlerin çıkarılması ve aracın şeritler içerisinde kalacak şekilde sürülmesidir. Bunu sağlayabilmek için kameradan gelen anlık görüntülerin düzeltilmesi ve gürültülerin giderilmesi sağlanmaktadır. Şerit tespit sisteminin Robotic Operating System (ROS – Robotik İşletim Sistemi) düğümleri (node) kullanılarak gerçekleştirilmiştir.

Şerit tespit algoritması için öncelikle aşağıya bakan kameranın ROS topic'ine abone olunur. Kameradan gelen görüntünün işlenebilir bir formatta olması için CvBridge fonksiyonu yardımıyla uygun formata dönüşümü sağlanmaktadır. Bu işlem sonucunda işlenecek anlık görüntü hazır hale gelmektedir.

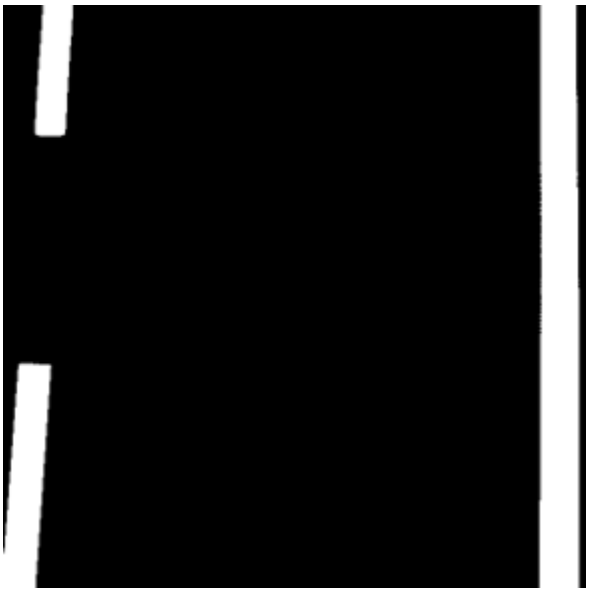
Şekil 37. Kamera Sensöründen Gelen Anlık Görüntü

Şekil 38. Kuşbakışı Görünüm

Kamera sensöründen gelen anlık görüntü Şekil 37'de görülmektedir. Bu adımdan sonra görüntüde sadece şeritleri tespit etmeyi amaçladığımız için görüntü üzerinde dört adet nokta belirlenmiştir. Bu noktaları kullanarak görüntüye perspektif dönüşüm uygulanmıştır. Perspektif dönüşüm sonucunda Şekil 38'deki görüntü elde edilmiştir. Bu dönüşüm sayesinde görüntüyü kuşbakışı şekilde görüntülenir ve ilgilenmediğimiz kısımlar görüntüden çıkarılmıştır. Bu sayede şerit tespiti kolaylaşmıştır.

Şekil 38'deki perspektif dönüşüm uyguladığımız görüntü, renk uzayı dönüşümü ile gri tonlamalı bir görüntüye çevrilir. Bu sayede görüntünün her bir pikselin renk değeri 0 – 255 arası tam sayı değeri alır. Gri tonlamaya çevrilmiş görüntü Şekil 39'da görülmektedir. Renk uzayı dönüşümünden sonra daha yumuşak bir görüntü elde etmek için gri tonlamalı görüntü GaussianBlur metodu ile bulanıklaştırma işlemi uygulanmıştır. Bulanıklaştırma metodu ile keskin olan piksel geçişlerini yumuşatmak amaçlanmıştır. Bu işlem sonrasında Şekil 39'da görülen gri tonlamalı görüntü maksimum ve minimum eşik değerlerinde eşikleme işlemini vermiş olduğumuz mod parametresi yardımı ile görüntümüz Şekil 40'ta görüldüğü gibi eşiklenmiştir. Eşikleme işleminde iki eşik değeri arasında kalan piksel değerleri beyaz olurken bu eşik değerleri dışında olan piksellerin değerleri sıfıra eşitlenerek tam olarak siyah beyaz bir görüntü elde edilmiştir. Eşikleme işleminden sonra resimdeki her bir pikselin renk değeri 0 veya 255 tam sayı değerleri olmuştur. Bu görüntüde 255 renk değerindeki pikseller sol kesikli şeritleri ve sağ şeridi göstermektedir. Böylece şeritlerin belirlenmesi tamamlanmış olur.

Şekil 39. Gri Tonlama Görüntü



Şekil 40 . Eşiklenmiş Görüntü

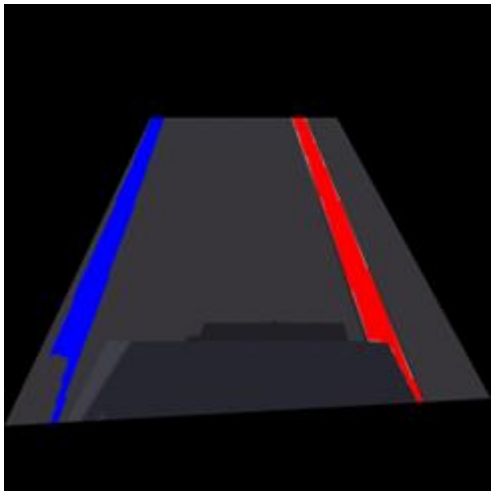
Şekil 40'ta görülen elde etmiş olduğumuz eşiklenmiş görüntü üzerinde tüm pikselleri tarayıp kontrol edilmesi için bir döngü oluşturulmuştur. Döngü içerisinde tarama işlemine Şekil 40'taki görüntümüzün sol üst noktasından sağ tarafa doğru tarama işlemini gerçekleştirmiştir. Bu döngü sonucunda 255 renk değerine sahip olan piksellerimizin koordinatlarını tarama sırasına göre bir liste değişkeni içerisinde saklanmıştır. Daha sonra sağ şerit başlangıç ve sol kesikli şerit başlangıç piksel koordinatlarını tespit edip bir liste değişkeninde tutmak için iki yeni liste değişkeni oluşturulmuştur. Bu sınıflandırmayı gerçekleştirmek için görüntümüzün genişliğinden yararlanılmıştır. Bu iki liste oluşturulduktan sonra bu listelerin sıfırıncı indeksleri sağ şerit ve sol kesikli şerit başlangıç piksellerimiz olmuştur. Başlangıç pikselleri belirlendikten sonra her iki şerit için iki farklı döngü yardımı ile tüm beyaz piksellerin koordinatlarının tutulmuş olduğu liste içerisinde sıra ile kontrol sağlanmıştır. Örneğin sağ şerit başlangıç pikseli ile beyaz pikseller listesi içerisinde birer birer kontrol sağlanmış ve mesafe belirli bir değerin altında ise bu piksel sağ şerit listesinde tutulmuştur. Her bir döngü sonunda başlangıç pikseli son bulunan piksel olarak güncellenmiştir. Mesafe olarak koordinatları bilinen iki noktanın arasındaki mesafe formülü uygulanmıştır. Aynı işlem sol kesikli şerit içinde tekrarlanmıştır, sağ şerit ve sol şerit pikselleri bu şekilde belirlenmiştir. Bu iki şeride ait pikseller ayrı liste değişkenlerinde tutulmuştur ve algoritmamızın ilerleyen kısımlarında kullanılmak üzere bu liste değişkenleri matris formuna

dönüştürülmüştür. Algoritmamızda sağ şeritler kırmızı sol kesikli şeritler mavi renk olarak modellenmiştir. Bu işlemler sonucunda tespit edilen şeritler Şekil 41'deki gibi görüntülenmiştir.

Şekil 41. Şerit Tespiti Sonucu Görüntü

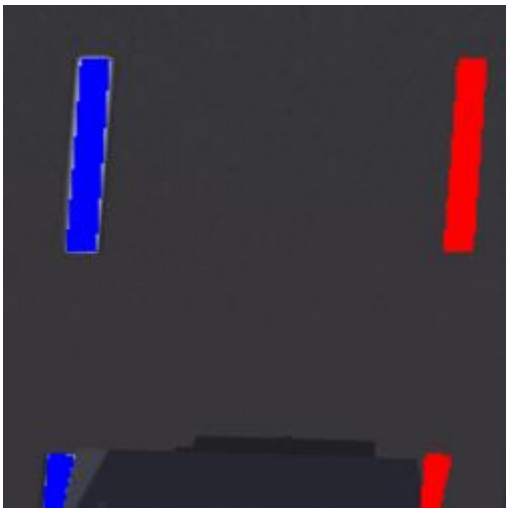
Araç kamerasından gelen anlık görüntüde sağda bulunan sürekli şeritler veya solda bulunan kesikli şeritlerden herhangi biri görülmediğinde örneğin, sol da bulunan kesikli şerit görünmediğinde algoritma sol kesikli şeridi tespit etmek için kontrollerini gerçekleştireceği sol kesikli şerit başlangıç pikseli bulamayacağı için sağ şeridi üstte anlatıldığı gibi tespit işlemini gerçekleştirip Şekil 42 ve Şekil 43'te görüldüğü gibi sağ şeridi belirli birim uzaklığında soluna kopyalayarak bir sol şerit oluşturur ve tekrar sol kesikli şerit görüntüye girene kadar kopyalama işlemini devam ettirerek doğrusal şekilde yol alması sağlanmıştır.

Şekil 42. Sağ Şeridin Perspektif Dönüşümü

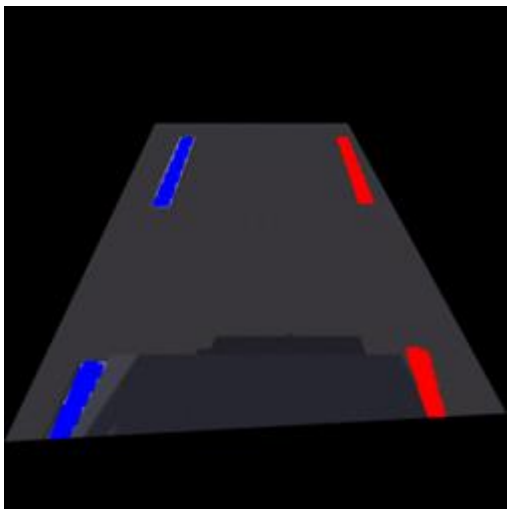


Şekil 43. Dönüşümün kamera frame'ine oturtulması

Diğer bir durumda araç kameramızdan gelen anlık görüntülerde sağ şerit kameramızın görüşünden çıkarsa, algoritmamız sağ şerit tespiti için bir başlangıç noktası bulamayacaktır ve sağ şerit tekrar kamera görüşüne girene kadar, tespit etmiş olduğu sol kesikli şeridi Şekil 44 ve Şekil 45'te görüldüğü gibi belirli bir birim uzaklıkta sağına kopyalama işlemini gerçekleştirmesi düşünülmüştür.

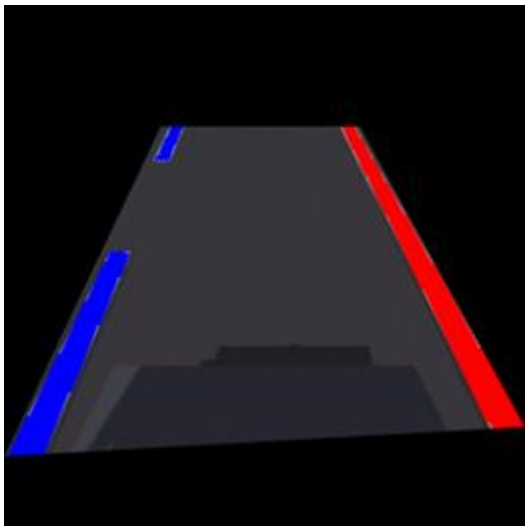


Şekil 44. Sol Kesikli Şeridin Perspektif görüntüsü



Şekil 45.Dönüşümün kamera frame'ine oturtulması

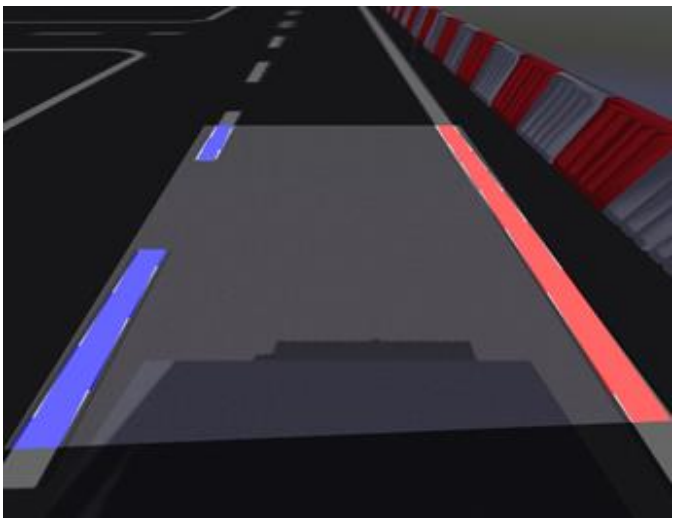
Kamera iki şeridi de görüntüleyemezse şerit olmadığına dair uyarı vermektedir. Algoritma aynı zamanda görüşü içerisinde bulunan sağ şerit daha önce dönüştürülmüş olan matris formuna giderek en büyük ve en küçük y değerlerine sahip olan indisleri belirleyip, bu indislerin dikey bileşenleri olan y değerlerini yatay bileşenleri olan x değerlerine oranlayıp bu ifadenin tanjantını alarak eğim kontrolü de sağlanması da amaçlanmıştır.



Şekil 46. Ters Perspektif Dönüşüm Uygulanmış Görüntü

Başlangıçta perspektif dönüşüm yardımıyla kuşbakışı görünüm elde ettiğimiz görüntü şeritler belirlendikten sonra ters perspektif dönüşüm uygulanmış ve Şekil 46'daki görüntü elde edilmiştir.

Son olarak şeritlerin belirlendiği Şekil 46'daki görüntü kameradan gelen anlık görüntü ile belirli oranlar ile toplama işlemine sokulmuştur ve Şekil 47'de ki görüntü elde edilmiştir. Algoritma simülasyon ortam içerisinde test edilerek en optimum haline getirilmiştir.



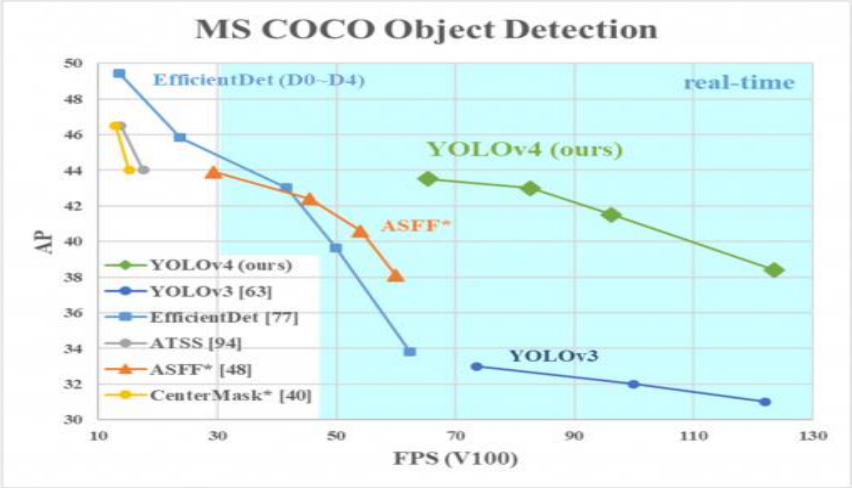
Şekil 47. Şerit Tespitinde Ağırlıklı Toplam Sonucu Elde Edilmiş Görüntü

5.2. Yapay Zeka Modeli

Otonom sürüşün sağlıklı bir şekilde gerçekleşmesi aracın karar verme mekanizmasının doğru işleyebilmesine bağlıdır. Araç, algılanan çevreden gelen veriler sonucu karar vererek sürüşünü devam ettirmektedir. Bu açıdan bakıldığı zaman çevre algılama otonom sürüşün temel

etkenlerinden biridir. Aracımız, yapay zeka ve derin öğrenme algoritmaları sayesinde trafik işaretlerini ve trafik ışıklarını algılamaktadır.

