

Araba Benzeri Robotun Otonom Sürüş Benzetimi

Martha Jeanneth Sanabria Trujillo

YÜKSEK LİSANS TEZİ

Elektrik – Elektronik Mühendisliği Anabilim Dalı

Mayıs 2017

Autonomous Driving Simulation of a Car-Like Robot

Martha Jeanneth Sanabria Trujillo

MASTER OF SCIENCE THESIS

Department of Electrical and Electronics Engineering

May 2017

Araba Benzeri Robotun Otonom Sürüş Benzetimi

Martha Jeanneth Sanabria Trujillo

Eskişehir Osmangazi Üniversitesi

Fen Bilimleri Enstitüsü

Lisansüstü Yönetmeliği Uyarınca

Elektrik – Elektronik Mühendisliği Anabilim Dalı

Kontrol ve Kumanda Sistemleri Bilim Dalında

YÜKSEK LİSANS TEZİ

Olarak Hazırlanmıştır

Danışman: Prof. Dr. Osman Parlaktuna

Mayıs 2017

ONAY

Elektrik – Elektronik Mühendisliđi Anabilim Dalı Yüksek Lisans öđrencisi Martha Jeanneth Sanabria Trujillo'nun YÜKSEK LİSANS tezi olarak hazırladıđı “Araba Benzeri Robotun Otonom Sürüş Benzetimi” başlıklı bu çalıřma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek oybirliđi ile kabul edilmiřtir.

Danıřman : Prof. Dr. Osman Parlaktuna

İkinci Danıřman :-

Yüksek Lisans Tez Savunma Jürisi:

Üye : Prof. Dr. Osman Parlaktuna

Üye : Yard. Doç. Dr. Hakan Korul

Üye : Yard. Doç. Dr. Burak Kaleci

Üye :-

Üye :-

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih
ve sayılı kararıyla onaylanmıřtır.

Prof. Dr. Hürriyet ERŐAHAN
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Prof. Dr. Osman Parlaktuna danışmanlığında hazırlamış olduğum “ Araba Benzeri Robotun Otonom Sürüş Benzetimi” başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 09/05/2017

Martha Jeanneth Sanabria Trujillo

İmza

ÖZET

Otonom sürüş (bir sürücü tarafından yönlendirilmeye ihtiyacı olmayan) taşıma yapan bütün araçların geleceğini oluşturmaktadır ve araç üreticileri, taşıma servisleri ve teknoloji şirketleri için stratejik öneme sahip bir konu olmuştur.. Ancak bir aracın otonom olarak hareket edebilmesi için aracın öncelikle tam olarak nerede olduğunu, etrafındaki ortamda bulunan cisimleri ve güvenli bir yolculuk gerçekleştirmek için en uygun yolu hesaplaması gerekmektedir. Bu işlemde kameralar, radarlar ve lazerler kullanılarak uygun bilgiler elde edilebilir.

Bilgisayarda gerçek ortamı yapılandırmak için simülasyon programları önemli araçlardır, simülasyon sayesinde uygulamaya geçilmeden önce ortamda değişiklikler yapılabilir. Otonom sürüş için gereken araçların maliyeti çok yüksek olduğu için ve ortamı oluşturmak zor olduğu için simülasyon programları büyük kolaylık sağlamaktadır.

Bu projede gerçekte var olan küçük ölçekli araba benzeri robotun ROS ve Gazebo simülatör araçları kullanılarak simülasyonunun oluşturulması üzerine odaklanılmıştır ve projede otonom aracın karar vererek oluşturduğu yolu takip etmesi sağlanmıştır. Simülasyon aracı kullanılarak gerçek ortamda bulunan tünel, engel ve çalışma alanı gibi özellikler ortama eklenebilir.

Anahtar Kelimeler: otonom sürüş yarışması, simülasyon ortamı, ackermann direksiyonu

SUMMARY

Autonomous driving (the one that does not need to be guided by a driver) is shaping up as the future of mobility, and has become a strategic aspect for car manufacturers, transportation services and technology companies. But to be able to drive autonomously the vehicle needs to know exactly where it is, to recognize the objects of its environment and to constantly calculate the optimal route to provide a safe trip, made by cameras, radars and lasers obtaining the most possible information, essential for safety.

The simulation on its part is an important tool that allows reconstructing a real environment in a computer, in which it can make changes before implementing them in real surroundings; for autonomous driving it can be a tool that brings the general public closer to expensive systems or requiring a great logistics.

This project focuses on the simulation of small-scale car-like robot, whose model is based on the existing real model using ROS and Gazebo simulator, which can autonomously navigate on a track making decisions about the route to follow. The simulated model adds features to the track that could be found in a real environment, such as a tunnel, obstacle and work zone.

Keywords: autonomous driving competition, simulation environment, Ackermann steering

TEŐEKKÜR

Tezim boyunca hiçbir desteęini benden esirgemeyen danıőmanım Prof. Dr. Osman Parlaktuna'ya ve Erasmus eęitimimde bana destek olan Leiria Üniversitesi Politeknik Enstitüsü Teknoloji ve Yönetim okulunda danıőmanlıęımı yapan Dr. Hugo Filipe Costelha de Castro ve ikinci danıőmanım olan Luis Manuel Conde Bento'ya teőekkür ederim.

Ayrıca hayatım boyunca maddi ve manevi hiçbir desteęini benden esirgemeyen deęerli aileme ve tez çalıőmam boyunca yanımda olan arkadaşlarıma teőekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xiii
SİMGELER VE KISALTMALAR DİZİNİ	xiv
1. GİRİŞ VE AMAÇ	1
1.1. Motivasyon	1
1.2. Amaçlar	4
1.3. Raporun Yapısı	5
2. LİTERATÜR ARAŞTIRMASI	6
2.1. Ulusal Robot Festivali –Portekiz	6
2.2. DARPA Büyük Mücadelesi (Grand Challenge - DRC)	9
2.3. AVC Sparkfun	11
2.4. Audi Otonom Sürüş Kupası	12
3. MATERYAL VE YÖNTEM	15
3.1. Fiziksel Nitelikler	16
3.2. Autorally Nodes (Düğüm) Fonksiyonları	20
3.3. İşlem Ayrıntıları	23
4. BULGULAR VE TARTIŞMA	26
4.1. Robot Tabanı	28
4.1.1 Donanım	30
4.1.2 Mekanik sürme sistemi	31
4.1.3 Düşük seviye kontrol sistemi	34

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
4.1.4 Düşük seviye kontrol yazılımı	36
4.2. Üst Düzey Denetim.....	42
4.2.1 Görüş sistemi – takip.....	42
4.2.2 Görüş sistemi - yaya geçidi (Zebra) tespiti	43
4.2.3 Görüş sistemi - tünel	43
4.2.4 Görüş sistemi - engel tanıma.....	44
4.2.5 Görüş sistemi - trafik ışığının tanınması	45
4.3. Temsili Robot.....	45
4.3.1 Gazebo, ROS'a bağlantı	49
4.3.2 Benzetilmiş robot kontrol.....	50
4.3.3 Benzetim ortamı	56
4.3.4 Simülasyon arayüzü	58
5. SONUÇ VE ÖNERİLER.....	60
5.1. Sonuçlar	60
5.2. Öneriler	61
KAYNAKLAR DİZİNİ.....	62
EK AÇIKLAMALAR	65
Ek Açıklamalar – A: Simulated robot control	65
A.1. Setup Instructions.....	65
Ek Açıklamalar – B: Basic Characteristics of ROS and Gazebo.....	69
B.1. Gazebo, connect to ROS	69
B.2. Robot description	70
B.3. Robot control - Gazebo	73

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
1.1. Yol trafik ölüm oranları	2
2.1. Portekiz Robot Yarışması.....	7
2.2. Parça genel görünümü	8
2.3. DARPA yarışması	10
2.4. Sparkfun yarışması için izin verilen araç örneği.....	11
2.5. Yarışmada kullanılan Audi Q5 modeli.....	13
2.6. Araba İç bileşenleri.....	14
3.1. Autorally platform ve anagövde modeli (a) Şasi (b) Bileşenleri olan aracın modeli (c) Kontrol platformu	17
3.2. Kullanılan Castle 2028 Extreme 800Kv Autorally Motor.....	18
3.3. Autorally Hesaplama kutusu (a) Sol yandan görünüm (b) Önden görünüş (c) Sağ yandan görünüm	20
3.4. Autorally platform başlatılırken düğümler ve konular	22
3.5. Önerilen düğüm eklenmesi.....	25
3.6. Bir mesajın yapısının temel şeması	25
4.1. Ackermann geometrisi.....	26
4.2. (a) Diferansiyel direksiyona karşı (b) Ackerman direksiyon	27
4.3. Bileşenleri olan aracın modeli	29
4.4. Genel kontrol ve bilgi akış sistemi	30
4.5. Konumlarına göre tekerleklerin taşınması (a) Düz yol (b) Eğik yol.....	31
4.6. Servomotor S60MH.....	32
4.7. Servomotor HS-322HD	33
4.8. Arduino Mega 2560.....	36
4.9. Bilgisayar görüntüsü amacıyla bir masaüstü ROS robotunun kavramsal taslağı.....	36
4.10. Bilgi değişim Şeması	37
4.11. Olası Yer değiştirmeler.....	39
4.12. Özerk kontrolörün parçası	39
4.13. (a) Orjinal görüntü (b) Seçilen görmüş segment (c) İşlem görmüş segment.....	40