Yarışma görevleri göz önünde bulundurularak nesne tespit modelleri araştırılmıştır. Gerçek zamanlı nesne tanıma düzeyinin, görevleri başarılı bir şekilde gerçekleştirmede önemli bir etken olduğuna karar verilmiştir. Gerçek zamanlı nesne algılama konusunda başarılı sonuçlar verdiğinden YOLO (You Only Look Once) modeli seçilmiştir. YOLO, konvolüsyonel sinir ağlarını (CNN) kullanarak nesne tespiti yapmaktadır. Tek aşamalı (one-state) nesne takibi yapması onu diğer algoritmalardan daha hızlı yapmaktadır. Görüntü bir kez sinir ağından geçmektedir. Bu sayede tek seferde görüntüdeki tüm nesneler için tahmin yapabilmektedir. YOLOV4 ve YOLOV4-TİNY modelleri için deneme eğitimleri yapılarak sonuçlara bakılmıştır. Sonuçlar doğrultusunda algoritmanın farklı versiyonları arasından doğruluk oranı yüksek olmakla birlikte daha hızlı tahminler yapabildiğinden YOLOV4 modeli seçilmiştir. Modellerin karşılaştırılması şekil 48'te gösterilmiştir.



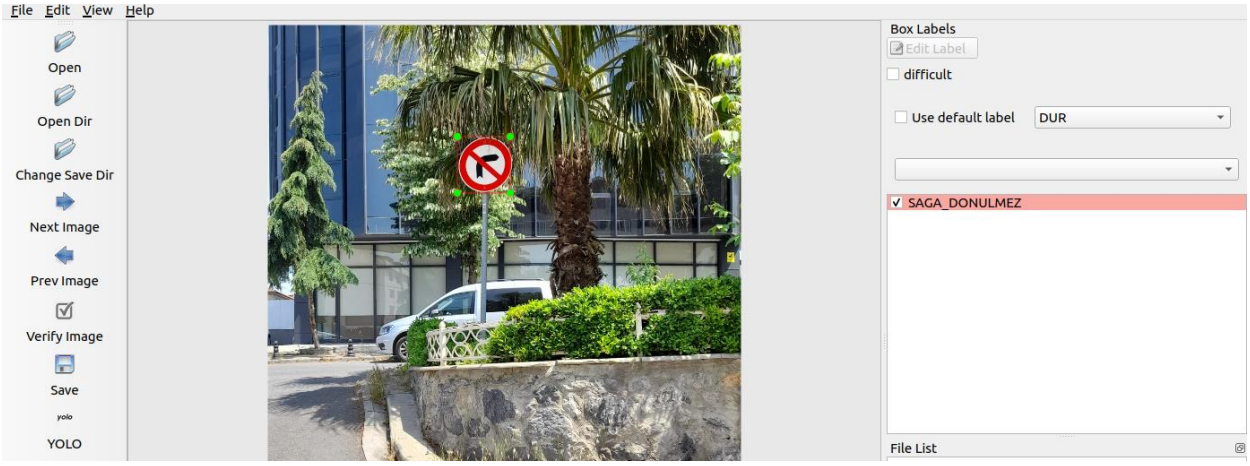
Şekil 48. Farklı Modellerin Hız ve Doğruluk Açısından Karşılaştırılması [11]

5.2.1. Trafik İşaretleri ve Trafik Işığı Tespiti

Veri kalitesi model eğitiminin sağlıklı bir şekilde gerçekleşmesi için önemlidir. Birçok farklı açıdan ve durumdan veriler toplanmıştır. Takımız tarafından hem sahada araç kamerasından hem de Google haritalar üzerinden görüntüler elde edilmiştir. Geçen sene toplanan verilerden kullanabileceklerimiz ayrıştırılmış yeni toplanan veriler üzerine eklenerek 11.000 fotoğraftan oluşan veri kümesi oluşturulmuştur.

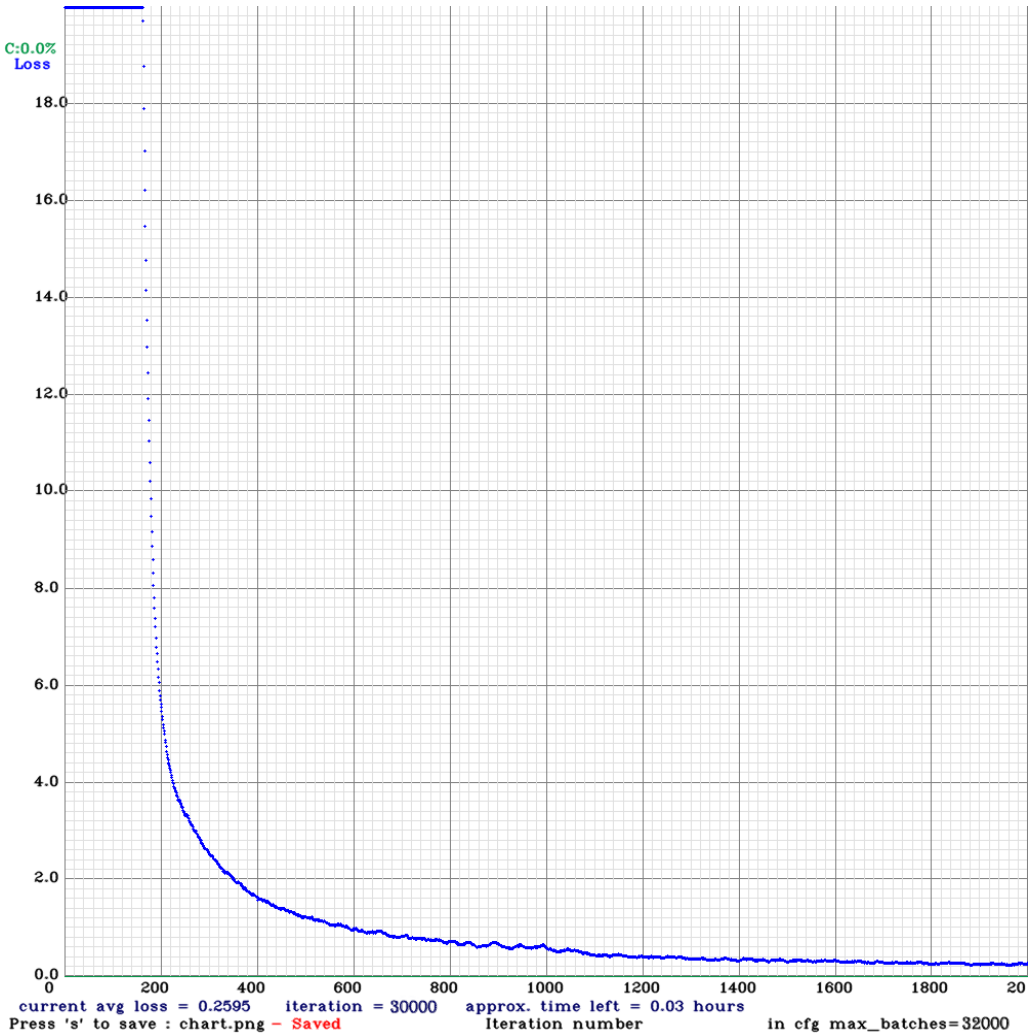
Etiketleme labelImage programında YOLO formatına uygun bir biçimde gerçekleştirilmiştir. Etiket dosyası 5 adet parametre içermektedir. Bunlar etiketlenen nesnenin sınıf numarası, nesnenin yükseklik ve genişliği, nesnenin merkezinin x ve y noktasıdır.Özel model YOLOV4 eğitimi için 16 sınıftan oluşan bir veri seti hazırlanmıştır. Hazırlanan veri seti yüzde seksen train ve yüzde yirmi test olmak üzere iki kısma ayrılmıştır.

Veri sınıfında dur, durak, girilmez yol, ileriden sola mecburi, ileriden sağa mecburi, ileri ve sağa mecburi, ileri ve sola mecburi, park yeri, park edilmez, sağa dönülmez, sola dönülmez, trafik lambası yeşil, trafik lambası kırmızı, trafik lambası sarı, engelli park yeri ve dönel kavşak yer almaktadır.



Şekil 49. LabelImage programı

Darknet, C/C++ ve CUDA kullanılarak yazılmış makine öğrenim kütüphanedir. Sınıf sayısı ve seçilen nesne tespit modeline (YOLOV4) göre konfigürasyon dosyası düzenlenmiştir. CUDA, yoğun hesaplama gerektiren programları hızlandırmamızı sağlayan bir hesaplama platformudur. Gerekli GPU desteğini sağlayan bilgisayar üzerinde Darknet ile CUDA ve opencv kullanılarak eğitim gerçekleştirilirmiştir.



Şekil 50. Model ortalama hata değerinin iterasyon sayısına göre değişim grafiği

Model eğitim süreci belirli iterasyonlarda durdurulup test edilerek devam etmiştir. Test veri setindeki görüntüler ve Darknetros [12] ile araç kamerasından canlı alınan görüntüler üzerinde ağırlık dosyası çıktılarına bakılmıştır. Nesne tespit doğruluk oranlarını göz önünde bulundurularak

veri setine eklemeler yapılmıştır. Nesne algılama modellerini değerlendirmek için ortalama hassasiyet değeri (mAP - Mean Average Precision) ve ortalama hata değeri (Avg) kullanılmaktadır. 89.68 mAp değeri ve 0.259 Avg değeri yeterli görülerek 30.000 iterasyonda eğitim tamamlanmıştır.

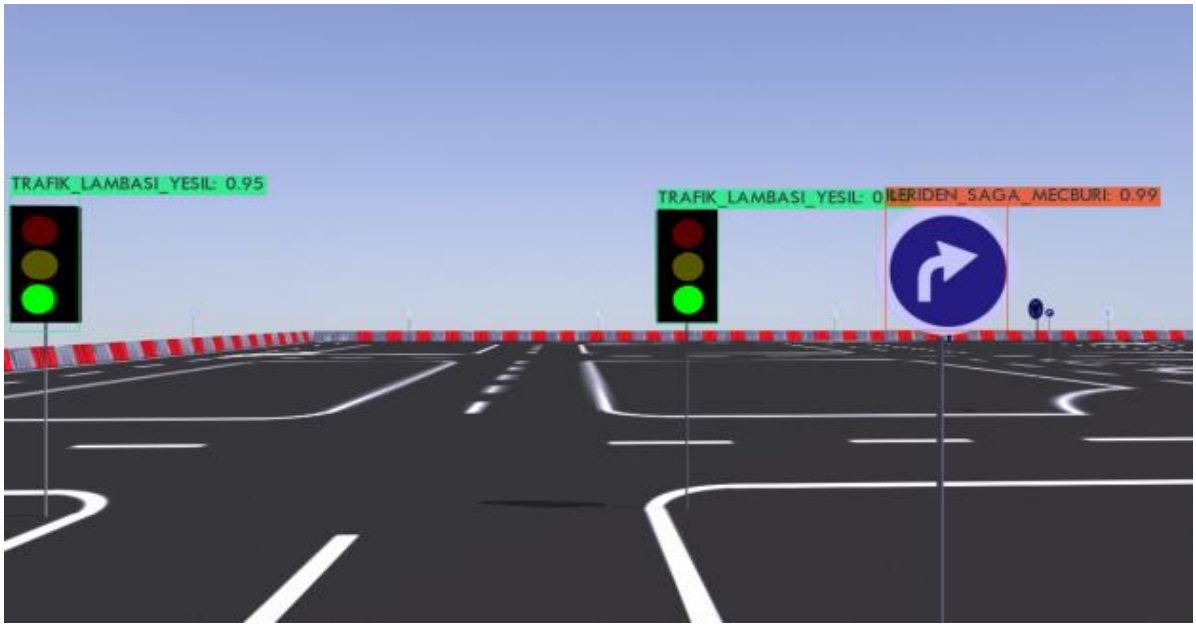
```
class_id = 0, name = GIRILMEZ_YOL, ap = 95.59%          (TP = 101, FP = 14)
class_id = 1, name = SOLA_DONULMEZ, ap = 87.62%         (TP = 115, FP = 25)
class_id = 2, name = SAGA_DONULMEZ , ap = 89.66%        (TP = 33, FP = 15)
class_id = 3, name = ILERIDEN_SOLA_MECBBURI, ap = 92.85%        (TP = 64, FP = 29)
class_id = 4, name = ILERIDEN_SAGA_MECBURI, ap = 92.71%         (TP = 68, FP = 17)
class_id = 5, name = DUR, ap = 67.09%           (TP = 26, FP = 14)
class_id = 6, name = PARK_YERI, ap = 94.22%     (TP = 135, FP = 48)
class_id = 7, name = PARK_EDILMEZ, ap = 96.18%          (TP = 155, FP = 30)
class_id = 8, name = DURAK , ap = 87.07%        (TP = 98, FP = 30)
class_id = 9, name = ILERI_VE_SOLA_MECBURI, ap = 93.63%         (TP = 79, FP = 34)
class_id = 10, name = ILERI_VE_SAGA_MECBURI, ap = 0.00%         (TP = 0, FP = 0)
class_id = 11, name = TRAFIK_LAMBASI_YESIL, ap = 95.50%         (TP = 37, FP = 10)
class_id = 12, name = TRAFIK_LAMBASI_KIRMIZI, ap = 96.75%       (TP = 39, FP = 1)
class_id = 13, name = TRAFIK_LAMBASI_SARI, ap = 100.00%         (TP = 1, FP = 0)
class_id = 14, name = ENGELLI_PARK_YERI, ap = 95.39%    (TP = 61, FP = 3)
class_id = 15, name = DONEL_KAVSAK, ap = 96.00%         (TP = 24, FP = 6)
```

Şekil 51. Simülasyon ortamı için eğitilen modelin mAP çıktısı

```
class_id = 0, name = GIRILMEZ_YOL, ap = 95.13%          (TP = 98, FP = 4)
class_id = 1, name = SOLA_DONULMEZ, ap = 94.85%         (TP = 115, FP = 15)
class_id = 2, name = SAGA_DONULMEZ, ap = 85.79%         (TP = 32, FP = 10)
class_id = 3, name = ILERIDEN_SOLA_MECBBURI, ap = 92.43%        (TP = 59, FP = 15)
class_id = 4, name = ILERIDEN_SAGA_MECBURI, ap = 94.18%         (TP = 65, FP = 6)
class_id = 5, name = DUR, ap = 87.90%           (TP = 26, FP = 8)
class_id = 6, name = PARK_YERI, ap = 93.93%     (TP = 129, FP = 31)
class_id = 7, name = PARK_EDILMEZ, ap = 96.26%          (TP = 153, FP = 15)
class_id = 8, name = DURAK, ap = 90.02%         (TP = 93, FP = 23)
class_id = 9, name = ILERI_VE_SOLA_MECBURI, ap = 92.19%         (TP = 76, FP = 21)
class_id = 10, name = ILERI_VE_SAGA_MECBURI, ap = 0.00%         (TP = 0, FP = 0)
class_id = 11, name = TRAFIK_LAMBASI_YESIL, ap = 98.62%         (TP = 36, FP = 3)
class_id = 12, name = TRAFIK_LAMBASI_KIRMIZI, ap = 100.00%      (TP = 40, FP = 0)
class_id = 13, name = TRAFIK_LAMBASI_SARI, ap = 100.00%         (TP = 1, FP = 0)
class_id = 14, name = ENGELLI_PARK_YERI, ap = 99.15%    (TP = 61, FP = 1)
class_id = 15, name = DONEL_KAVSAK, ap = 100.00%        (TP = 24, FP = 1)
```

Şekil 52. Gerçek ortam için eğitilen modelin mAP çıktısı

Simülasyon ve gerçek ortamlarda kullanılmak üzere iki farklı model eğitimi gerçekleştirlmiştir. Model eğitimleri için iki farklı veri kümesi hazırlanıp kullanılmıştır. İlk veri kümesi simülasyon ortamından elde edilen görüntülerden oluşmaktadır. Simülasyon ortamında gerçekleştirdiğimiz testler için kullanılmıştır. Gerçek ortamda eğitilen model ile nesne algılama yapıldığında doğruluk oranları düşük çıkmıştır.Bu yüzden gerçek ortamda gerçekleştireceğimiz testler için ikinci bir veri kümesi oluşturulmuştur. Bunun sebebi ise her iki ortamda da doğruluk oranı yüksek nesne algılama yaparak görevleri doğru bir şekilde gerçekleştirmektir.Eğitilen modellerin nesne algılama çıktı örnekleri Şekil 53 ve 54'te gösterilmiştir.