ŞEKİLLER DİZİNİ (devam)

<u>Sekil</u>	<u>Sayfa</u>
4.14. Kontur takip eden algoritma örneği (a) Orijinal resim (b) İşlem görmüş olan görüntü (c) Noktaların kaydedilmiş vektörü	41
4.15. (a) Zebranın saptanması (b) Zebranın uygulanması	43
4.16. Tünel içinde görüş. 4.3. (a) Tünelnin saptanması (b) Tünelnin uygulanması	44
4.17. Bir engelden kurtulan görüş alanı. (a) Engelinin saptanması (b) Engelinin uygulanması	44
4.18. (a) Orijinal sinyal trafiği (b) Trafik sinyallerinin uygulanması	45
4.19. Yarışma için kullanılan robot	46
4.20. Benzetimli gerçek modelin şeması, (a) Normal görünüm (b) Tel kafes görünümü ...	47
4.21. Gazebo ile Gerçeklenen PID Kontrolör Şeması	48
4.22. Mesajların dağıtımında kontrolörün gösterilmesi.....	49
4.23. ROS TF çerçeve ağacı (Yazılar çok küçük, okunmuyor).....	52
4.24. Düğümler ve önce yayınlanan başlık döngüsü	54
4.25. Döngüyü kapatıldıktan sonra yayınlanan düğümler ve başlık.....	55
4.26. Rota ve trafik ışığı üzerindeki robot	56
4.27. Pistteki engel örneği. Engel, ray üzerinde herhangi bir yere yerleştirilebilir	56
4.28. Çalışma alanı. Konikler ve kısıtlama bandı	57
4.29. Pistteki tünel yaklaşık 90 derecelik bir şekle sahiptir.....	57
4.30. Parçanın robot ve ilave elemanlarla perspektif görünümü	58
4.31. Kontur bulma işleminden sonra, ilgi alanındaki parçanın ve görüntü izinin alınması	58
4.32. Şekil 4.30 teki durum için arayüz görüntüsü.....	59

ÇİZELGELER DİZİNİ

Cizelge

Sayfa

3.1 Castle 2028 Extreme 800Kv Autorally Motor Teknik Özellikleri	19
4.1 Servomotor S60MH Teknik Özellikleri	32
4.2 Servomotor HS-322HD Teknik Özellikleri.....	33
4.3 Arduino Mega 2560 Teknik Özellikleri	35

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltmalar</u>	<u>Açıklama</u>
ODE	Açık dinamik makine (Open Dynamic Engine)
NTP	Ağ zaman protokolü (Network Time Protocol)
YAML	Ain't Another Markup Language
ICC	Ansal eğri merkezi (Instantaneous Center of Curvature)
IMU	Ataletsel ölçüm birimi (Inertial Measurement Unit)
zFAS	Audi Platform için ileri sürüş asistanı (zentralen Fahrerassistenzsteuergeräts - Advanced Driver Assistance System Platform for Audi)URDF Birleştirilmiş robot tanım formatı (Unified Robot Description Format)
PWM	Darbe genişlik modülasyonu (Pulse Width Modulation)
ESC	Elektronik Hız Kontrolörü (Electronic Speed Controller)
XML	eXtensible Markup Language
GPS	Global pozisyonlama sistemi (Global Positioning System)
DARPA	İleri Savunma Araştırma Projeleri Ajansı (Defense Advanced Research Projects Agency)
RGB	Kırmızı Mavi Yeşil (Red Blue Green)
MPPI	Model tahminli yol integrali (Model Predictive Path Integral)
PID	Oransal, tümevsel türevsel (Proportional Integral Derivative)
AVC	Otonom Araç Yarışı (Autonomous Vehicle Competition)
RAM	rasgerle ulaşılan hafıza (Random Access Memory)
ROS TF	Robot işletim sistemi _çerçeve ağaç (Robot Operating System - Frame Tree)
Rviz	ROS gösterim (ROS visualization)
PPS	Saniyede darbe sayısı (Pulse Per Second)
DOF	Serbestlik derecesi (Degrees Of Freedom)
STL	STereo Lithography
TCP/IP	Transmission Control Protocol/Internet Protocol
3D	Üç boyutlu (Three Dimensions)

1. GİRİŞ VE AMAÇ

Otomobiller, milyonlarca insanın her gün hareket etmesini sağlamak ve onlara özgürlük duygusunu vermek; hatta şehirlerin şekil ve yapılarını tanımlamak gibi tarihin gelişiminde kritik bir rol oynamışlardır. Araba kullanmak çoğu insan için rutin bir alışkanlıktır ama gelecek bir kaç yıl içinde belki buna gerek kalmayabilir.

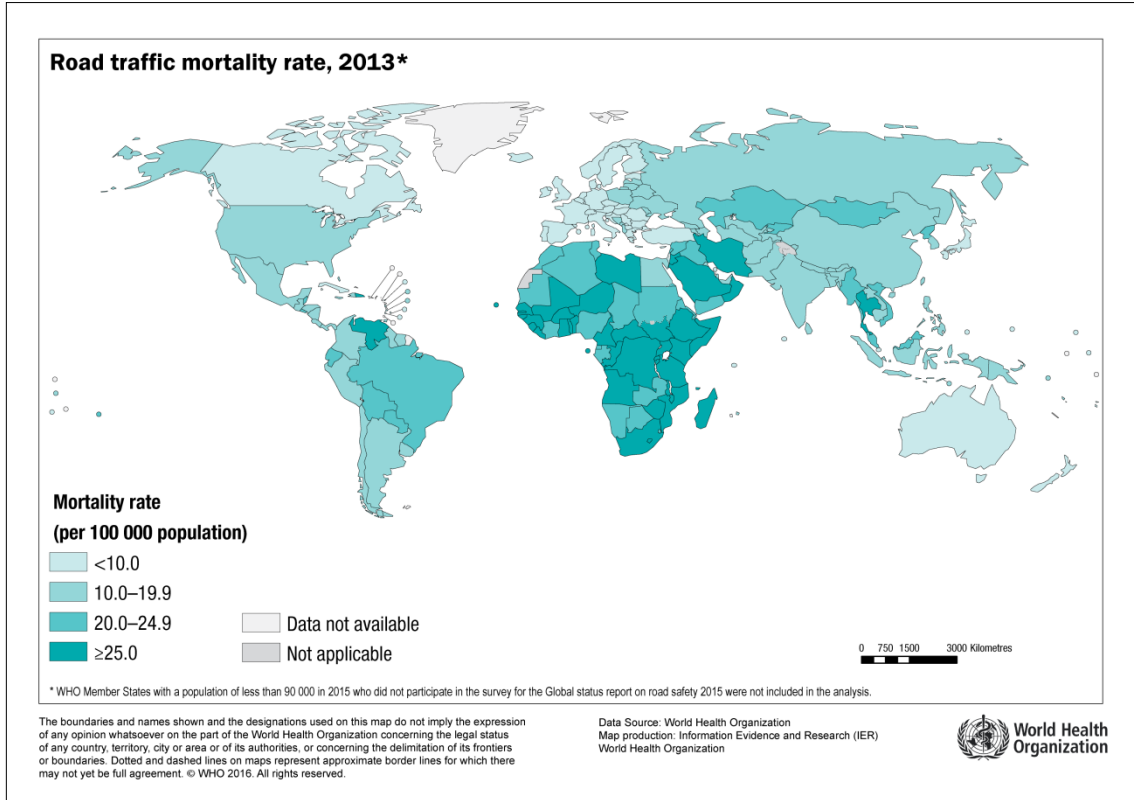
Otomotiv endüstrisindeki bazı şirketler, araç içerisinde bulunabilecek bağlantılar alanında büyük ilerlemeler kaydetti. Her koltukta bulunan, uygulamalar ile kullanılan bir ekran, gelişmiş GPS sistemleri, arabayı çalıştırıp kapatmaya yarayan ses komutları gelecek yıllar içinde her arabada olan özelliklerden sadece bir kaçı olacaktır. Bir arabanın nelere sahip olması gerektiğini düşünmekten daha çok diğer şirketler bir araba ne yapabilir kısmına odaklandı. İşte bu şekilde kendi kendini yöneten araba fikri ortaya çıktı; Otonom Araçlar (Kendi kendini idare eden araçlar). (Ross, 2017)

Son 10 yıldır, insan etkileşimi olmadan çalışan ve kullanıcılarına tüm odaklarını dikkatini yola vermek yerine başka şeyler yapabilme özgürlüğünü sunan araba teknolojisini geliştirmek adına çalışmalar sürmektedir

Otonom araçların başlıca faydaları arasında fiziksel engelli insanlara yardımcı olma, insan hatalarından kaynaklanan kazaların azaltılması yer alır, öte yandan araç sisteminin bilgisayar tarafından kontrol edilebilir olması yakıt tüketimini düzeltmeye yardımcı olacaktır ve hatta aile başına düşen otomobil sayısını azaltabileceği için çevreyi de olumlu şekilde etkileyecektir. (National Council on Disability, 2015)

1.1. Motivasyon

Her yıl trafik kazaları ve ölümcül taşımalar sebebi ile bir çok insan hayatını kaybediyor. Trafik kazaları; kurbanlarına, onların ailelerine ve genel olarak ülkelerine hatırı sayılır ekonomik kayıplara sebep olmaktadır.