Şekil 53. Simülasyon ortamında tespit edilen trafik ışıkları ve işaretleri

Şekil 54. Gerçek ortamdan tespit edilen trafik işaretleri

5.3 Kontrol

Araç kontrol biriminde tasarlanırken kullanılan ve geliştirilen algoritmalar bu bölümde sunulmaktadır.

5.3.1 Kullanılan Sensör Füzyon Algoritması (Kalman Süzgeci)

Kalman süzgeci, durum-uzay gösterimi ile modellenen dinamik bir sistemin geçmiş durumlarıyla birlikte giriş ve çıkış değerlerini süzerek şimdiki durumlarını tahmin edebilen tekrarlamalı bir matematiksel işlemdir. Kalman süzgeci lineer sistemler için kullanılmaktadır. Lineer olmayan sistemler için ise Genişletilmiş Kalman Süzgeci kullanılmaktadır. Lineer olmayan sistem lineerleştirilmekte ardından Kalman süzgece benzer işlemlerle tahmin gerçekleştirilmektedir [13].

Projede lineer olmayan sisteme sahip aracın konum bilgisini elde etmek için GPS ve AHRS sensörlerinin Genişletilmiş Kalman Süzgeci yardımıyla füzyonu yapılmaktadır. Denklem 1'de aracın lineer olmayan sistem modelinin matris olarak gösterimi görülmektedir. Sistem bilgisi olarak Genişletilmiş Kalman Süzgeci'nde kullanılmaktadır. Denklem 2'de ise Genişletilmiş Kalman Süzgeci'nden elde edilecek durum bilgisi için sistemin ölçüm modeli matris gösterimi şeklinde görülmektedir.

$$\mu_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \omega_k \\ v_k \\ a_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_{k-1}\Delta t \cos(\theta_{k-1}) \\ y_{k-1} + v_{k-1}\Delta t \sin(\theta_{k-1}) \\ \theta_{k-1} + \omega_{k-1}\Delta t \\ \omega_{k-1} \\ v_{k-1} + a_{k-1}\Delta t \\ a_{k-1} \end{bmatrix} = f(x_{k-1}, u_k)$$

Denklem 1. Sistem Modeli

$$z_k = h(x_k) = \begin{bmatrix} x_{gps_k} \\ y_{gps_k} \\ a_k \\ \omega_k \end{bmatrix} = H \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \omega_k \\ v_k \\ a_k \end{bmatrix}$$

Denklem 2. Ölçüm Modeli

Denklem 3'te görülen tahmin aşamasında, AHRS sensöründen elde edilen ivme ve yönelim bilgileri sistem modeli kullanılarak işlenir. Bu işlem sonucunda aracın şimdiki ("k" anındaki) konum bilgisi μ_k ve bu konuma itimadı belirten kovaryans matrisi P_k hesaplanmaktadır. Burada sistem modelinin durum modeline göre lineerleştirilmiş hali F matrisi, durum vektörünün sistem gürültü vektörüne göre lineerleştirilmiş hali G matrisi, sistem gürültü matrisi Q olarak yer almaktadır.

$$\mu_k^{\sim} = f(\mu_{k-1}, u_k)$$

$$P_k^{\sim} = F_{k-1}P_{k-1}F_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T$$

$$Q = \begin{bmatrix} W_\omega\ (açısal\ hız) & 0 \\ 0 & W_a\ (doğrusal\ ivme) \end{bmatrix}$$

Denklem 3. Tahmin Aşaması

Denklem 4 ve Denklem 5'in kullanıldığı düzeltme aşamasında, tahmin aşamasında hesaplanan konum bilgisi ve kovaryans matrisi, GPS'ten "k" zamanı (şimdi zaman) için elde edilen enlem ve boylam verisi kullanılarak güncellenir.

$$K_k = P_k^{\sim} H^T (H P_k^{\sim} H^T + R)^{-1}$$

Denklem 4. Kalman Kazancı hesabı

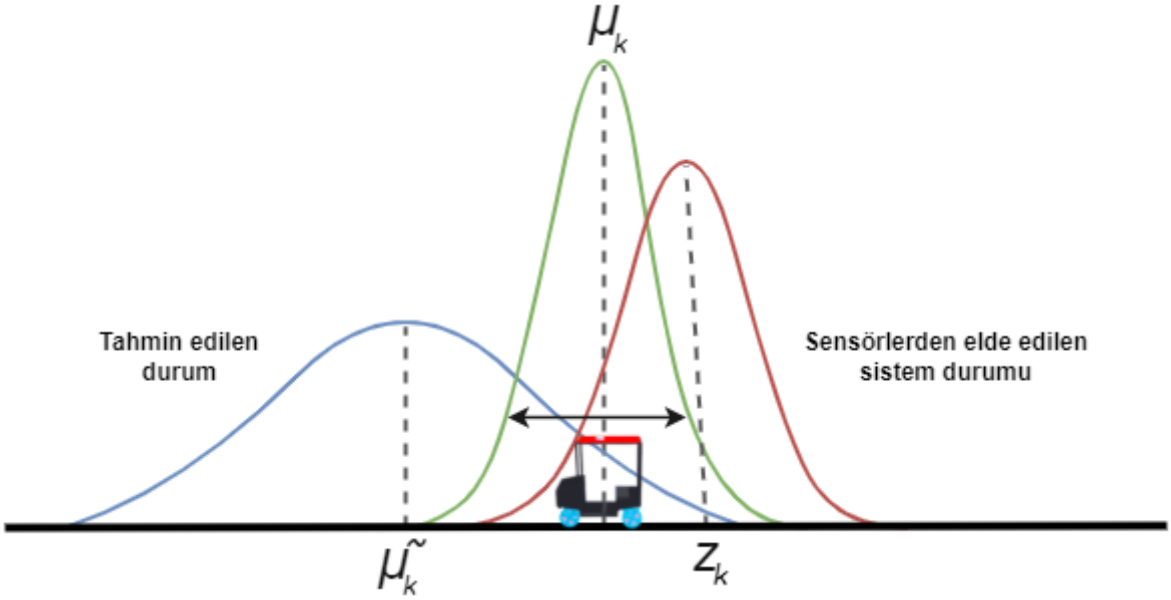
Düzeltme aşamasında ilk olarak Denklem 4'teki denklem ile Kalman Kazancı hesaplanır. Bu denklemdeki R ise GPS ve AHRS sensörlerinden gelen gürültü matrisidir. Sonrasında, "k" anındaki ölçüm modeli z_k ve Kalman Kazancı K_k ile konum bilgisi ve kovaryans matrisi güncellenir.

$$\mu_k = \mu_k^{\sim} + K_k(z_k - H_k\mu_k^{\sim})$$

$$P_k = (I - K_k H)P_k^{\sim}$$

Denklem 5. "k" anındaki (şimdiki) kovaryans matrisinin hesabı ve sistem durumu

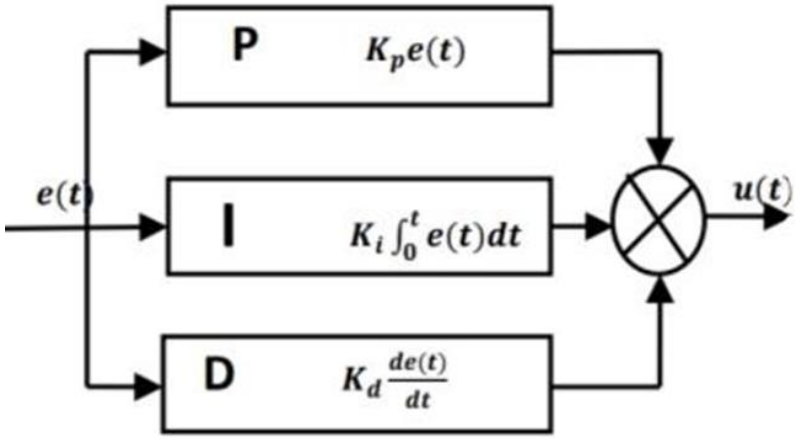
Şekil 9'da aracın konum tespiti için gerçekleştirilen kalman süzgeci aşamaları, tahmin edilen durum (Tahmin) ve sensörlerden elde edilen sistem durumu (Düzeltme) olarak görülmektedir. Tahmin edilen durum kısmında kestirilen (estimate) konum, sensörlerden elde edilen ölçüm ile füzyon edilerek aracın tahmini (predict) konumu elde edilir [14].



Şekil 55. Aracın konum tespiti için Kalman Süzgeci aşamalarının gösterimi

5.3.2. Kullanılan Kontrol Birimi (PID)

Araç, ackermann direksiyon geometrisi prensibine sahiptir ve bu prensip doğrultusunda kontrol edilmektedir. Araç kontrolünün temelinde Şekil 56'da blok diyagramı görülen PID kontrolör bulunmaktadır. Kontrol edilecek veri için istenen değer ile mevcut değer arasındaki hata *e(t)*, sıfırlanmak üzere oran, integral ve türev işlemlerine tabi tutulmaktadır. Hatayı sıfırlamak ve sıfıra yakın tutmak için gerekli sistem giriş değerleri kontrolün sonucunda kontrol çıkışı *u(t)* olarak üretilmektedir.

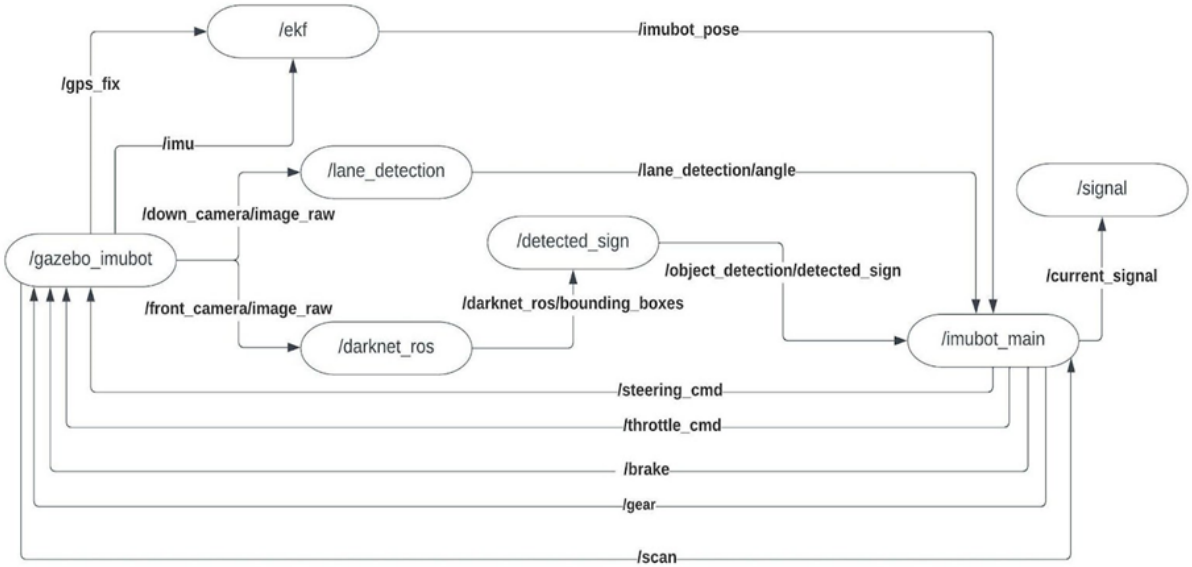


Şekil 56. PID kontrol blok diyagramı [15]

Kontrol algoritması, hareket planlama modülünden gelen komutlara göre aracı kontrol etmektedir. Hareket planlama algoritmasında planlanan hareketleri sağlayacak hız, direksiyon, fren ve vites (ileri-geri) değerlerini araç üzerindeki donanımlara ileterek gerçeklemeyi sağlamaktadır. Ayrıca araç hareketlerine bağlı sinyalizasyonu da sağlamaktadır.

Planlanan harekete uygun olarak belirlenen yön ve hız değerinde aracı sürmek için araç motorunu kontrol eden motor sürücüye yön ve hız verisini göndermektedir. Belirlenen direksiyon (steering) açısına göre yönelim sağlamak için ise direksiyon kontrolünü sağlayan step motora direksiyon açısını vermektedir. Aracın durması veya yavaşlaması gereken durumlarda da gelen fren tork değerine göre araç frenini kontrol eden servo motora uygun pozisyon verisini göndermektedir.

ROS (Robotik İşletim Sistemi / Robot Operating System) ortamında gerçeklenen kontrol modülü yine ROS ortamında gerçeklenen hareket planlama modülünde belirlenen verileri araç üzerindeki donanımlara ileterek aracı kontrol etmektedir. Kontrol, dört rostopic'i üzerinden yapılmaktadır.



Şekil 57. ROS yazılım mimarisi

i) ROS topic /steering_cmd

Araca direksiyon açısı vermek için kullanılan topic'tir. Bu topic, hareket planlama algoritmasında belirlenen direksiyon açısına göre -0.67 ile 0.67 radyan arasında değer almaktadır. Bu radyan açı

değerine göre aracın tekerlekleri -38 ile 38 derece arasında açı yapacak şekilde yönelim yapabilmektedir. Eksi değerler sağa dönüşü, artı değerler ise sola dönüşü sağlamaktadır.

Tekerlek dönme açılarının -38 ile 38 derece arasında değer alması aracın fiziksel kısıtlarından kaynaklandığından buna bağlı radyan değer aralığı kontrol algoritması için -0.67 ile 0.67 radyan arasında olmaktadır.

Topic üzerinde aktarılan verinin tipi (topic message type): 64-bit ondalıklı sayı (std_msgs/Float64)

ii) ROS topic /throttle_cmd

Araca hız değeri vermek için kullanılan topic'tir. Aracın hızı, fiziksel kısıtlamalar ve yapay zekanın algı verimliliğini arttırmak için maksimum 10 m/s ile sınırlandırılmıştır.

Topic üzerinde aktarılan verinin tipi (topic message type): 64-bit ondalıklı sayı (std_m sgs/Float64)

iii) ROS topic /gear

Aracın ileri yada geri gitmesini sağlamak için kullanılan topic'tir. 0 ve 1 değerlerini alabilmektedir. 0 değeri aracı ileri doğru sürmeyi sağlarken, 1 değeri geri doğru sürmeyi sağlamaktadır.

Topic üzerinde aktarılan verinin tipi (topic message type): 8-bit tamsayı (std_msgs/UInt8)

iv) ROS topic /brake

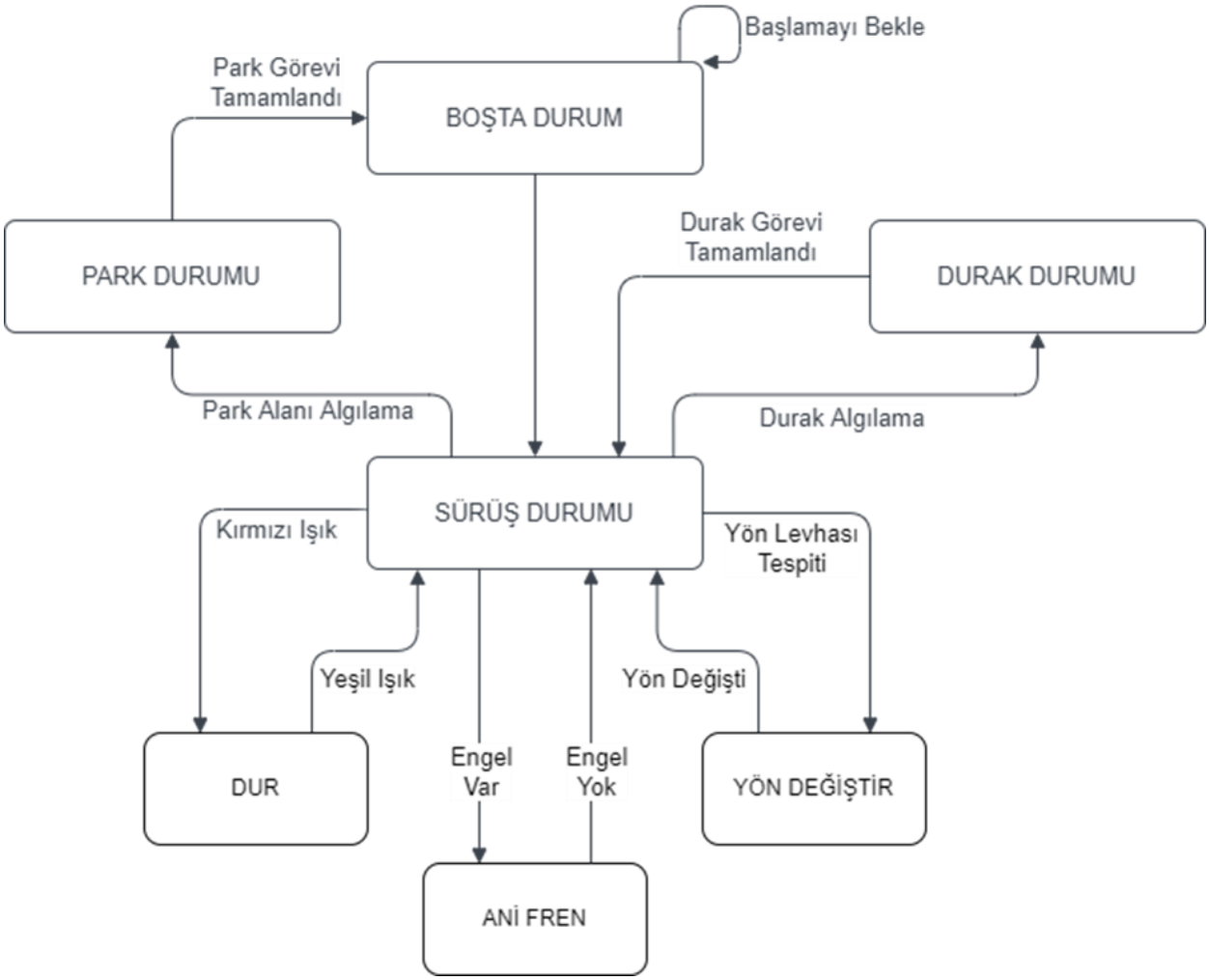
Araca fren yaptırmak için kullanılan topic'tir. Topic'e verilen değer, kontrolör tarafından newton olarak değerlendirilmektedir.
Topic üzerinde aktarılan verinin tipi (topic message type): 64-bit ondalıklı sayı (std_msgs/Float64)

5.4. Hareket Planlama

Aracın hareket planlaması dört farklı durum temelinde gerçekleşmektedir. Şekil 58'de durum diyagramı olarak da görülen bu durumlar; aracın otonom harekete hazır olduğu boşta durum, aracın seyir halinde olduğu yol alanlarında sefer kabiliyeti sağlayan sürüş durumu, durak alanlarında durağa girmesini ve duraktan ayrılmasını sağlayan durak durumu ve park alanında park etmesini sağlayan park durumudur. Planlama, bu görevleri yerine getirmek üzere özelleştirilmiş durumlar arasında geçişler yaparak araca otonom hareket sağlamaktadır.

5.4.1. Boşta Durum

Boşta durumda araç, verilecek görevleri yerine getirmek üzere hazır halde beklemektedir. Araç üzerinde harekete dair herhangi bir planlama veya kontrol yapılmamaktadır. Bu durum, otonom hareketi sağlayan durum makinesinin başlangıcıdır ve bu durumda araca verilecek komut beklenmektedir. Araca bir görev veya komut bilgisi geldiğinde ilgili durumlara geçilerek hareket planlanmaktadır.



Şekil 58. Planlama Durum Makine Diyagramı (State Machine Diagram)

5.4.2. Sürüş Durumu

Sürüş durumu, aracın seyir halinde yol boyunca hareketlerinin planlandığı durumdur. Burada planlama ile birlikte çevre algılama da çalışmaktadır. Araç yola çıktığında aksi bir komut verilmedikçe araç sabit hızda seyir edecek şekilde kontrol edilir. Çevre algılama kısmında, görüntü işleme ile takip edilen şeritlere göre elde edilen seyir konumu ve direksiyon açısına göre aracın yönelimi planlanmaktadır. Ayrıca seyir esnasında engel veya kırmızı ışık gibi durmayı gerektirecek durumlarla karşılaşıldığında gazı kesip fren yapmayı sağlayacak plan kontrolöre iletilmektedir. İstenildiği durumlarda aracın hızı artırılıp azaltılabilmektedir.

5.4.2.1. Düz Yol

Sürüş durumunda; yol boyunca herhangi bir dönüş, engel veya trafik uyarıcısı bulunmadığı durumda çalışan planlama kısmıdır. Bu kısımda, aracın yol boyunca gitmesi için gerekli olan direksiyon açısı görüntü işleme ile elde edilmektedir. Bu işlem için belirli bir ileri bakma mesafesi seçilir. Sonrasında soldaki kesikli şerit çizgisi ve sağdaki düz şerit çizgisi tespit edilip orta noktaları bulunur. Buradan aracın doğrultusu ile ileri bakma mesafesindeki noktanın açı farkı hesaplanır. Bu değer araca direksiyon açısı olarak verilir. Ancak aracın fiziksel kısıtları olması sebebiyle araç direksiyon doğrultusuna dönüşünü ani olarak gerçekleştiremez. Bu yüzden gerekli doğrultuya gelmesi PID kontrolcüsü kullanarak sağlanmaktadır.

5.4.2.2. Trafik Lambaları

Sürüş durumunda trafik lambası ile karşılaşıldığında çalışan planlama kısmıdır. Çevre algılama ile trafik lambası algılandığında görüntü işleme ile trafik lambasında yanan renk tespit edilir. Yeşil ışık yanıyorsa araç sürüşe devam etmektedir. Kırmızı ışık yanıyorsa trafik lambasını geçmeyecek ve yaya geçitlerine girmeyecek şekilde trafik ahlakına uygun belirli bir mesafeye kadar gelip durmaktadır. Bu konumda trafik lambalarını görüş açısında kalmaya devam etmektedir. Bu sayede trafik lambası yeşil yandığı anda araç tekrardan harekete geçmektedir.

5.4.2.3. Dönüşler

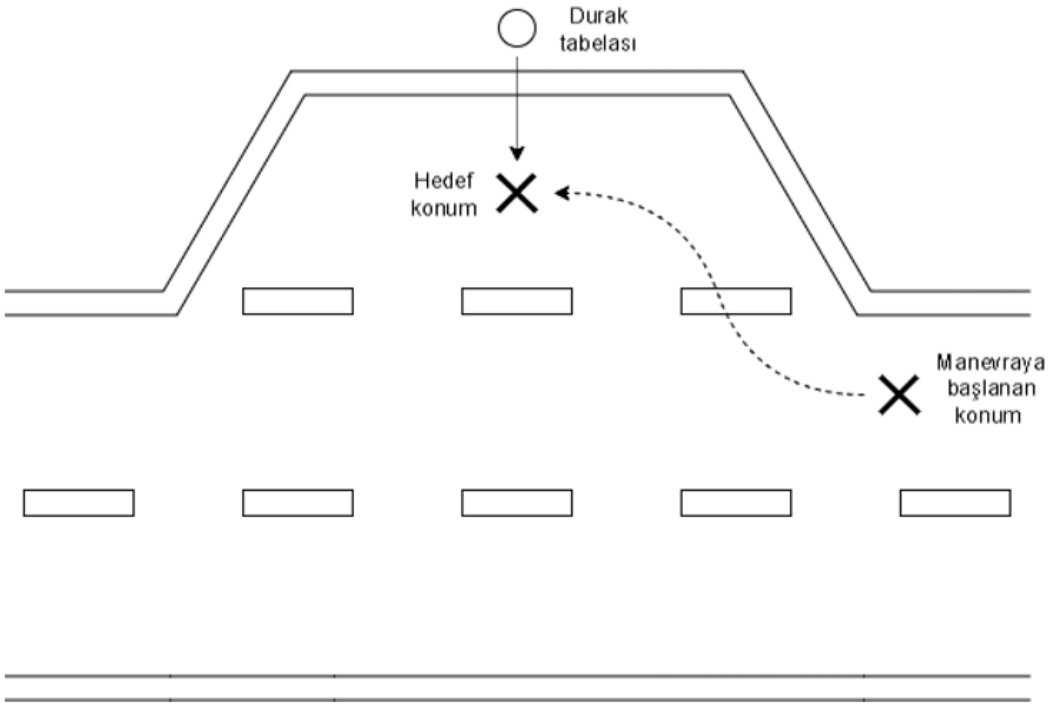
Sürüş durumunda mecburi yön tabelası ile karşılaşıldığında çalışan planlama kısmıdır. Çevre algılama ile mecburi yön levhası algılandığında görüntü işleme ve yapay zeka ile levhanın belirttiği yön bilgisi tespit edilmektedir. Tespit edilen yön doğrultusunda, düz gitmede olduğu gibi görüntü işleme ile şerit tespiti yapılmaktadır. Şeritler tespit edilirken, düz gitmeden farklı olarak şeritlerden birinin şerit çizgileri daha ileride oluşmaktadır. Bu durumda şeridi algılayabilecek uygun bir ileri bakma mesafesi belirlenip direksiyon açısı tespit etmektedir. Belirlenen direksiyon açısı araca verilerek dönüş sağlanmaktadır. Sağa ve sola dönüşler direksiyon açısının durumuna göre tespit edilebilmektedir. Bu sayede uyarıcı olarak aracın üstündeki sağ ve sol işaret lambaları yakılmaktadır.

5.4.3. Durak Durumu

Durak durumu, aracın durak olarak belirlenmiş yol cebine girişi ve çıkışını sağlayacak hareketlerin planlandığı durumdur. Çevre algılama kısmındaki görüntü işleme ve yapay zeka ile durağın algılanmasıyla çalışmaya başlamaktadır. Araç üzerindeki çevre algılama ekipmanları ile aracın konumu ve durağın haritası belirlenerek uygun şekilde durağa girmeyi ve ayrılmayı sağlayacak hareketler planlanmaktadır.

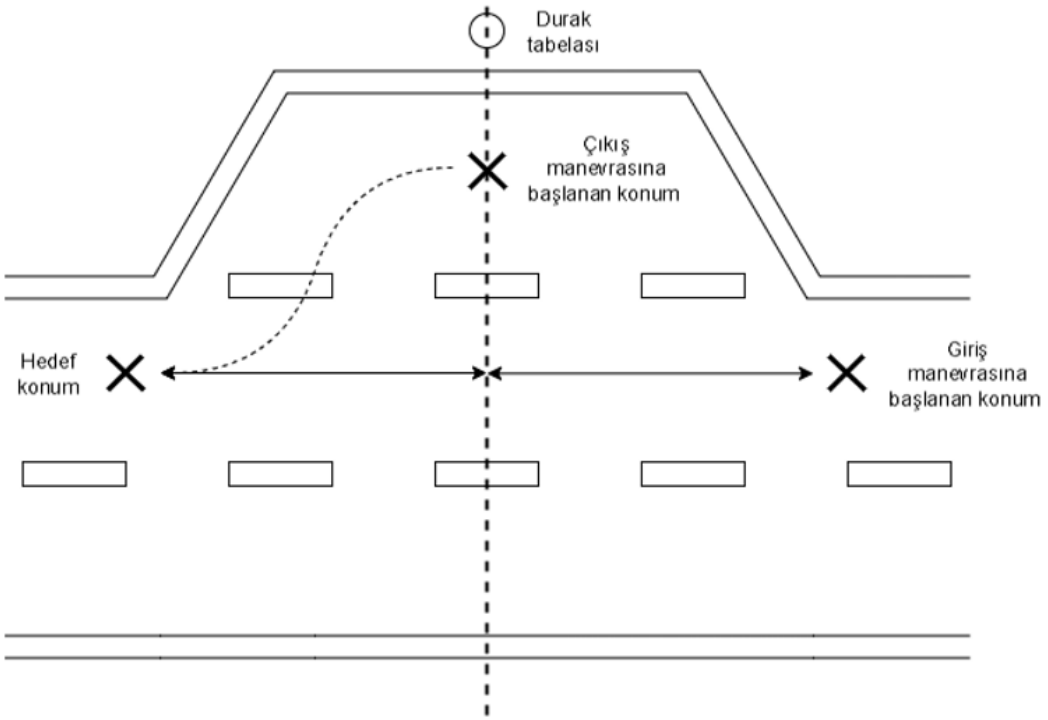
Bir yerin durak olduğunu anlamak için üç unsur kullanılmaktadır. Bunlardan birincisi, aracın sağındaki ve solundaki şeritlerin ikisinin birden kesikli olması, ikincisi durak levhasının tespiti, üçüncüsü ise durak levhasına olan uzaklıktır. Durak levhasının görüntü işleme tarafından tespit edilmesi ile aracın durak planlaması başlamaktadır. Durağa yaklaşırken sağ ve sol şeritlerin kesikli olduğu görüntü işleme tarafından tespit edilmesiyle durağa gelindiği tespit edilir. Durak tabelasının uzaklığının lidar sensörü tarafından tespit edilmesi ve konumunun araç konumuna göre hesaplanmasıyla uygun manevralar planlanarak manevralara başlanır.