Şekil 1.1. Yol trafik ölüm oranları (World Health Organization, 2016)

- Trafik kazaları sonucu her yıl yaklaşık 1.25 milyon insan ölmektedir. Bkz. Şekil 1.1
- Trafik kazaları 15-29 yaş arasındaki genç insanlar arasındaki ölümlerin öncelikli sebebidir.
- Dünya karayolları üzerindeki ölümlerin yarısı “savunmasız kullanıcılar”dır. : yayalar, bisiklet kullanıcıları ve motosiklet kullanıcıları.
- Trafik kazalarının 2030 yılına kadar önde gelen 7. ölüm nedeni olacağı öngörülüyor.
- Yeni kabul görmüş 2030 Sürdürülebilir Kalkınma Gündemi, karayolu trafik kazalarındaki küresel ölüm ve yaralanmaların 2020 yılına kadar yarı yarıya indirilmesi yönünde iddialı bir yol güvenliği hedefi belirledi. (World Health Organization, 2015)

Yol kazalarının azaltılmasının yanı sıra, kendi kendini kullanan taşıtlar, engelli kişileri harekete geçirmeye yardımcı olabilir ve deneyimsiz sürücülere fayda sağlayabilir.

Tam donanımlı otonom bir otomobil, çevreyi algılayabilen, varış noktasına giden yola karar veren ve o noktaya kadar kendini süren bir araba olarak tanımlanabilir. Bunun gelişimi önemli bazı değişikliklere yol açabilir. İnsan kontrolüne veya müdahalesine gerek kalmadan arabanın içindeki herkes bir yolcu olabilir veya hiçbir yolcu taşınmadan bile araba hareket edebilir. (KPMG LLP ve Center for Automotive Research (CAR), 2012)

Bu proje küçük ölçekli otomobiller üzerinde yoğunlaştığından, gelişmiş algılama ve kontrol araştırması için yüksek performanslı bir sına ortamı olan Autorally platformun sürüş mantığı üzerinde çalışılmıştır. Georgia Tech'te geliştirilen robot ROS ile entegre edilmiş ve harici bir algılama veya bilgi işlem gerektirmeyen kendi kendine yeten bir sistem olarak tasarlanmıştır.

Modeli geliştirme sürecinin bir parçası olarak, otomobilin hızı ve manevrası için referans olarak kullanılmak üzere Georgia Teknoloji Enstitüsü'nde Autorally mantığı tasarlanmasına karar verildi.

Autorally araba yaklaşık bir metre uzunluğunda, 21 kg ağırlığında ve saatte neredeyse 100 kilometre üst sınırı hızına sahiptir. R/C kamyon anagövdesi temel alınarak, taşıma yüküne destek olmak için bazı 3D baskılı değişikliklerle birlikte, GPS, IMU, tekli kodlayıcılar, bir çift hızlı video kamera ve bir Nvidia GTX 750ti GPU'lu dört çekirdekli işlemci i7 bilgisayar ve 32 Mb RAM içermektedir. Tüm bu gereçler büyük bir çarpışmada bile en az hasar getirecek bir alüminyum içinde korunmaktadır.

Temel olarak devre üzerinde oldukça hızlı sürüş yapmayı öğrenirken, 2 dakika içerisinde hızı yavaşlatabilirken 5 dakika içerisinde en hızlı seviyesine çıkabilir.

Araba gerçek zamanlı olarak 2,560 km olası yolu hesaplar (2,560 adet muhtemel rota hesaplar); Sonraki 2,5 saniyede ne olacağı 32 GB RAM'li bir GPU tarafından tahmin edilir. Yönü ve gücü kontrol etmek için gereken en iyi değer buradan gelir. Bu işlem, saniyede 60 kez iki hızlı kameradan alınan verilerle gerçekleştirilir. Tüm bunlar yalnızca 17 ms gerektirir; ayrıca sürtünmeyi ve arazinin kaymasını hesaba katar.

Bu proje bir aracın sürtünme sınırlarının yakınında kontrol edilmesiyle ilgili doğrusal olmayan dinamikleri ele almak için özel olarak geliştirilen Georgia Teknoloji Enstitüsü tarafından geliştirilen metod Model Tahminli Yol İntegral kontrolü (Model Predictive Path Integral control MPPI) ile agresif sürüş kullanan bir aracın otonom kontrolü üzerinde yoğunlaşır. (Williams, Aldrich, & Theodorou, 2017)

Kontrol modeli araç-dinamik verilerindeki istatistiksel yöntemleri kullanarak bir yol-integral yaklaşımına dayanan bir olasılıksal optimizasyon kapasitesi döngüsünü kullanır, bir çok olasılıklar arasından en istikrarlı yörüngeleri hesaplar.

MPPI algoritmasında otonom kontrol için önemli bir husus, üç ana bölümden oluşan maliyet formülasyonudur: (1) Pistte kalmak için bir maliyet, (2) istenen hız elde etmek için bir maliyet ve (3) Kontrol için bir maliyet. Maliyet yaklaşımı, bir robot aracın hızı en üst düzeye çıkarabilmesi için önemlidir.

1.2. Amaçlar

Bu projeye, Portekiz Ulusal Robotik Festivali'ndeki otonom sürüş yarışında kullanılan otonom araç modelinin bir benzetim ortamında kullanılması amaçlanmıştır.

- Yarışma için Gazebo ve ROS'u kullanarak benzetim için ortam yaratmak
- Kullanılan yapılandırılmış ve kısmen kontrol edilen çevreyi, yani iki yönlü bir yolu göz önünde bulundurarak, araç navigasyon ve kontrol sistemini, zikzak çizgisi ve kavşaklarla uygulamak
- Mevcut sistemde pano ışığı tarafından gösterilen, gösterge tablosu ışığı tarafından görüntülenen beş farklı sinyalden oluşan (trafik ışıkları simülasyonu) sinyal belirleme ve araba reaksiyonunu uygulamak
- Bilinmeyen bir konumdaki izlenecek yollardan birini engelleyen nesnelere kaçınarak takip edebilme algoritmasını uygulamak
- İzlenen yolun bir bölümünü kapsayan şu andaki tünel geçiş performansını korumak
- "Çalışma bölgesi - working zone " olarak adlandırılan yapılandırılmamış kısmın üstesinden gelmek.

1.3 Raporun Yapısı

Bu kısım, her bölümde sunulan bilgiler hakkında genel bir fikri vererek bu belgenin nasıl sunulduğunu ve organize edildiğini açıklayacaktır.

Bölüm 1 Genel bir giriş ile Otonom Araba ile çalışmanın ana motivasyonlarını hem de projenin spesifik hedeflerini kapsar.

Bölüm 2 araçların yeterliliklerinin önemini açıklayan ve her biri için en önemli ve en tanınmış bazı ölçütleri listeleyen otonom sürüş yarışmalarına odaklanmıştır.

Projenin temellerinden biri, daha sonra simüle edilmiş bir model oluşturmak için gerçek modeli başlangıç noktası alan Autorally projesinin benzetimidir. Bölüm 3'te fiziksel özelliklerin ve konuların ve devrelerin etkileşimi de dahil olmak üzere bir açıklama yapılmıştır.

Bölüm 4'te, araba benzeri robotun tanımı detaylandırılmıştır. Bu bölüm, robotun fiziksel özelliklerini ve onu oluşturan unsurları tanımlar, önemli sistemler bu tanım ve açıklama Gerçek modele dayalı olarak robotu taklit etmek için gerekli sistemlerin açıklamasını yapar ve aynı zamanda gerçek model için simüle edilen modelde kullanılan denetleyicinin bileşenlerini ayrıntılarıyla belirtmektedir.

Bölüm 5'te, yapılan çalışmadan elde edilen bazı sonuçların yanı sıra ileride üzerine çalışılması devam etmesi beklenen konular yer almaktadır.

Son olarak, projedeki programı başka bir bilgisayarda kurmak ve çalıştırmak için bir el kitabı ve ayrıca ROS ve Gazebo araçlarının en önemli yönlerini anlamak için temel kavramlar kılavuzu içermektedir.

2. LİTERATÜR ARAŞTIRMASI

Önemli şirketler otonom araç alanında önemli avantajlara sahip olmasına rağmen; bu alandaki genellikle akademik yarışmalar, yeni yeteneklerin deneyim kazanmasına ve otonom araç kontrolü dünyasına katılmasına izin verir.

Genellikle otonom sürüş yarışmalarında farklı yerlerden gruplar, aracın farklı parametrelerde tutulmasına izin veren sensörlerin proses bilgisini içeren ölçeklenmiş model araçlarını kullanırlar.

Temel konsept, aracın mümkün olduğunca hızlı otonom bir şekilde parkuru geçmesi ve bunu en az sayıda hata vererek yapabilmesidir. Genellikle araçlar engellerden kaçınılmalı ve yörüngeleri sırasında trafik sinyallerini algılamalıdır.

Bilgilendirici bir yol olarak, bazı robot yarışmalarına katılımların bazı temel özelliklerinden bahsedilecek ve bu doğrultuda taklit edilen ortamın oluşturulması temel alınacaktır. (Costa, Rossetti, & Sousa, 2016)

Kullanılan resimlerin bazıları kuralların açıklayıcı belgelerinde bulunurken diğerleri bazı önemli hususları açıklamak için yapılmıştır.