Planlanan hareketler aracın durağa girmesi ve çıkması için gereken manevraları kapsamaktadır. İlk olarak aracın durağa girmesi için Şekil 59'da görüldüğü gibi durak tabelasından durağa doğru yönde, yanaşmaya uygun nokta olacak şekilde belirli bir mesafe ilerisi hedef konum olarak belirlenir. Belirlenen bu hedef konum ile aracın mevcut konumu arasında Şekil 59'da görüldüğü gibi yanaşmaya uygun bir yol planlanır. Aracın belirlenen yolu takip etmesini sağlayacak hız ve direksiyon bilgileri '/throttle_cmd' ve '/steering_cmd' topic'leri üzerinden kontrolöre iletilir. Kontrolörün, gelen hız ve direksiyon bilgilerini donanım üzerinde gerçeklemesiyle araç yolu takip ederek durağa yanaşmış olur. Aracın durağa yanaşması ile yolcu alışverişi için beklemeye geçilir.



Şekil 59. Durağa giriş için hareket planlaması.

Belirli bir süre beklemenin ardından tekrar yola çıkmak için hafızaya alınmış olan aracın durağa giriş manevrasına başladığı konumun, Şekil 60'da görüldüğü gibi durak tabelasının yola dik eksenine göre aynalanması ile yeni hedef konum belirlenir. Belirlenen bu yeni hedef konum ile aracın durak içindeki mevcut konumu arasında Şekil 60'da görüldüğü gibi duraktan çıkışa uygun bir yol planlanır. Belirlenen yolun takip edilmesini sağlayacak hız ve direksiyon bilgilerinin '/throttle_cmd' ve '/steering_cmd' topic'leri üzerinden kontrolöre iletilip donanım üzerinde gerçeklemesiyle araç yolu takip ederek yola çıkmış olur.



Şekil 60. Duraka çıkış için hareket planlaması.

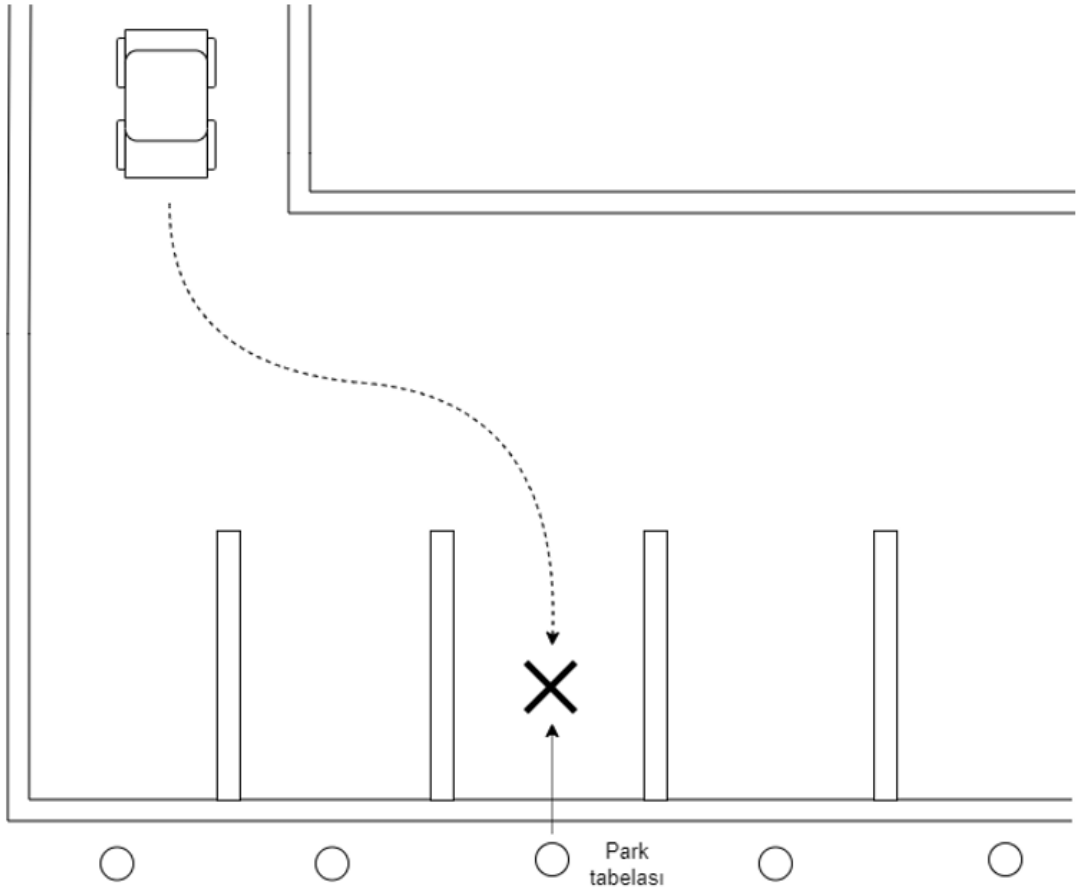
Aracın yaptığı manevralar öncesinde çevreye uyarı olarak şağ-sol işaret lambaları yakılmaktadır. Araç durağı algılayıp giriş manevrasına başlayacağı zaman durağa yanaşacağı uyarısında bulunmak için kontrolör ile '/current_signal' topic'i üzerinden 'signal' düğümüne sağ sinyal bilgisi gönderilerek sağ işaret lambası yakılmaktadır. Durakta yolcu alışverişinin tamamlanıp çıkış manevraları başlayacağı zaman yola çıkılacağı uyarısında bulunmak için kontrolör ile '/current_signal topic'i üzerinden 'signal' düğümüne sol sinyal bilgisi gönderilerek sol işaret lambası yakılmaktadır.

Araç duraktan çıkıştaki hedef konumuna varıp yola çıktıktan sonra, diğer etkenlere bakmaksızın durumunu, Durak durumundan Sürüş durumuna tekrardan getirmektedir. Bu sayede durumlar arası geçiş sağlanmaktadır.

5.4.4. Park Durumu

Park durumu, aracın park alanındaki uygun park bölmelerine park etmesini sağlayacak hareketlerin planlandığı durumdur. Aracın park alanına gelmesiyle çalışmaya başlamaktadır. Durak durumunda olduğu gibi çevre algılama ekipmanları ile aracın konumu ve park alanının haritası belirlenerek uygun park bölmesine park etmeyi sağlayacak hareketler planlanmaktadır.

Park alanına gelindiği iki şekilde tespit edilmektedir. Birincisi şeritlerin bitmesi, ikincisi ise park levhalarının tespitidir. Her iki durum da görüntü işleme ile gerçekleştirilir. Park alanı algılandıktan sonra uygun bir park bölmesine park edilmek üzere hareket planlamasına başlanmaktadır.



Şekil 61. Park için hareket planlaması.

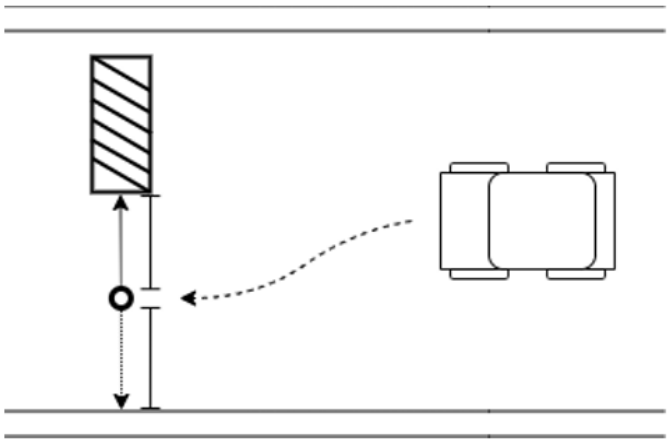
Çevre algılama kısmındaki görüntü işleme ve yapay zeka ile park alanı taranmaktadır. İşaret

levhalarının da incelendiği bu tarama ile engelli park tabelası olmayan ve park etmeye mani olacak bir engel bulunmayan park bölmesi tespit edilir. Tespit edilen park bölmesinin başında bulunan park levhası lidar sensörü ile algılanarak konumu hesaplanır. Şekil 61'de görüldüğü gibi park levhasının park alanına doğru yönde, aracın park bölmesine uygun parkını sağlayacak belirli bir mesafe ilerisi hedef konum olarak belirlenir. Belirlenen hedef konum ile aracın mevcut konumu arasında Şekil 61'de görüldüğü gibi diğer park bölmelerine girmeden hedefe uygun park edecek bir yol planlanır. Aracın planlanan yolu takip etmesini sağlayacak hız ve direksiyon bilgileri '/throttle_cmd' ve '/steering_cmd' topic'leri üzerinden kontrolöre iletilir. Kontrolörün, gelen hız ve direksiyon bilgilerini donanım üzerinde gerçeklemesiyle araç yolu takip etmeye başlar. Park bölmesine yaklaşıldığında planlanan yola ek olarak lidar ve şerit algılama ile park bölmesi ortalanarak daha güveli bir park süreci sağlanmış olur.

Aracın park bölmesine girmesi ile araç durur ve kontrolör ile '/current_signal topic'i üzerinden 'signal' düğümüne fren/stop bilgisi gönderilerek stop lambaları yakılmaktadır. Görevlerin tamamlanması ile araç tekrar başta olduğu gibi Boşta duruma geçerek yeni görev ve komutları beklemektedir.

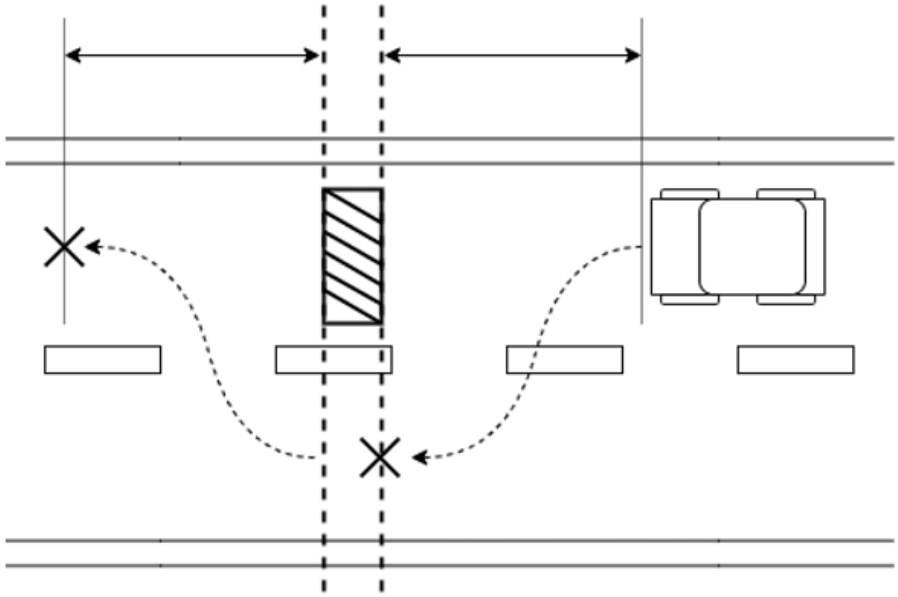
5.4.5. Engelden Kaçınma

Engelden kaçınma algoritması, otonom kara aracının yarışma parkuru üzerinde herhangi bir hareketli/hareketsiz engel ile karşılaşması durumunda aracın engelden kaçabilmesi veya güvenli mesafede fren yapabilmesini sağlamaktadır. Araç üzerindeki çevre algılama ekipmanları ve algoritmalarının engel algılaması ile devreye girmektedir.



Şekil 62. Engeli geçebilmek için hareket planlaması.

Engelden kaçınma algoritmasında lazer tarayıcı verileri temel teşkil etmektedir. Lazer tarayıcı verileri noktalar kümesi olarak okunarak çevre bilgisi toplanır. Aracın güvenli mesafeden engelden kaçabilmesi veya fren yapabilmesi adına güvenli manevra mesafesi 5 metre olarak belirlenmiştir. Dolayısıyla lazer tarayıcıdan gelen veriler, 5 metre içerisindeki noktalar kümesini içerecek şekilde sınırlandırılır. Lazer tarayıcı verileri işlenerek aracın etrafındaki nesnelerin dış hatları belirlenir. Sınırlandırılmış noktalar kümesi içerisinde bir nesne tespit edilmesi halinde ani fren ile yavaşlanır. Eğer mümkünse engelden kaçınarak engeli geçmek üzere hareket planlanır. Engeli geçmek mümkün değilse araç durarak bekleme durumuna geçmektedir.



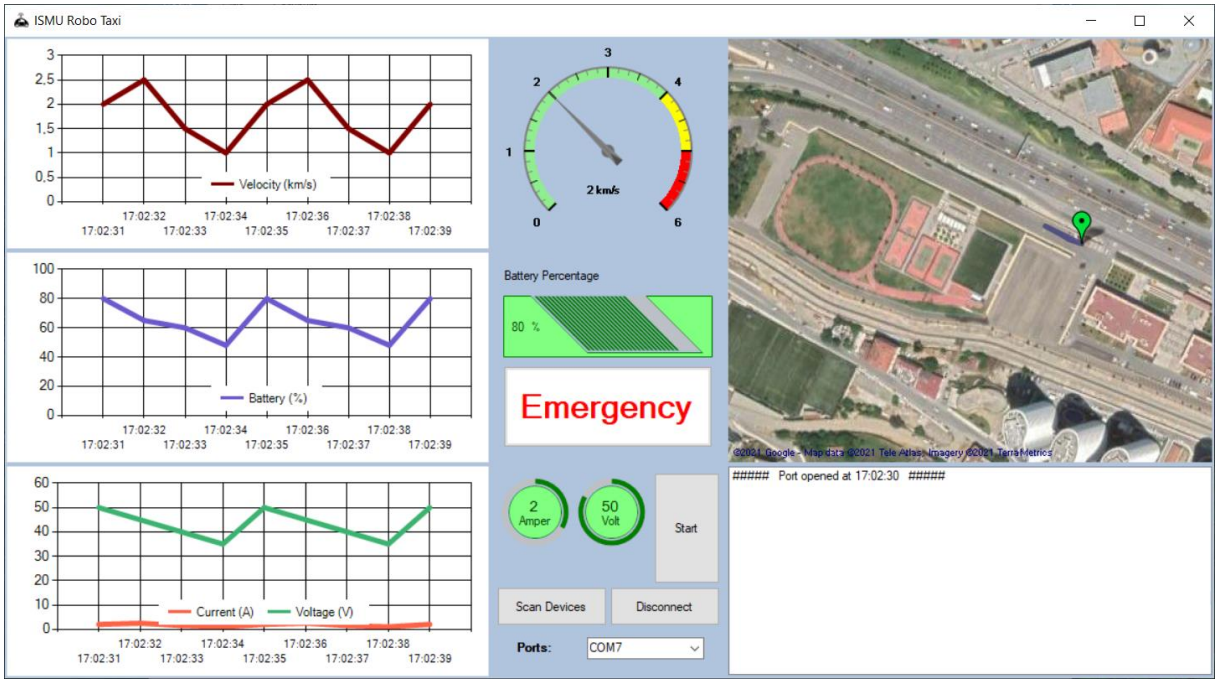
Şekil 63. Öndeki aracı geçme (sollama) için özelleştirilmiş hareket planlaması.

Algılanan engelden kaçınılarak engeli geçmenin mümkün olup olmadığı belirlenirken çevre algılama ekipmanları ve algoritmaları birlikte çalışır. Engelin çevresinde bulunabilecek devamlı şerit, bariyer, geçilemez alan (yol daralması), başka bir engel gibi unsurlardan herhangi biri bulunuyorsa Şekil 62'de görüldüğü gibi bu unsur ile engel arasındaki orta nokta hizalanarak şerit takibindeki davranışa benzer şekilde harekete devam edilir. Engel teşkil eden bu unsurlardan herhangi birinin araca yakınlığının 5 metrenin altına düşmesiyle ani fren yapıp durularak engelin geçilemeyeceğine karar verilir. Engellerin yakınlığı 5 metrenin altına düşmeden engel lazer tarayıcının tarama bölgesinden çıkarsa araç, engelle karşılaşmadan önce gerçekleştirdiği hareket planına devam eder. Eğer engel ile düz yolun sağ şeridinde karşılaşılıp orta nokta hizalaması ile engelin etrafından geçilebiliyorsa Şekil 63'te görüldüğü gibi durak durumunda gerçekleştirilen aynalama metoduna benzer şekilde hedef konumlar belirleyip şerit orta noktası hizalama yönteminin de desteğiyle güvenli şekilde sağ şeride geri dönülebilmektedir. Bu planlama ile araca öndeki aracı geçme (sollama) kabiliyeti kazandırılmak istenmiştir.

5.5. Araç Arayüz Uygulaması

Araç için kullanılmak üzere C# dili ile Windows Forms kullanılarak Windows işletim sisteminde çalışan bir arayüz geliştirilmiştir. Arayüz Şekil 28'de görülmektedir. Bu arayüz ile aracın anlık hız, akü gerilimi ve akımı gözlemlenmektedir. Bunun yanı sıra aracın anlık konumu ve izlediği güzergah harita üzerinden izlenebilir.

Bu arayüz kullanılarak araç otonom sürüşe başlatılabilir. Acil bir durumda ise aracın otonom sürüş yazılımı iptal edilir, aracın gazı kesilir ve frenlemesi yapılır. Böylelikle aracın güvenli bir şekilde durması sağlanmış olur.



Şekil 64. Yer istasyonu arayüzü

5.6. Ekstrem Durumlar

Sensörlerden veri gelmemesi durumları detaylı olarak incelenmiş, aracın bulunduğu duruma göre farklı kararlar vermesi planlanmıştır. Otonom araçta dizayn edilen durumlar kontrol kısmında detaylı olarak ele alınmıştır. Bu durumlar; boşta durum, sürüş durumu, durak durumu, park durumu ve engelden kaçınma durumudur. Araç üzerinde mevcut sensörler; iki adet kamera, GPS, IMU ve lazer tarayıcı sensörleridir. En kötü senaryo tekniği ile tüm durumlar için gerekli minimum sensör verileri ve olasılıkları hesaplanmıştır.

Boşta durum: Bu durumda araç hareketine başlamamıştır. Araç üzerindeki şerit tespiti için bulunan kameranın çalışması zorunludur. Eğer şerit tespiti için kullanılan kamera çalışmıyorsa araç hareketine başlamayacaktır. Şerit tespiti için kullanılan kamera çalışıyorsa, lazer tarayıcıdan veya çevreyi algılamak için kullanılan kameradan ikisinden birisi bozuk olursa araç hareket edebilir. Ancak hem lazer tarayıcı hem de çevreyi algılamak için kullanılan kamera bozuksa araç hareketine başlamayacaktır. IMU ve GPS sensörlerinin bozuk olması aracın boşta durumdaki hareketini engellemeyecektir. Her ikisi birden bozuk olsada araç diğer şartları sağlarsa hareketine başlayabilir.

Sürüş durumu: Bu durumda araç hareket halindedir. Araç üzerindeki şerit tespiti için bulunan kameranın çalışmaması durumunda, aracın hızı düşürülür. Ancak aracın tamamen durdurulmasına ihtiyaç yoktur. Eğer lazer tarayıcı bir engel görmüyorsa araç hareketine düz olarak devam eder. Engelle karşılaşırsa ve çevreyi algılayan kameradan anlamlı bir veri gelmiyorsa araç kendisini durdurur. Ancak lazer tarayıcıdan anlamlı veri geliyorsa ve GPS ile konumunu tespit edebiliyorsa hareket etmeye devam edebilir.

Durak durumu: Araç durak durumuna gelmişse lazer tarayıcı önemlidir. Lazer tarayıcı sayesinde direği tespit ederek rota çizmektedir. Eğer lazer tarayıcı çalışmıyorsa, şerit takibi yapan kamera kontrol edilir. Şerit takibi yapan kamera çalışırsa, araç hareketine devam edebilir. Ancak şerit takibi yapan kamerada da arıza meydana gelirse o zaman araç duracaktır. Eğer durak durumu başladığında IMU, GPS ve lazer tarayıcı çalışıyorsa ve durak durumu biraz ilerledikten sonra lazer tarayıcı bozulmuşsa, IMU ve GPS değerleri ile hareketine devam edebilir. Ancak IMU ve GPS

sensörleri de bozulursa, araç hareket etmeyecektir.

Park durumu: Araç park etme durumuna gelmişse, şerit takibi için kullanılan kameranın çalışması zorunlu değildir. Şerit takibi için kullanılan kamera, park durumunda parkın daha iyi şekilde gerçekleştirilmesi için kullanılmaktadır. Şerit takibi için kullanılan kamera çalışmasa da, araç hareketine devam edebilir. Bu durumda çevreyi algılamak için kullanılan kameranın çalışması gerekmektedir. Çünkü çevre algılamak için kullanılan kamera, trafik işaretlerini de tespit etmektedir. Eğer park yapmak için gerekli park edilebilir levhasını tespit etmişse ve kamera levhayı tespit ettikten sonra bozulmuşsa ve araç üzerinde lazer tarayıcı sensörü düzgün çalışıyorsa araç hareketine devam edebilir. Çünkü levhanın mevcut konumu yerel olarak tespit edilmiştir Ancak kamera park edilebilir levhasını tespit etmeden önce bozulmuşsa, o zaman araç duracaktır. Hareketine devam edemez. Hem çevre algılama için kullanılan kamera hemde lazer tarayıcı bozuksa araç hareket etmez. IMU ve GPS sensörlerinin bozuk olması, park durumunda aracın hareket etmesine engel değildir. Çünkü tehlike meydana gelmez. GPS ve IMU sensörlerinin eksikliği sonucunda çevreye zarar verilmez.

Engelden kaçınma durumu: Araç engelden kaçınma durumunun içerisindeyse, bu bilgiye iki şekilde ulaşmış olabilir. İlk yol, çevre algılamak için kullanılan kamera, ikinci yol ise lazer tarayıcı verisi olabilir. Ancak engelden kaçınma aşamasına girildikten sonra da engelin tespit edilebilmesi gerekmektedir. Engel sabit ise tespit zorunlu olmamakla beraber, engelin sabit veya hareketli olduğu bilgisi olmadığı için engelin sürekli tespit edilmeye devam edilmesi gerekir. Bu yüzden lazer tarayıcı ya da çevre algılayan kameradan en az birinin çalışması gerekir. Her ikiside çalışmıyorsa, o zaman araç kendini emniyete alarak çalışmayı durdurur. Engelden kaçınma için şerit takip kamerası önemli olmakla beraber zorunlu değildir. Şerit takip kamerası bozulsada algoritmik olarak hareketine devam edebilir. Ancak yola tekrardan girdiği anda şerit takip kamerası hala bozuksa o zaman araç çalışmayı durdurmalıdır. GPS ve IMU sensörleri engelden kaçma durumunda kullanılır. Ancak engelden kaçınma durumunda asli sensörler değildir. Bu sensörlerin olmaları hareketi kolaylaştırırken, olmadıkları durumda da araç hareket edebilir.

6. Özgün Bileşenler

Bu bölümde ön tasarım raporunda da belirtilen aracın tasarım ve/veya yazılım bakımından özgün yönleri değerlendirilmelidir. Bu yönlerden varsa son tasarımda bulunmayanların neden gerçekleştirilmediği anlatılmalıdır.

6.1. Aracın Yapısı

Araç aynı anda iki kişinin binebileceği boyutlarda olup, arazi aracı olarak kullanılabilecek kapasitede ve güçtedir. Bu yönüyle yarışmaya katılan diğer araçlardan farklı olarak, istenildiğinde takım üyeleri normal bir araç gibi kullanabilmektedir.

6.2. Tasarım

Aracın tasarımı tamamen özgün olup BNB bisiklete aittir. Aracın laboratuvarımıza hibe edilmesi sonrasında üzerinde yapılan mekanik ve elektronik sistemler laboratuvarımız kapsamında gerçekleştirilmiştir. Bu sayede araç otonom olarak sürüşe elverişli hale getirilmiştir.

6.3. Yazılım

Kontrol ve simülasyon yazılımı tamamen takımımızın çalışması sonucu üretilmiştir. Bu yazılımlar C++ ve Python dillerinde yazılmış olup, ROS kurulu Ubuntu 18.04 işletim sistemi üzerinde çalışmaktadır.

6.4. Tasarımda Değişen Şeyler

Ön tasarım sırasında değiştirilenler bu bölümde sunulmuştur.

6.4.1. Işıklar

Önceki yarışmalarda sadece fren lambası bulunmaktaydı. Buna ilaveten, bu seneki yarışma için araca sinyal lambaları eklenmiştir. Ayrıca ileri-geri vites kontrolü de sağlanmıştır.. Bu sinyallerin düşük seviye kontrolleri, aracın gaz ve frenini kontrol eden Arduino Uno üzerinde sağlanmıştır. Sonuç olarak bir Arduino Uno direksiyonu düşük seviye kontrolünü sağlarken, diğeri de gaz, fren, ileri-geri vites ve sinyal lambalarının kontrolünden sorumludur.

6.4.2. Elektronik

Araca eklenen bu yeni özellikler olan yön sinyalleri ve ileri geri-vites değişimi için Arduino Uno ile kullanılmak üzere yeni devre tasarlanmış ve uygulanmıştır.

Araç, istenen görevleri yerine getirmek üzere donanım ve sensörlerle donatılarak ackermann direksiyon geometrisine sahip şekilde takım tarafından dizayn edilmiştir. Donanım ve sensörler, görevler göz önünde bulundurularak optimum fayda verecek şekilde üyeler tarafından tercih edilmiş ve konumlandırılmıştır. Dolayısıyla araç yarışma doğrultusunda özgün olarak ortaya konulmuştur.

Son tasarımda araç üzerine ön tasarımda bulunmayan uyarı (sinyal) lambaları eklenmiştir. Uyarı lambalarının hareket planlama çıktıları ve aracın anlık hareketlerine göre yanması için takım üyeleri tarafından özgün bir sinyalizasyon algoritması geliştirilmiştir.

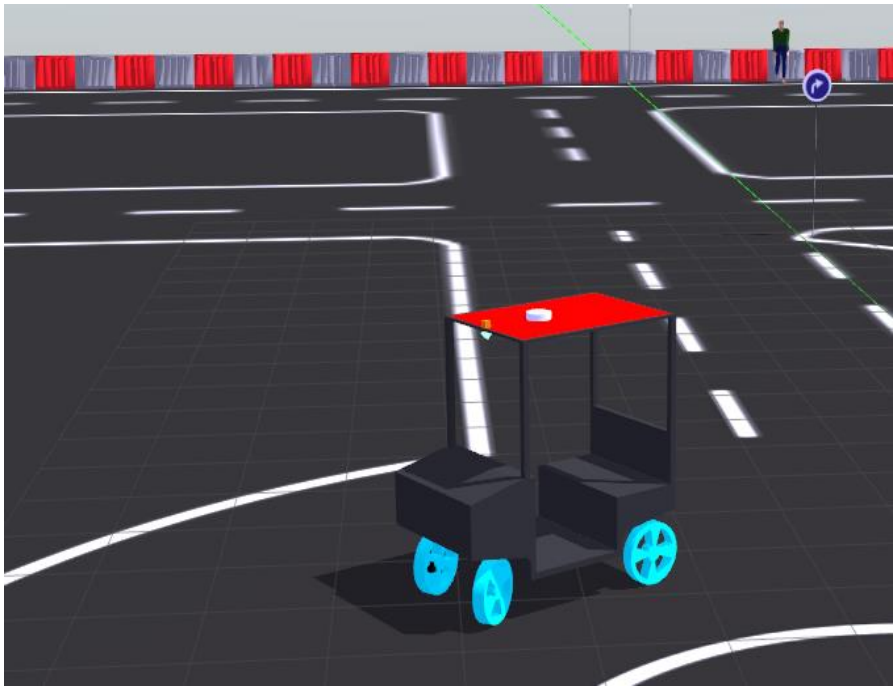
Yarışmada belirlenen görevleri yerine getirmek üzere 5.2. bölümde sunulan hareket planlama yöntemleri takım üyeleri tarafından geliştirilmiş ve algoritmaya dökülerek yazılım üzerinde gerçeklenmiştir. Araç kinematiğine ve ackermann modele uygun kontrol birimi üyeler tarafından dizayn edilerek yazılım ortamına aktarılmıştır. Tüm bunlar hareket planlama ve kontrol birimlerini özgün kılmaktadır.

Yarışma ortamının, yarışma gereksinimleri doğrultusunda gerçekçi biçimde canlandırılması için takım tarafından fizik motoruna sahip Gazebo simülasyon ortamında özgün bir simülasyon oluşturulmuştur. Simülasyon ortamında, yarışma şartnamesinde belirtilen parkur takım üyeleri

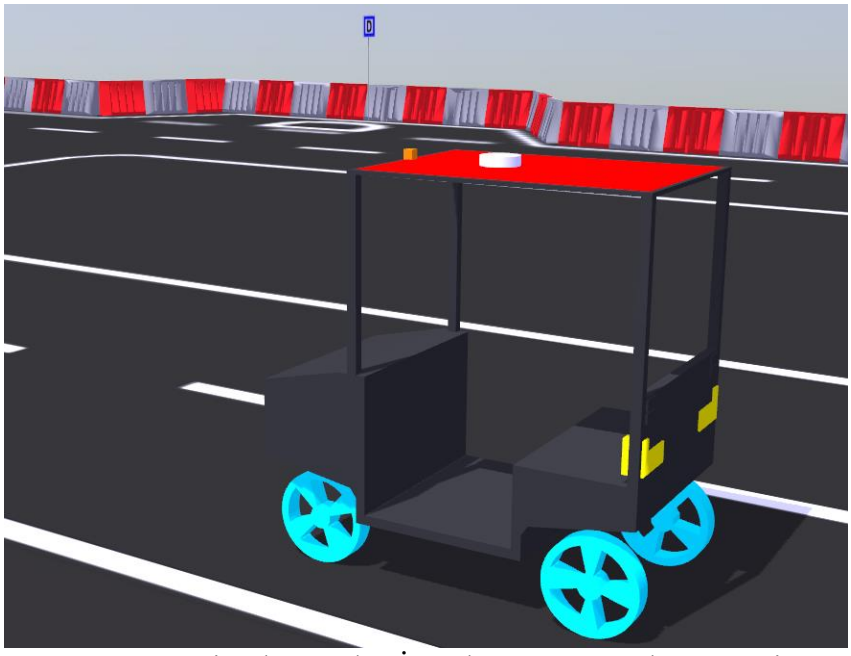
tarafından oluşturulmuş, trafik ışıkları ve trafik levhaları tasarlanmıştır. Bu doğrultuda araç üzerinde testler gerçekleştirilerek sistemsel iyileştirme yapabilmeyi sağlayan özgün bir simülasyon elde edilmiştir.