2.1. Ulusal Robot Festivali –Portekiz

Otonom Sürüş yarışması, seyir halinde bir otonom robotun, klasik bir yola benzeyen kapalı bir yolda bir rota boyunca geçmesi gereken orta karmaşıklıkta bir teknik zorluğu meydana getirir.



Şekil 2.1. Portekiz Robot Yarışması

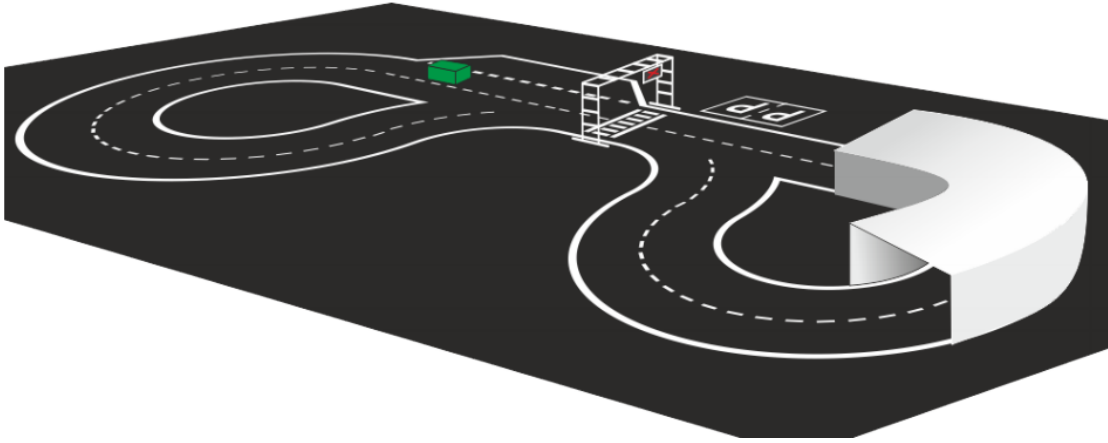
Ackerman direksiyon ve ROS tabanlı yazılımı kullanarak, insan kontrolü veya müdahalesi olmadan kendini sürme becerisine sahip küçük ölçekli bir araç zaten mevcuttur. Bir lazer sensör, birden fazla kaynaktan bilgi edinmek için iki kamera ve araç hızının kapalı çevrim kontrolüne izin veren tekerlerdeki kodlayıcılar bulunur. IPEiria'nın 2014 ve 2015 yıllarında kazanımı olmasını sağlayan yarışmada başarıyla test edilen geliştirilmiş sistem versiyonu, henüz ne enkoderlerden alınan bilgileri ne de kamera tarafından alınan bilgileri kullanmadı. Bkz. Şekil 2.1

Bir benzetim ortamı da, başlangıçta bir algoritma analiz aracı olarak görev yapacak olan GAZEBO ve ROS kullanılarak geliştirildi. Böylece paralel iş geliştirme, donanım sisteminin çalışmasını sağlarken, aynı zamanda da otonom sürüş algoritmalarına odaklanıyor. Devre yapısı, mesajlaşma, mevcut sensör sürücüler ve görselleştirme araçları, sistemin performans analizine yardımcı olmaktadır. (Sousa & Pereira, 2017)

Yarışma dört turda organize edilir ve her takımın her turda 10 dakikalık bir zaman dilimi vardır. Ekiplerin kendi zaman dilimlerini nasıl yönetebilecekleri konusunda esneklikleri mevcuttur. Sınıflandırma maddeleri şunlardır:

- a) Sürüş Görevi: (hız+)
 - i. Hız
 - ii. Engeller
 - iii. Tünel
 - iv. Çalışma alanı

- b) Park etme Görevi: ($=\text{hız}$)¹
- Engeller ile paralel park etme
 - Engelsiz paralel park etme
- c) Dikey Trafik Sinyalleri: ($-\text{hız}$)



Şekil 2.2. Parça genel görünümü

Parça genel görünümü, Gazebo'daki simülasyon ortamına PNG formatında bir resim olarak yerleştirilen Şekil 2.2'de açıklanmaktadır.

Hız açısından dikkat edilmesi gereken nokta:

- Sürüş: saf hız
- Engeller: Yeşil renk
- Tünel: yaklaşık 90°
- Çalışma yeri: Üçgen koni boyutları 30cm

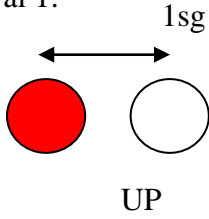
Sinyal Verme Methodu

- Gösterge ledleri
- Yeşil: Bilgi Sinyali
- Kırmızı: Çalışma Sinyali
- Mavi: Zorunlu Sinyal

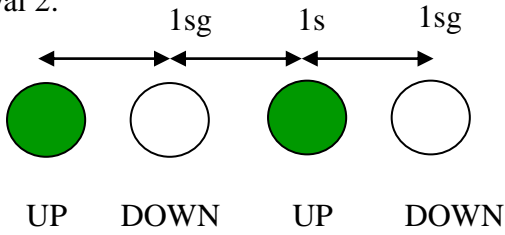
¹ Aracın park alanında performansı, gelecekteki yaklaşımlar için düşünülmesi gereken, bu proje kapsamında olmayan bir konudur.

Sinyal Zamanlayıcısı:

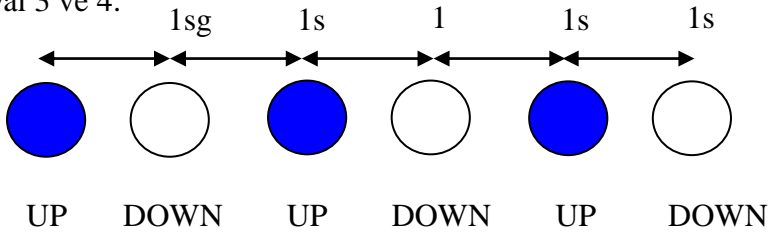
Sinyal 1:



Sinyal 2:



Sinyal 3 ve 4:

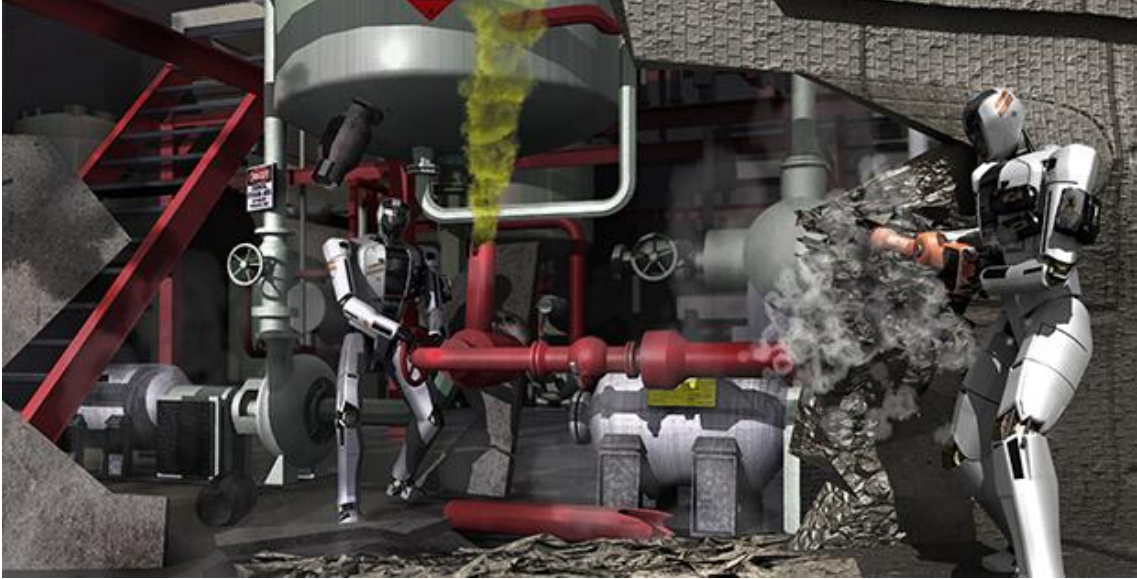


2.2. DARPA Büyük Mücadelesi (Grand Challenge - DRC)

Savunma Bakanlığı, afet yardımı sorununa teknolojik yenilik robotları yoluyla yaklaşmayı amaçlayan DARPA Robotik Yarışması (DARPA Robotics Challenge - DRC) stratejik planını hazırladı.

DRC rekabetinin temel amacı, tehlikeli ya da zor erişimli ortamlarda çeşitli görevleri yerine getirebilecek robotlar oluşturmaktır; bu nedenle, gerekirse, bir araç gibi basit veya daha karmaşık araçlar kullanılır.

Yarışmanın zorluk derecesinin yüksek olduğu düşünülüyordu, bu nedenle katılan ekipler, konunun önemli ilerlemelerine sahip araştırma ve geliştirme organizasyonlarının temsilcileri ile kısa bir süre içerisinde robotun donanım ve yazılımını ve İnsan etkileşimi arabirimlerini önerme ve geliştirmeye odaklanıyorlar. (DARPA Robotics Challenge, 2015)



Şekil 2.3. DARPA yarışması

Benzetimde, gerçek yaşamda banliyö unsurları olarak bulunabilecek unsurları içeren ve robotun yüksek karmaşıklıkta ortamlarda etkileşime girdiği göz önüne alınarak DARPA yarışmasının Sanal Robot Yarışması'nın (Virtual Robotics Challenge - VRC) benzetilmiş bir versiyonu da bulunmaktadır.