7. Test

İvmelenme, frenleme ve yönlendirme testlerinin aşağıda belirtilen senaryolar ile simülasyon ortamında ve gerçek araç üzerinde aynı kontrolcü girişleri ile gerçekleştirilmesi sağlanmıştır. Araç testleri simülasyon ve gerçek ortam olmak üzere toplamda iki şekilde gerçekleştirilmiştir. Testlerin ilk aşamasında rapor şablonunda istenen testler Şekil 65'te görüldüğü üzere Gazebo simülasyon ortamında modellenen araç ile gerçekleştirilmiştir. Simülasyon ortamında Şekil 66'da görüldüğü üzere araç hareket halindeyken IMU verileri rosbag ile kaydedilmiştir. Bu verilerden hız ve konum bilgileri hesaplanarak kaydedilmiştir.



Şekil 65. Simülasyondan Yönlendirme (Direksiyon) Testi



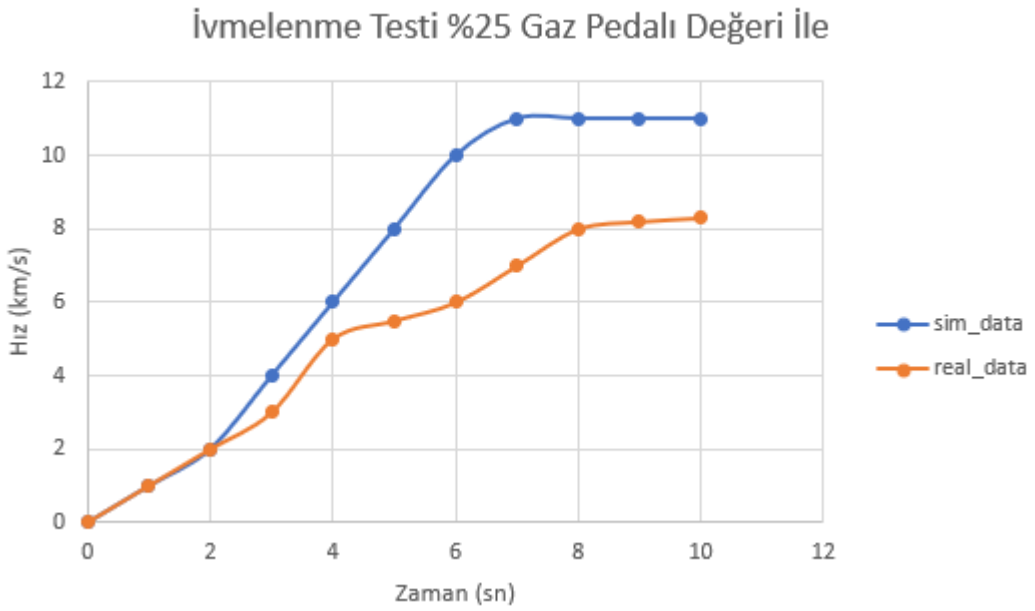
Şekil 66. Simülasyondan İvmelenme ve Frenleme Testi

Testlerin ikinci aşaması olan gerçek hayat testlerinde Şekil 67'de görülen, otonom araç üzerindeki AHRS verileri rosbag ile kaydedilmiştir. Bu testlerde kaydedilen rosbag verileri işlenerek konum ve hız bilgileri elde edilmiştir. Bu verilerden yararlanılarak Excel ortamında grafikler elde edilmiştir.

Şekil 67. Gerçek ortamdan frenleme test görüntüsü

Şekil 68. Gerçek ortamdan ivmelenme test görüntüsü

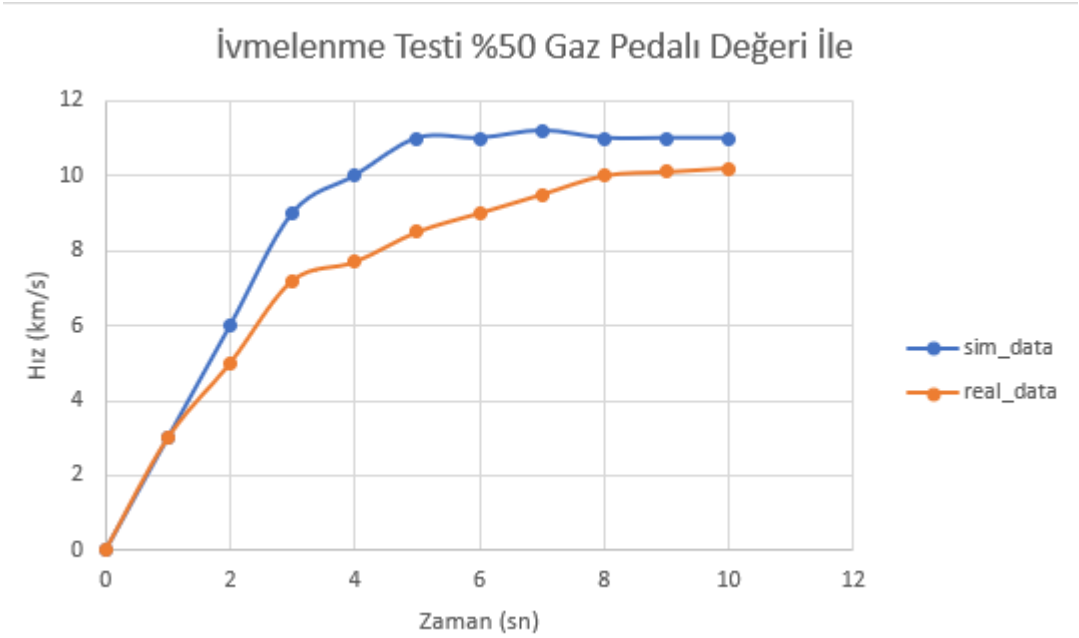
7.1. İvmelenme



Şekil 69. %25 gaz pedalı ile ivmelenme testi

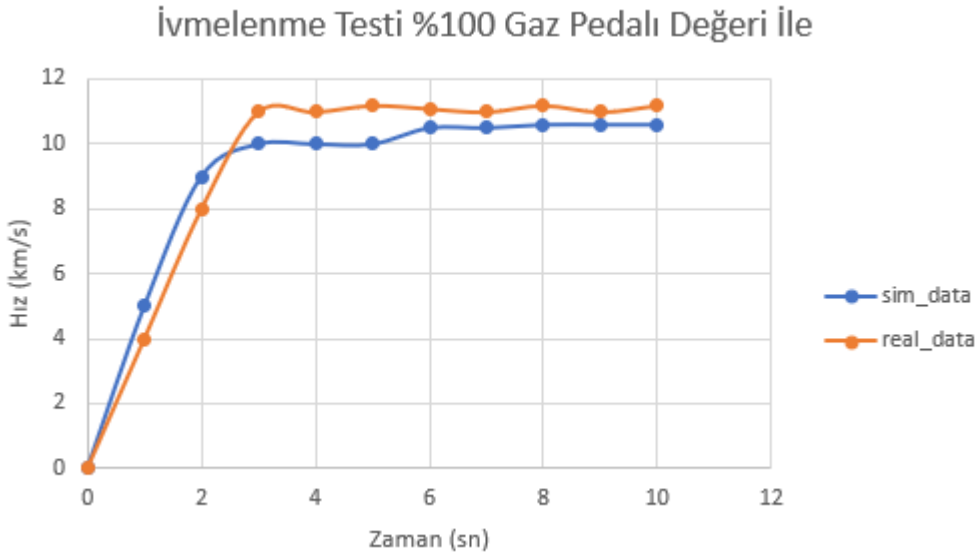
Şekil 69'da %25 gaz pedalı ile simülasyon ve gerçek ortamda yapılan testin çıktıları verilmiştir. Şekil 69'da görüldüğü üzere simülasyon ortamında araç beklenen hız değerine lineer şekilde

ulaşırken gerçek hayatta ulaşmadığı görülmüştür bunun muhtemel sebepleri; gerçek hayattaki sürtünme yüzeylerinin çokluğu, araç ağırlığının yer çekimine bağlı olarak hıza etki etmesi, motorun istenilen verimde çalışmaması olarak sıralanabilir.



Şekil 70. %50 gaz pedalı ile ivmelenme testi

Şekil 70'te %50 gaz pedalı ile simülasyon ve gerçek ortamda yapılan testin çıktıları verilmiştir. Şekil 70'te görüldüğü üzere simülasyon ortamında araç beklenen hız değerine lineer şekilde ulaşırken gerçek hayatta ulaşmadığı görülmüştür bunun muhtemel sebepleri; gerçek hayattaki sürtünme yüzeylerinin çokluğu, araç ağırlığının yer çekimine bağlı olarak hıza etki etmesi, motorun istenilen verimde çalışmaması olarak sıralanabilir.



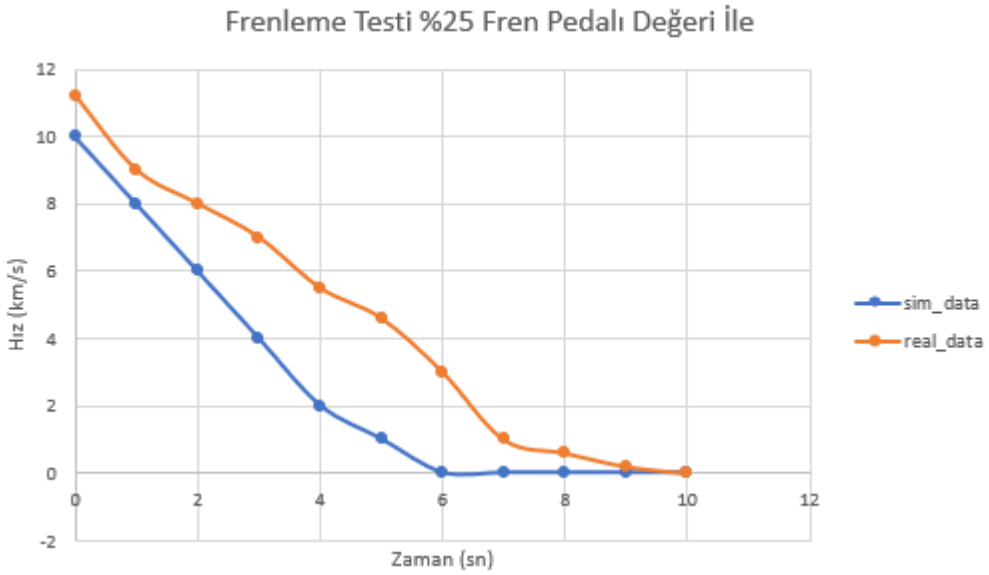
Şekil 71. %100 gaz pedalı ile ivmelenme testi

Şekil 71'de %100 gaz pedalı ile simülasyon ve gerçek ortamda yapılan testin çıktıları verilmiştir. Şekil 70'te görüldüğü üzere simülasyon ortamında araç beklenen hız değerine lineer şekilde ulaşırken gerçek hayatta bu değeri aştığı görülmüştür bunun muhtemel sebepleri; motorun tam verimle çalışmaması, voltaj dalgalanması, sensör hassasiyet yeterliği olarak sıralanabilir.

Tablo 6. Aracın İvmelenme Test Tablosu

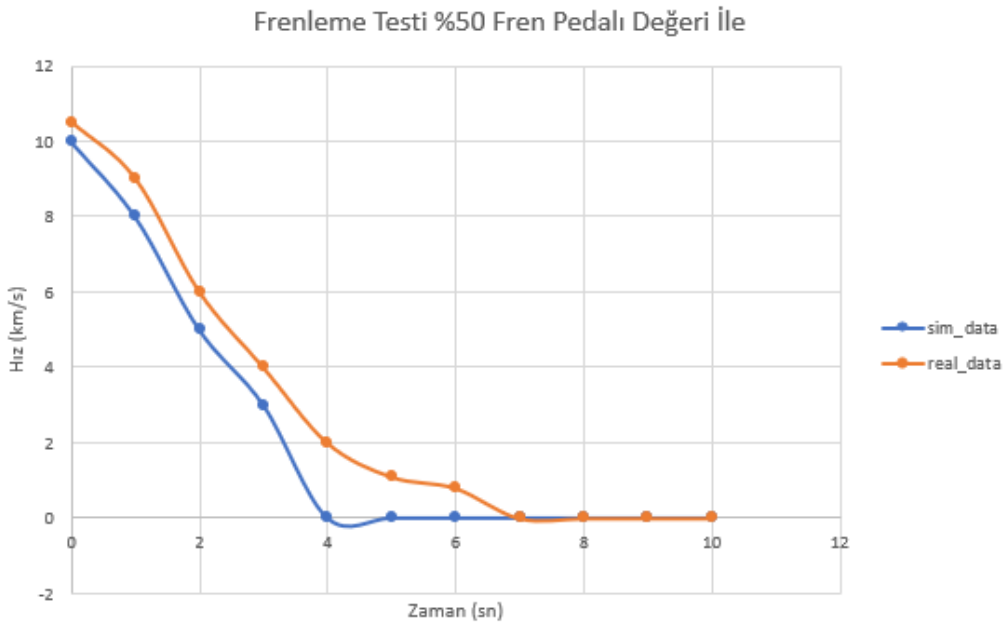
Gaz Pedalı Değeri	Araç Hızı	Sonuç
%25	0 km/s -- >10 km/s	Şekil 69. Aracın sabit durumdan %25 gaz pedalı giriş değeri ile 10 km/s hıza çıkış eğrisi
%50	0 km/s -- >10 km/s	Şekil 70. Aracın sabit durumdan %50 gaz pedalı giriş değeri ile 10 km/s hıza çıkış eğrisi
%100	0 km/s -- >10 km/s	Şekil 71. Aracın sabit durumdan %100 gaz pedalı giriş değeri ile 10 km/s hıza çıkış eğrisi

7.2. Frenleme



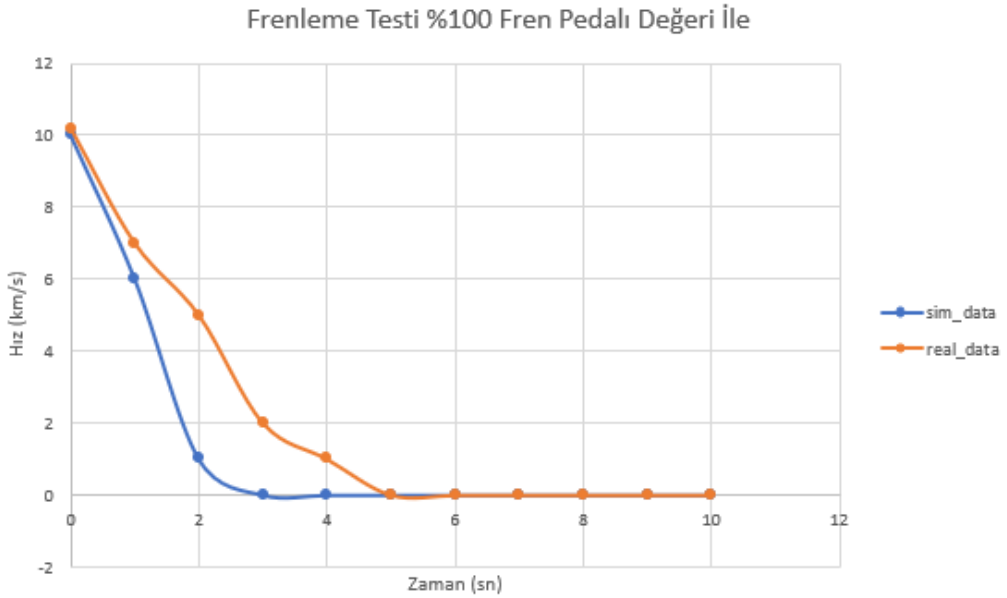
Şekil 72. %25 fren pedalı ile frenleme testi

Şekil 72'de %25 fren pedalı ile simülasyon ve gerçek ortamda yapılan testin çıktıları verilmiştir. Şekil 72'de görüldüğü üzere simülasyon ortamında araç beklenen frenleme değerine lineer şekilde ulaşırken gerçek hayatta ulaşmadığı görülmüştür bunun muhtemel sebepleri; gerçek hayattaki sürtünme yüzeylerinin çokluğu, araç ağırlığının yer çekimine bağlı olarak hıza etki etmesi, motorun istenilen verimde çalışmaması olarak sıralanabilir. Aracın gerçek hayatta frenlemesinin lineer olmadığı görülmektedir.



Şekil 73. %50 fren pedalı ile frenleme testi

Şekil 73'te %50 fren pedalı ile simülasyon ve gerçek ortamda yapılan testin çıktıları verilmiştir. Şekil 73'te görüldüğü üzere simülasyon ortamında araç beklenen frenleme değerine lineer şekilde ulaştığı ve gerçek hayatta frenlemenin simülasyon değerlerine yakınsadığı görülmüştür bunun muhtemel sebepleri; gerçek hayattaki sürtünme yüzeylerinin çokluğu, araç ağırlığının yer çekimine bağlı olarak hıza etki etmesi, motorun istenilen verimde çalışmaması olarak sıralanabilir. Aracın gerçek hayatta frenlemesinin lineer olmadığı görülmektedir. Araç gerçek hayatta sıfır hız değerine simülasyon ortamından daha yavaş ulaştığı gözlemlenmiştir.



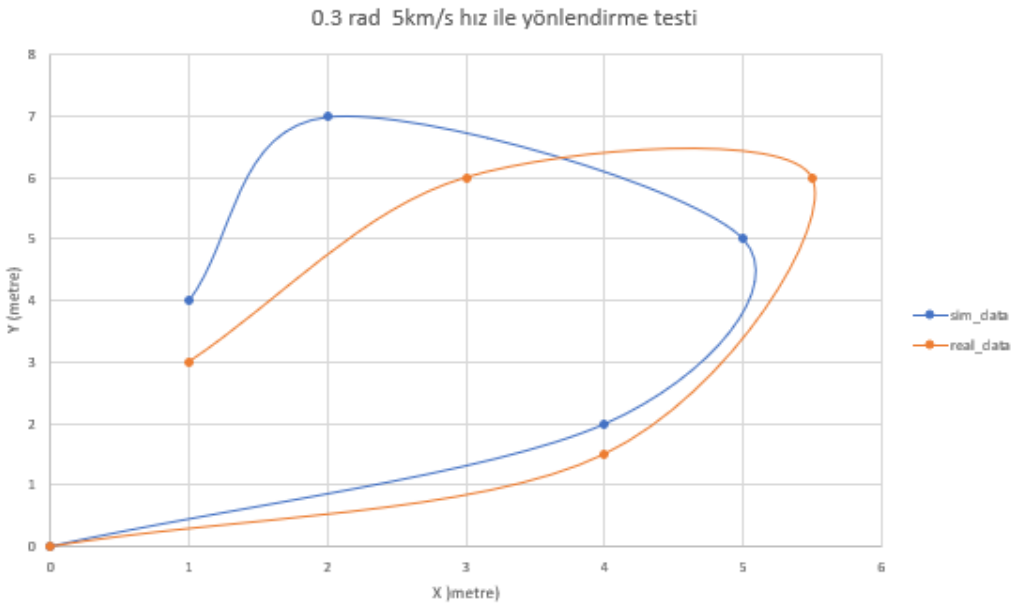
Şekil 74. %100 fren pedalı ile frenleme testi

Şekil 74'te %100 fren pedalı ile simülasyon ve gerçek ortamda yapılan testin çıktıları verilmiştir. Şekil 74'te görüldüğü üzere simülasyon ortamında araç beklenen frenleme değerine lineer şekilde ulaştığı fakat gerçek hayatta frenlemenin simülasyon değerlerine yakınsamadığı görülmüştür bunun muhtemel sebepleri; gerçek hayattaki sürtünme yüzeylerinin çokluğu, araç ağırlığının yer çekimine bağlı olarak hıza etki etmesi, motorun istenilen verimde çalışmaması olarak sıralanabilir. Aracın gerçek hayatta frenlemesinin lineer olmadığı görülmektedir. Araç gerçek hayatta sıfır hız değerine simülasyon ortamından daha yavaş ulaştığı gözlemlenmiştir. Bu değerler aracın performansı için yeterlidir.

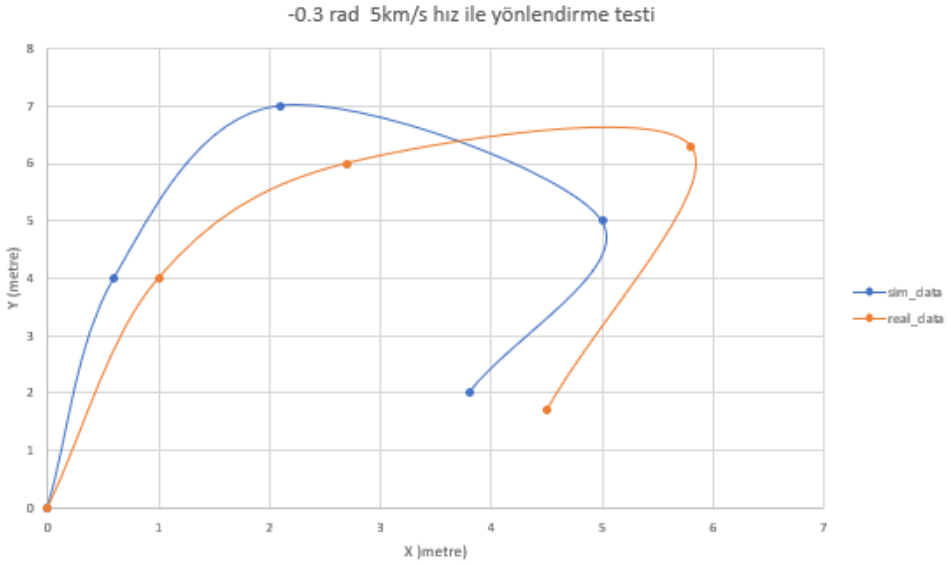
Tablo 7. Aracın Frenlme Test Tablosu

Fren Pedalı Değeri	Araç Hızı	Sonuç
%25	10 km/s -- > 0 km/s	Şekil 72. Aracın 10 km/s hızında ilerlerken %25 fren pedalı girişi ile aracın tamamen durdurulma eğrisi
%50	10 km/s -- > 0 km/s	Şekil 73. Aracın 10 km/s hızında ilerlerken %50 fren pedalı girişi ile aracın tamamen durdurulma eğrisi
%100	10 km/s -- > 0 km/s	Şekil 74. Aracın 10 km/s hızında ilerlerken %100 fren pedalı girişi ile aracın tamamen durdurulma eğrisi

7.3. Yönlendirme



Şekil 75. 0.3 rad açı değerinde 5km/s hız ile frenleme testi



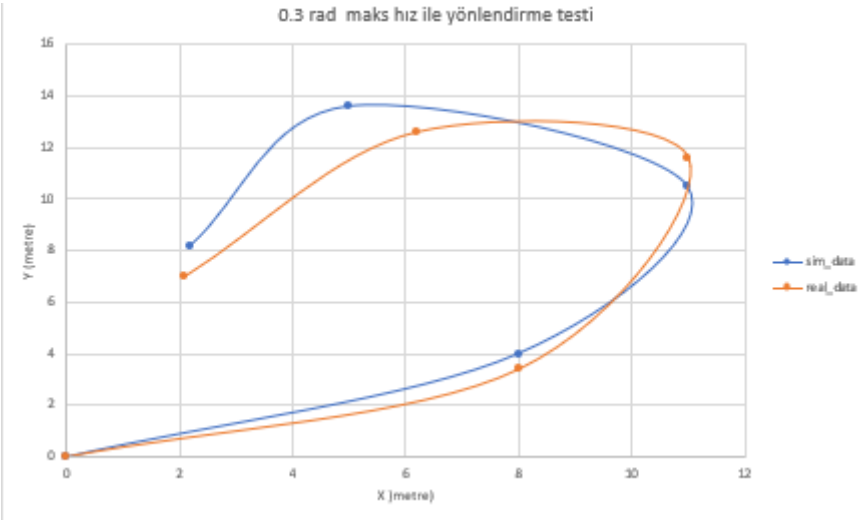
Şekil 76. -0.3 rad açı değerinde 5km/s hız ile frenleme testi

Tablo 8. Aracın ±0.3 radyan ve 5km/s değerleri için Yönlendirme Test Tablosu

Yönlendirme (Direksiyon) Sistemi Açı Değeri	Araç Hızı	Sonuç
0.3 rad	5 km/s	**Şekil 75.** Aracın 5 km/s hız ile sürüşü esnasında yönlendirme girişi 0.3 rad değeri ile aracın yatay yer değişiminin x-y ekseninde grafiğinin oluşturulması
- 0.3 rad	5 km/s	**Şekil 76.** Aracın 5 km/s hız ile sürüşü esnasında yönlendirme girişi -0.3 rad değeri ile aracın yatay yer değişiminin x-y ekseninde grafiğinin oluşturulması

Tablo 9. Aracın maksimum açı ve 5km/s değerleri için Yönlendirme Test Tablosu

Yönlendirme (Direksiyon) Sistemi Açı Değeri	Araç Hızı	Sonuç
Yönlendirme (Direksiyon) Sistemi Maksimum Açı Değeri	5 km/s	Aracın 5 km/s hız ile sürüşü esnasında aracın maksimum yönlendirme girişi değeri ile aracın yatay yer değişiminin x-y ekseninde grafiğinin oluşturulması
Yönlendirme (Direksiyon) Sistemi Maksimum Açı Değeri (Ters Yön)	5 km/s	Aracın 5 km/s hız ile sürüşü esnasında aracın maksimum yönlendirme girişi değeri (ters değeri) ile aracın yatay yer değişiminin x-y ekseninde grafiğinin oluşturulması



Şekil 79. -0.3 rad açı değerinde maks hız ile frenleme testi

Şekil 80. -0.3 rad açı değerinde maks hız ile frenleme testi

Tablo 10. Aracın maksimum açı ve hız değeri için Yönlendirme Test Tablosu

Yönlendirme (Direksiyon) Sistemi Açı Değeri	Araç Hızı	Sonuç
0.3 rad	Maksimum araç hızı	**Şekil 79.** Araç maksimum hızında hareket halinde iken 0.3 rad direksiyon açı değeri için x-y ekseninde konum değişimi grafiğinin oluşturulması
- 0.3 rad	Maksimum araç hızı	**Şekil 80.** Araç maksimum hızında hareket halinde iken -0.3 rad direksiyon açı değeri için x-y ekseninde konum değişimi grafiğinin oluşturulması

Araç simülasyon ortamında ideale yakın çalıştığı için simülasyon çıktılarının daha yüksek doğrulukta olduğu görülmektedir. Konum bilgilerinin uyuşmazlığının nedenleri; AHRS sensor hassasiyeti, tekerlek çapının simülasyon ortamından farklı olması, tekerlek açısının simülasyon ortamına göre daha az doğrulukta verilmesi olarak sıralanabilir.

7.4. Hareket Planlama Yöntemleri

Yarışmada verilen görevleri yerine getirmek üzere tasarlanan her bir hareket planlama yöntemi farklı açı ve konumlandırmalara sahip 10 farklı senaryo ve her bir senaryo için 10 kez tekrarlama ile test edilmiştir. Testlerde başarı, Tablo 11'de yer alan başarı ölçütleri doğrultusunda değerlendirilerek aynı tablo üzerinde sunulmuştur.

Tablo 11. Hareket planlama yöntemleri test sonuçları.

Test Edilen Planlama Yöntemi	**Başarı Ölçütü**	**Başarı Oranı**
Şerit Takibi	Sağdaki kesiksiz, soldaki kesikli çizgilerin ve şerit daralmalarının üzerinden geçilmeden seyir etme.	%94
Dönüşler	Dönüşleri, bulunulan şeritten çıkmadan gerçekleştirme.	%92

Durak Cebine Girme-Çıkma	Kesiksiz çizgiler üzerinden geçmeden durağın uzunlamasına eksenine maksimum 25 derecelik bir açıyla yanaşma.	%95
Park Alanına Park Etme	Park alanındaki kesiksiz çizgiler üzerinden geçmeden park bölmesinin uzunlamasına eksenine maksimum 25 derecelik açıyla park etme.	%90
Engelden Kaçınma	Engelle karşılaşılması halinde engelle çarpışmadan durma veya engeli geçme.	%96

8. Referanslar

[1] NVIDIA.DEVELOPER "Nvidia Jetson Xavier NX Görseli''
https://developer.nvidia.com/embedded/jetson-xavier-nx-devkit
Son erişim tarihi: 6 Haziran 2022

[2] NVIDIA "Model Özellik Tablosu''
https://www.nvidia.com/tr-tr/autonomous-machines/embedded-systems/jetson-xavier-nx/
Son erişim tarihi: 4 Haziran 2022

[3] Stereo Labs. "ZED Camera and SDK Overview",
https://cdn.stereolabs.com/assets/datasheets/zed-camera-datasheet.pdf
Son erişim tarihi: 31 Mart 2022

[4] Stereo Labs. "ZED Camera Görseli"
https://www.stereolabs.com/zed-2/
Son erişim tarihi: 29 Mayıs 2022

[5] Logitech "Web Camera Görseli"
https://www.logitech.com/tr-tr/products/webcams/c920-pro-hd-webcam.960-001055.html
son erişim tarihi: 11 Haziran 2022

[6] RPLİDAR DATASHEET
https://www.mouser.cn/datasheet/2/744/LD601_SLAMTEC_rplidar_datasheet_S1_v1.0_en-1578380.pdf
Son erişim tarihi: 28 Mayıs 2022

[7] Github Inc. " gps_umd"
https://github.com/swri-robotics/gps_umd
Son erişim tarihi: 31 Mart 2022

[8] Red Shift Labs. "UM7 DATASHEET". https://redshiftlabs.com.au/wp-content/uploads/2018/02/um7_datasheet_v1- 6_10.1.2016.pdf
Son erişim tarihi : 31 Mart 2022

[9] ROS.org " Um7 Orientation Sensor"
http://wiki.ros.org/um7
Son erişim tarihi: 31 Mart 2022

[10] Pololu Corp. "Um7 Orientation Sensor Görseli"
https://www.pololu.com/product/2741
Son erişim tarihi: 30 Mayıs 2022

[11] Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. 2020. "YOLOv4: Optimal speed and accuracy of object detection", arXiv preprint arXiv:2004.10934.

[12] Github Inc. "Bjelonic, M. 2018.YOLO ROS: Real-Time Object Detection for ROS"
https://github.com/leggedrobotics/darknet_ros
Son erişim tarihi: 25 Mart 2022

[13] Ribeiro, M. I. (2004). Kalman and extended kalman filters: Concept, derivation and properties. Institute for Systems and Robotics, 43, 46.

[14] MATLAB Tech Talks. "Understanding Kalman Filters, Part 3: Optimal State Estimator".
https://www.youtube.com/watch?v=ul3u2yLPwU0
Son erişim tarihi: 31 Mart 2022.

[15] Olubiwe, M., Uzoechi, L. O., & Uchegbu, V. C. (2016). Improved electrical discharge machine (EDM) servomechanism controller for machining micro pits. International Journal of Engineering Research, 5(04).