DARPA yarışmasına dayanarak DRCim ile Gazebo ortamında benzetilen Sanal Robot Yarışması (VRC) katılımcıları tarafından kullanılan model, ortam, tamamlayıcı ve araçlardan oluşan bir koleksiyon olan DRC Benzetim Programı oluşturuldu. Gazebo, DRC Benzetim Programı veya DRCSim üzerine inşa edilmesi, ek dünyalar, modeller ve kodlar DRC alanını benzetmek için gerekli olan araçları oluşturur.

Sanal yarışmanın en göze çarpan modeli açık ve kapalı ortamlarda etkileşime giren Atlas Robot benzetim programıdır. Program, engel tespiti için algılayıcıların ve araçların

bileşenlerini içerir. Şekil 2.3'deki örnek. (Defense Advanced Research Projects Agency DARPA, 2013)

2.3. AVC Sparkfun

Elektronik şirketi Sparkfun, akademik yarışmadan ziyade daha amatör bir karakter yarattı.

Yarışma iki otonom araç sınıfına sahiptir: Klasik ve Güç Yarışı. Klasik sınıf araç, bir R / C otomobilinin büyüklüğündedir ve Discombobulator (benzinli bir döner platform), bir top çukuru, keskin viraj dönüşleri ve isteğe bağlı bir çamur tuzağı kısayolunu içeren tehlikeli, uygun büyüklükteki bir parkurda yarışılır. Güç Yarışı Sınıfı, küçük çocuklar tarafından yönlendirilebilecek boyutta olan Güç Direksiyonu araçlarına dayanıyor ancak bu yarışta çocukların yetişkinleri taşıması gerekiyor. Hem otonom, hem de insan güdümlü Güç Yarışçıları için yarışlar vardır.



Şekil 2.4. Sparkfun yarış için izin verilen araç örneği

Kara Araçlarının, bu amaç için binanın dört dış duvarının tamamını dolaşması gereklidir.

Çevresel deęişiklikler yapılmasına izin verilmemektedir (rakipler şeridi, kızılötesi sinyal lambalarını veya dięer geçici veya yarı kalıcı işaretleri koyamazlar). Çok az katı kurallar bulunmakla birlikte, araçta (Şekil 2.4'deki örnek):

- Yüksüz aęırlık zorunluluęu yoktur
- Boyut zorunluluęu yoktur
- Gerilim sınırı zorunluluęu yoktur
- Ana gövde için bütçe limiti yoktur
- Uzaktan kapatma anahtarı teşvik edilir, ancak zorunlu deęildir
- 40 lb'yi aşan herhangi bir araç kesinlikle yasaktır. 25 lb üzerindeki her araç acil durumlar için güvenli kontak kapatma anahtarı içermek zorundadır. (AVC.Sparkfun, 2016)

2.4. Audi Otonom Sürüş Kupası

Bu yarışma, karmaşık bir devre ile karşı karşıya olan Audi Q5 modelinin 1:8 ölçekli otonom sürüş modeli ile Audi şirketi tarafından yönetilmektedir. Şekil 2.5'deki modele bakınız.

Katılımcılar Audi tarafından sağlanan temel yazılımı başlangıç noktası olarak kullanmakta ve daha sonra sensör verilerini işlemek, durumları doğru yorumlamak ve arabayı kontrol etmek için kendi yazılımlarını yaratmaları gerekmektedir.



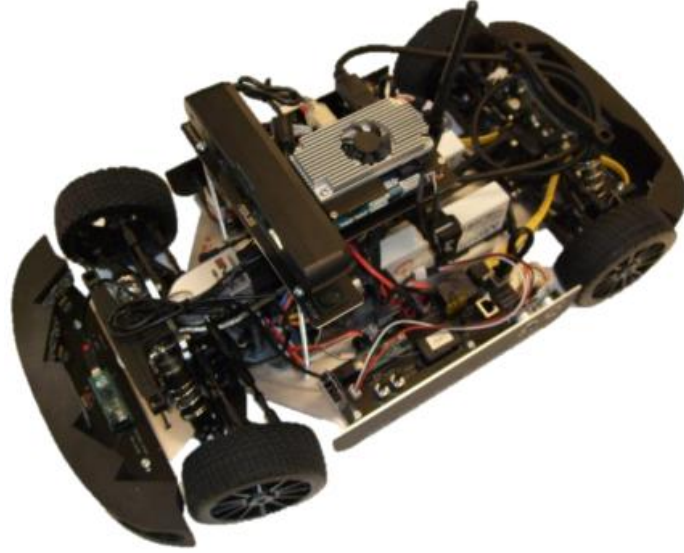
Şekil 2.5. Yarışmada kullanılan Audi Q5 modeli

Bu yarışma, elektrikli tahrik aracının mümkün olan en kısa sürede devreyi otonom biçimde çalıştırması ve en az sayıda hata yapması sağlaması amaçlamaktadır. Tasarım devresi, genellikle yolda bulunan gerçek sürüş koşullarından bazılarını temsil eder. Robotlar farklı engelden kaçmak, hareket eden trafik koşullarında kavşaklara yaklaşmak ve trafik sinyallerini gözlemleyerek, ayrıca akan trafikte diğer araçları izleyerek, onları aşmak zorundadır; Güvenli mesafenin korunması. Buna ek olarak, öngörülme engeller ortaya çıktığı zaman yaklaşan trafikte sola dönme, hassas park manevraları ve acil frenleme eylemleri gibi zorluklar da bulunmaktadır.

Katılımcılar son versiyonun sonuçlarına ve işlevlerine dayanarak çalışırlar. Zorlukların karmaşıklığı yıldan yıla artmaktadır. Kazalar, zaman kayıpları ve uygulamadaki hatalar için puan düşülür. Zorunlu kısım ücretsiz bir uygulama tarafından takip edilir. Rekabetin son günü ekipler, araçlarının ölçeklenebilmesi için ek bir test sunabilirler. Bu görsel örnekleme ücretsizdir ve her şeyden önce yaratıcılık felsefesinin gelişimine dayanmaktadır

Kullanılan ana sensör, video kameranın yüksek teknoloji ürünü bir modelidir. Gerçek hayatta olduğu gibi, bu kamera da yol yüzeyini, trafik işaretlerini, engelleri ve diğer kullanıcıları otomobilin önünden tespit ediyor. Model araba ölçeği on adet ultrasonik sensör ile tamamlanmaktadır: Önde beş, arkada üç ve her iki tarafta birer tane. İki

santimetre ve dört metre arasındaki algılama aralığı, çevrenin yüksek hızlı sirkülasyonda bile hassasiyetle tanımlanmasını sağlar. Bkz. Şekil 2.6.



Şekil 2.6. Araba İç bileşenleri

Ayrıca, ölçek modelinin yaptığı ve diğer tüm sistemlerin yaptığı gibi değişiklikleri kaydeden bir hızlanma sensörü içerir ve bu bilgileri gerçek zamanlı olarak yüksek hızlı bir dört çekirdekli işlemci bulunan merkezi bilgisayara gönderir. Bu konfigürasyon, Audi'nin gelecekteki elektronik mimarisini yansıtıyor. İçindeki kontrol merkezi birimi (zFAS) verileri değerlendirir, çevre için bir model oluşturur ve tüm yardım sistemlerine gerekli bilgileri sağlar. (Audi Corporation, 2017)

3. MATERYAL VE YÖNTEM

AutoRally platformu gelişmiş algılama ve kontrol araştırması için yüksek performanslı bir test yatağıdır. ROS ile entegredir ve harici bir algılama veya bilgi işlem gerektirmeyen bağımsız bir sistem olarak tasarlanmıştır. Robot araştırmacılara ve hobileri bu yönde olan insanlara agresif otonom arazide sürüş alanını açmaktadır.

Amaç IPEiria Üniversitesi'nin hali hazırda sahip olduğu robot modeline hangi devrelerin ve performans özelliklerinin uygulanabileceğini belirlemek için Autorally platformunu gözden geçirmektir.

Gazebo / ROS'daki benzetilmiş grafik ortamı, bu özellikleri tanımlamak için iyi bir araçtır, bu nedenle benzetim analizi, her devrenin gerçekleştirdiği işlev üzerinde yoğunlaşmıştır.

Benzetimde, sözü edilen uygulama ile pek alakalı olmayan bazı parametreler bulunduğu için bunların atlanmaları ve belirli işlevler için yeni devreler uygulanmasının göz önüne alınması mümkün olabilir.

AutoRally platformu, aynı mesajla kontrol edilen 3 aktüatöre sahiptir:

- direksiyon
- gaz kumandası / arka fren
- ön fren

Autorally ana gövde devresi, aşağıda açıklanan adımları takip ederek herhangi bir programın herhangi bir aktüatörü kontrol etmesine izin verir. Herhangi bir zamanda, kendi ana gövde Komut konularına otomatik olarak msgs / ana devre komut mesajları yayınlayan birçok devre olabilir. Autorally anagövde tüm bu mesajları alır ve her aktüatör için en yüksek öncelikli geçerli komutu seçer. Bu tasarım, farklı programların, bağımsız direksiyon ve gaz kumandası kontrolörleri gibi farklı aktüatörleri aynı anda kontrol etmesini sağlar. Bu yapı, yüksek öncelikli denetleyicilerin, daha düşük öncelikli

denetleyicilere istenen şekilde kontrol devretmesini sağlayarak, denetleyiciler arasındaki önceliklerine göre kesintisiz, otomatik anahtarlamaı destekler. Autorally anagövdesi (şasisi), her aktüatör komutunu arChassisConfig şasi kalibrasyon .yaml dosyasını dikkate alarak bir PWM sinyaline çevirir ve aktüatörlere gönderir.

Durum tahmin düğmesi GPS konum ve IMU ölçümlerini (doğrusal ivme ve açısız hız) kaydeder ve araç konumu ve hızını içeren doğru ve hızlı bir durum tahmini üretir. Bu tahmin daha sonra, gerçek zamanlı olarak, araç üzerindeki diğer işlemler ve kontrolörler tarafından kullanılmak üzere poz tahmini başlığında yayınlanır.

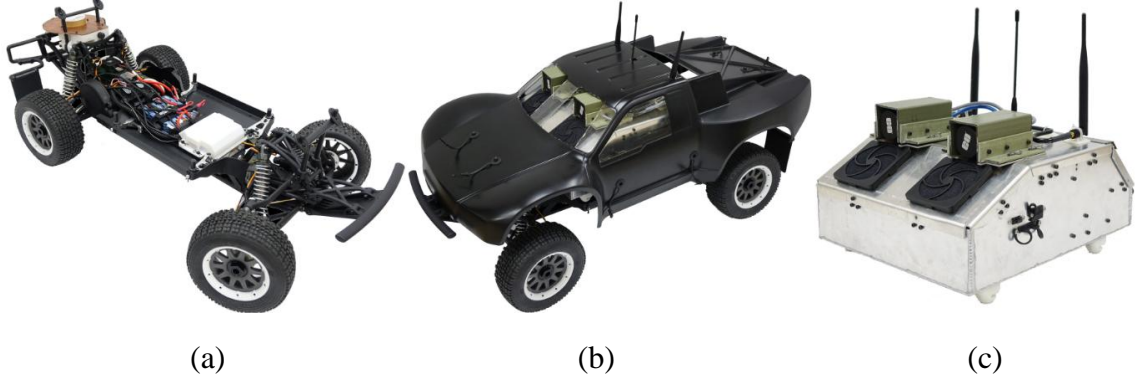
Durum tahmincisi, GPS ve IMU mesajlarının sistem saatine göre doğru şekilde zaman damgasını vurmasını gerektirir.

Yol noktası İzleyici denetleyicisi, durum tahmincisi tarafından kullanılan global koordinat çerçevesinde tanımlanan ardışık yol işaretleri arasındaki AutoRally platformunu yönlendirecektir. (Williams, Aldrich, & Theodorou, 2017)

Sabit hız kontrol cihazı, gerçek veya benzetilmiş AutoRally platformunun hızını, hız komutu mesajında kullanıcı tanımlı bir değerde kontrol etmek için bir PID denetleyicisi kullanır. Belirli bir hız için gaz kelebeği ayar noktası, servo komut değerleri ile kararlı araç hızlarına eşleşen bir gaz kelebeği kalibrasyon dosyasından gelir.

3.1. Fiziksel Nitelikler

Autorally anagövdesi, standart bir RC vericisi veya elektronik kutusuna bağlı bir bilgisayar ile kontrol edilir. Otonom ve manuel kontrol arasındaki kesintisiz uzaktan geçiş, verici üzerinde bir anahtar ile yapılır. Verici üzerindeki son kanal, başka bir düğme ile uzaktan kumanda şeklinde kontrol edilir. Gaz kumandası sinyali ile Elektronik Hız Kontrol Cihazı (ESC - Electronic Speed Controller) arasında oturan anagövde canlı bir röleyi kontrol eder.



Şekil 3.1. Autorally platform ve anagövde modeli (a) Şasi (b) Bileşenleri olan aracın modeli (c) Kontrol platformu

Kullanılan anagövde, 2.4GHz Radyo Sistemi ile yönetilen ve süspansiyon ve aktarma organlarını yöneten ve 26cc Benzinli Motora sahip 1/5 ölçekli HPI Baja 5SC anagövdedir. VVC / HD Şoklar, 8000 RPM Debriyaj, Yüksek Akış Susturucu, Viskoz Tork Diferansiyelleri, Monokok Kasa, Çift Ventilli Disk Frenler ve gerçek yüksek performanslı Beadlock Tekerlekler gibi standart özelliklere sahip Baja 5SC, yüksek performanslı bir makinedir. HPI 2.4GHz Spread Spectrum DSSS telsiz sistemi ayrıca, yüksek performanslı servolar ve düşük voltaj kaçaklarını önlemek için güvenilir işleviyle birlikte kitin standart ekipmanıdır. Bkz. Şekil 3.1

Baja 5SC, HPI Baja serisini, Baja 5SC'nin otantik Kısa Yol görünümünü veren SC-1 gövdesi ile Kısa Yol Rotasına taşımaktadır. Dayanıklılığa ve gerçekçi görünüme katkıda bulunanlar, kaydırma plakası ön tampon ve gerçek lastik çamurluk kapaklı büyük arka tampondur. Güç, Baja 5SC'yi 40mph'in üzerinde hızlandıran ve düzenli kurşunsuz pompa benzinli tek bir depoda 45 dakikadan fazla süren muazzam bir 26cc Yakıt motorundan geliyor. Herhangi bir toprak yüzeyde ekstra çekiş sağlamak için, HB Rodeo lastikleri artık standart ekipman olup, TR-10 kordon kilidi kamyon tekerleklerine monte edilmiştir. Dayanıklılık, Super Heavy Duty 9mm tahrik milleri, Heavy Duty arka göbek taşıyıcıları ve Ağır Hizmet ön altıgen göbekleri ile sağlanmaktadır. 2.4GHz radyo sistemi, frekans kristalleri sorunsuz hassas kontrol ve uzun mesafeli alım için standart bir ekipmandır. (Georgia Institute of Technology, 2017)

Özellikler:

- Uzunluk: 900mm
- Genişlik F / R: Ön 440mm / Arka 460mm
- Yükseklik: 320mm
- Parça F / R: Ön 370mm / Arka 380mm
- Aks Mesafesi: 570mm
- Ağırlık: 12.6kg / 27.9lb (yakıtsız)

Kullanılan motor “Castle 2028 Extreme 800Kv’dir”. HPI Baja 5B ve 5T gibi 1/5 ölçekli araçlarla ağır gaz ve uzun süreli kullanım sağlayan yüksek verimli fırçasız motor; 2WD için 30 lb'ye ve 4WD için 25 lb'ye kadar ağırlığındaki herhangi bir 1: 5 ölçekli RC hobi aracı (gazdan elektriğe dönüşümler dahil) için mükemmel bir yükseltmedir. Yüksek kaliteli stator ve mıknatıs şekilleri, geliştirilmiş çalıştırma performansı için daha düşük vuruntu momenti sağlar. Bkz. Şekil 3.2.



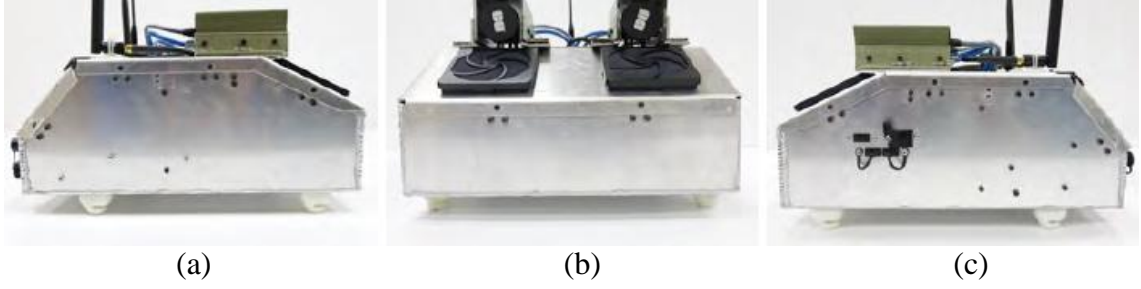
Şekil 3.2. Kullanılan Castle 2028 Extreme 800Kv Autorally Motor

Çizelge 3.1'deki Teknik Özellikler:

Çizelge 3.1: Castle 2028 Extreme 800Kv Autorally Motor Teknik Özellikleri

Giriş Voltaj Aralığı:	12S MAX
Kv:	800
Max. RPM:	45
Boyut:	Uzunluk: 4.4" (113.2mm) Çap: 2.2" (57mm)
Ağırlık (w/wires):	44 oz.
Mil Boyutu:	Uzunluk: 30mm Çap: 8mm
Montaj Boşluk Özellikleri:	4mm aralıkla @ 30mm boşluk.
Soğutma Fanı:	Var, ancak dahil değil.
Konnektörler:	8.0mm eril konnektörler
Çalışma İçin Gerekli Parçalar:	Uyumlu Elektronik Hız Kontrolü (ESC - Compatible Electronic Speed Control)
Ürün Kullanım İfadeleri:	N/A
Başvuru Kılavuzları:	N/A
Maxs. Önerilen Araç Ağırlığı:	25 lbs 'den daha az ağırlığa sahip 4WD araçlar. 30 lbs'den daha az ağırlıkta 2WD araçlar.
Teknik Notlar:	Motor sıcaklığını yakından izleyin; Bir motorun aşırı ısınması mıknatıslara kalıcı hasar verecektir. Motor veya ESC'de kesinlikle 180 ° F'yi aşmayın.

Ayrıca benzetim programı Hesaplama Kutusu ve Operatör Kontrol İstasyonu (OCS - Operator Control Station. Bkz. Şekil 3.3) adlı bir kontrol tabanı içerir ve dağıtık bir ROS sistemi içindeki tüm bilgisayarlarda saatleri senkronize etmek için kroni kullanılır. Robot bir NTP sunucusu görevi görür ve tüm harici makineler saatlerini robota senkronize eder. Robottaki hesaplama kutusu saatini, kroni ve gpsd'yi kullanarak GPS'lerin PPS sinyaliyle birleştirilmiş olarak bildirilen GPS zamanına senkronize eder. GPS uyduları bir zemin 0 saatidir, bu nedenle robottaki kroni sunucu bir zemin 1 saati olarak yapılandırılmıştır.



Şekil 3.3. Autorally Hesaplama kutusu (a) Sol yandan görünüm (b) Önden görünüş (c) Sağ yandan görünüm

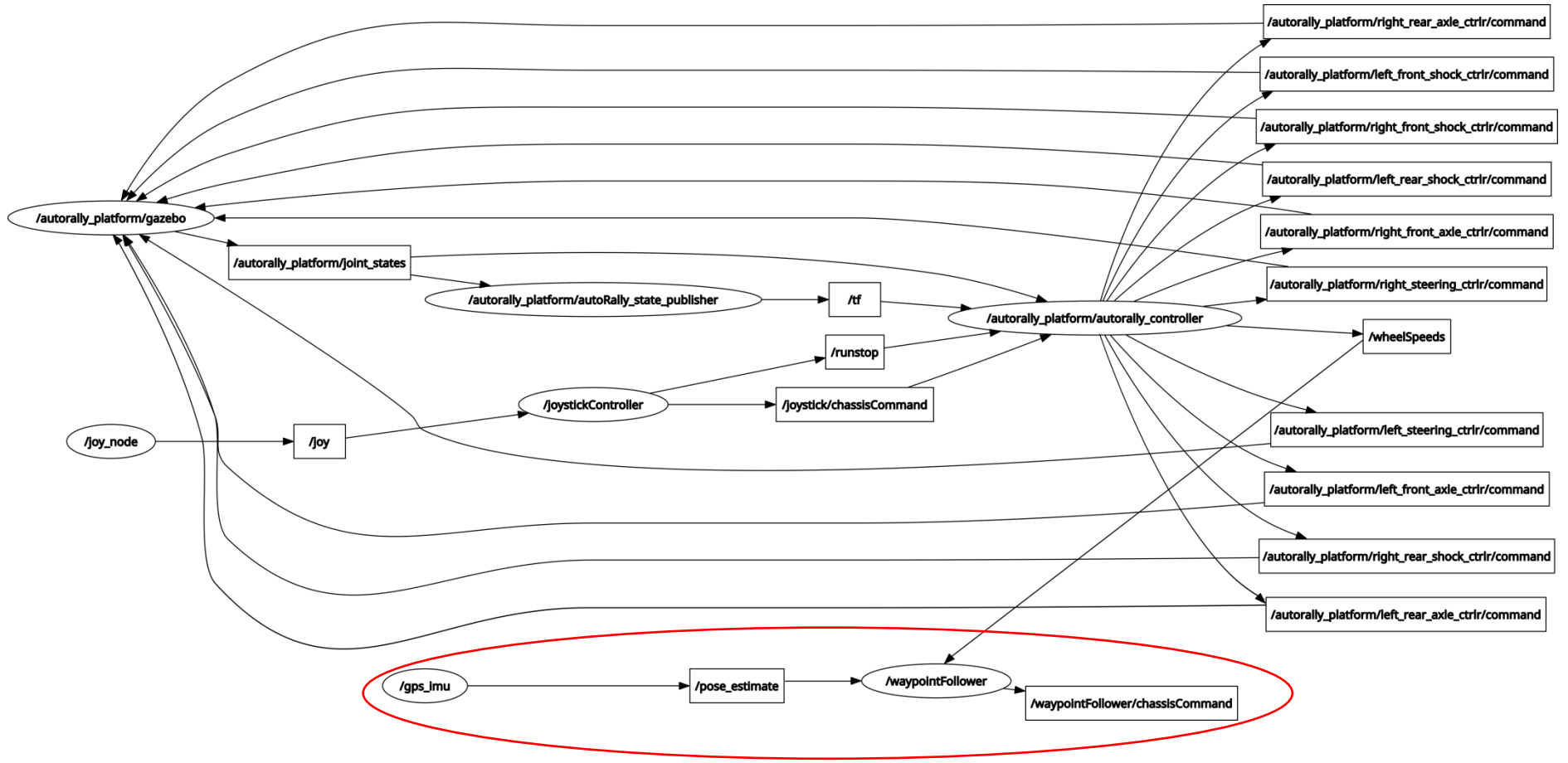
3.2. Autorally Nodes (Düğüm) Fonksiyonları

Benzetimde başlatma sırasına göre düğümlerin işlevleri aşağıda açıklanmıştır:

- Autorally çekirdek yöneticisi (autorally_core_manager): Adının belirttiği gibi hız, şasi, 'Arduino On board' ve Servo Arayüzü gibi Autorally'nin temel işlevlerini denetler.
- autorally platform / autoRally bölge yayıncısı (autorally_platform / autoRally_state_publisher): tf için bir robot durumunu yayımlar. Bölgede bir kez yayımlandıktan sonra, sistemdeki tf'yi kullanan tüm bileşenler için erişilebilirdir. Paket, robotun eklem açılarını girdi olarak alır ve robotun kinematik ağaç modelini kullanarak robot bağlantılarının 3D pozlarını yayımlar.
- autorally platform / autorally kontrolör (autorally_platform/autorally_controller): Ackermann direksiyon ile bir aracın tekerleklerini kontrol eder.
- autorally platform / kontrol avcısı (autorally_platform/controller_spawner): Bir anda denetleyiciler kümesini otomatik olarak yükler ve başlatır ve otomatik olarak denge konumunda olan, birleşme bölgeleri ve amortisör argümanları dahil edilen aynı kontrolleri birden bire durdurur ve yüklemeyi kaldırır.
- autorally_platform/gazebo: Ackermann direksiyon ile bir aracı görselleştirirken hem RViz hem de Gazebo tarafından kullanılan düğümleri başlatır.
- Eğlence düğümü (joy_node): Eğlence paketi, sevimli bir Linux oyun çubuğunu ROS'a bağlayan bir düğüm olan eğlence düğümünü içerir. Bu düğüm, joystick'in düğmelerinin ve eksenlerinin her birinin geçerli durumunu içeren bir "Joy" mesajı yayımlar.

- Eğlence Çubuğu Denetleyici (joystickController): Bu düğüm gaz kelebeği ve direksiyon etkinleştirme düğmelerinin parametre bilgilerini içerir.
- gps_imu: Robotun başlangıç konumunu belirler
- Yol noktası İzleyici (waypointFollower): Parçanın içinde robotun izleyebileceği bir dizi nokta atar.
- Sabit hız kontrol cihazı (ConstantSpeedController): Özerk denetleyiciyi başlatır. (2017 Georgia Tech AutoRally, 2016)

Yerleşik modeldeki işlevlerin, geçerli robottaki uygulama için büyük bir önemi olmadığı için, Durum Tahmincisi ve Yol noktası İzleyici düğümleri ihmal edilebilir. Bahsedilen düğümlerin atlanmasına ilişkin grafik, Şekil 3.4'deki gibidir



Şekil 3.4. Autorally platform başlatılırken düğümler ve konular

3.3. İşlem Ayrıntıları

Eklenen Autorally Mesaj dosyaları:

- Tekerlek Hızları.mesajı (WheelsSpeeds.msg):

Her bir tekerleğin mevcut hızı m / s olan mesajları içerir. Tekerlek hızının alınması, arduinoOnboard.ino dosyasındaki tekerlek hız sensörlerinden başlar; tekerlek dönüşü ve yerleşik veri paketlerinin gerilimini ölçmek için 4 Hall Efekt sensöründen veri toplar ve tüm verileri bir Arduino Verileri mesajıyla yayınlar.

Arduino kartı başlangıçta veri yayınlar. ArduinoOnboard_class bu veriyi bir arduino veri iletisi olarak yayınlar. Nihayet verileri sınıf uygulama dosyasında yayınlamak için (ArduinoOnboard.cpp) kullanılır.

Bilgiler, float64 biçiminde lfSpeed, rfSpeed, lbSpeed ve rbSpeed alanlar içeren Tekerlek Hızları.mesajı iletisiyle işlenir.

- Şasi Emir .mesajı (chassisCommand.msg):

ArduinoData mesajından gelen, float64 biçiminde kasa üzerindeki gaz, direksiyon ve ön fren kontrolü mesajlarını da içerir.

Birden fazla komutan herhangi bir zamanda komut gönderiyor olabilir, aktüatörlere sadece en yüksek öncelik komutu iletilir. Her aktüatör ayrı bir kumanda tarafından kontrol edilebilir.

- dur.mesajı (runstop.msg):

Hareketin yazılım aracılığıyla etkinleştirilip etkinleştirilmediğini kontrol eden mesajlar içerir. Şasinin kendiliğinden sürülmesi için hareket etkinleştirilmiş değişkeni doğru olarak ayarlanmış bir runstop mesajının yayıncısı olmalıdır.

- Şasi durum.mesajı (chassisState.msg):

Mevcut kontrol durumlarına sahip mesajları ve komut verilen aktüatör değerlerini içeren teşhis, çok fazla şasi bilgisi ve mesaj oranı bilgisi içerir.

Oluşturulan Autorally mesajlar:

- std mesajı (std_msgs):

Şoklar, akslar, direksiyon vb. gibi geçerli bileşenin durumunu Float64 formatında içeren mesajları içerir

- sensor_msgs:

Sensörden alınan bilgileri farklı senaryolarda /JointState, /Image, /CompressedImage olarak dönüştürür.

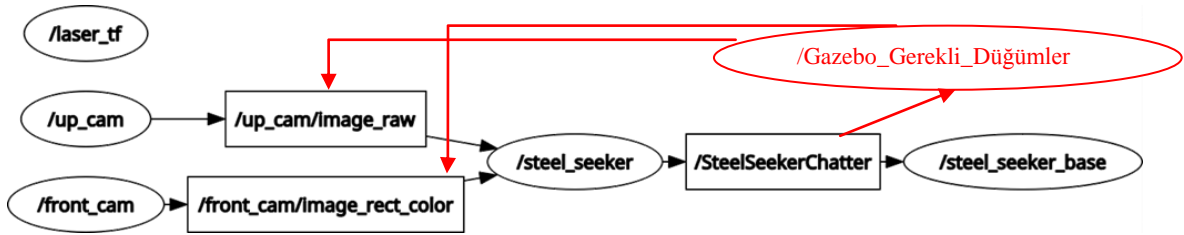
- geometri mesajları (geometry_msgs):

Olası ulaşılabilecek noktaların ve yol işareti düğümünde olası rota yaklaşımının bilgisini içerir.

- Tanılama mesajı (diagnostic_msgs):

GPS yayıncılığında gelen bilgileri içerir ve yol noktası düğümündeki başlangıç konumunu tahmin etmek için kullanılır. (Ackerman, 2016)

Önerilen sistem açıklaması: Gazebo kontrollerini ve durum yayıncısını içeren yeni düğümler ekleyin. Bkz. Şekil 3.5:



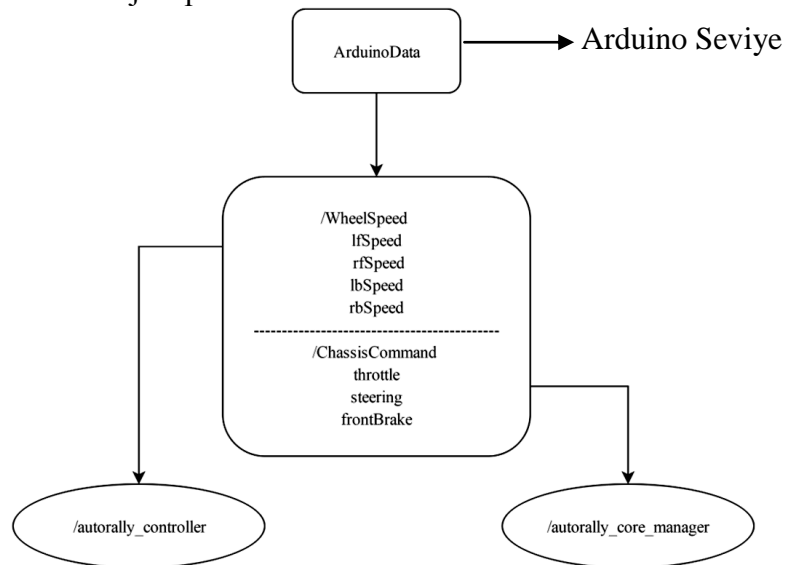
Şekil 3.5. Önerilen düğüm eklenmesi

Aşağıdaki şema (Şekil 3.6) mesajın temel etkileşimini temsil etmeyi amaçlamaktadır:

○ → Devre Düğümleri

□ → Başlık

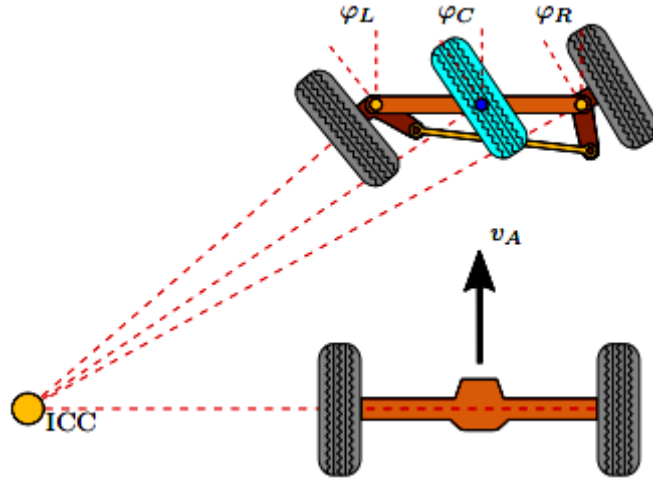
▭ → Mesaj Yapısı



Şekil 3.6. Bir mesajın yapısının temel şeması

4. BULGULAR VE TARTIŞMA

Özerk bir aracı araştırmak için hareketlerin eşitliklerini ve robotun bileşenlerini benzetim ile gözden geçirmek gerekir. Robotlarda, otomobillerde bulunana benzer mantıksal bir Ackerman direksiyon uygulanmaktadır. Buna ek olarak, Ackerman operasyonuna sahip araçların Ackerman direksiyonunun robotları olarak adlandırılacağını vurgulamakla birlikte, diferansiyel operasyonların robotları da düşünülmelidir. Ackermann sürüş robotunun simülasyon stratejisini anlamak için, direksiyon araçlarının geometrisinin ve özellikle Ackermann direksiyonunun daha ayrıntılı incelenmesi gerekiyor.



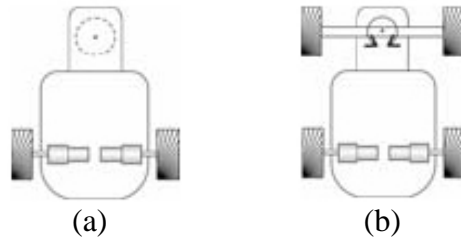
Şekil 4.1. Ackermann geometrisi

Bir aracın her bir tekerleğine dik olan çizgi, tekerlekleri senkronize etmek için kesişmelidir. Kesişme noktası ICC (Instantaneous Center of Curvature - Anlık Eğrilik Merkezi) 'dir. Her bir tekerlek için eğri yarıçapı, tekerlek ve ICC arasındaki mesafe ile tanımlanır.

Ackerman direksiyonlu robotlarda, iki gerçek ön tekerlek arasındaki ortada üçüncü bir sanal ön tekerlek eklenerek tüm araç için bir eğri yarıçapı tanımlanmaktadır. Eğrinin yarıçapı R_C ve direksiyon açısı φ_C olarak tanımlanır.

Robot yönünde x eksenini ve solunda y eksenini olan bir Kartezyen koordinat sistemi tanımlamak yaygın bir uygulamadır. Tekerleklerden gelen eğrilerin yarıçapı, koordinat sistemine bağlı olarak pozitif ve negatif değerler olabilir. Kartezyen koordinat sistemi ile sola dönme pozitif, negatif bir yarıçapa doğru dönüş ve ICC'yi ileriye doğru itmek yaklaşık olarak sonsuza kadar gidiyor. ICC sonsuza yakınsa, tüm tekerleklerin dikey çizgileri paraleldir. Bkz. Şekil 4.1.

Dönüş merkezi, maksimum ve minimum değer açısına sahiptir ve bu da her bir tekerleğin mümkün olan en küçük eğrinin yarıçapını sıfırdan büyük yapar. Bu da, Ackerman yapılandırma robotlarının tek bir noktada, yani diferansiyel konfigürasyonun robotlarıyla mümkün olduğu kadar dönmesini imkansız kılar. Bkz. Şekil 4.2.



Şekil 4.2. (a) Diferansiyel direksiyona karşı (b) Ackerman direksiyon

Ackerman yapılandırma robotları, kullanılan hareket komutları arasında bir dönüşüm kullanarak diferansiyel konfigürasyon robotlarıyla uyumludur. Ackermann yapılandırma robotları bir hız ve direksiyon açısı koruyan mesajlar aldığından, her iki konfigürasyonda kullanılan hareket komutlarının yapısı hakkında bilgi gereklidir, bu mesajlar u_A olarak tanımlanır.

$$u_A = \begin{pmatrix} v_A \\ \varphi_C \end{pmatrix} \quad (4.1)$$

Diferansiyel konfigürasyon robotları genellikle altı parametre, üç doğrusal hız parametresi ve üç açısal hız parametresi içerebilen rotasyon mesajları kullanır. Diferansiyel konfigürasyon robotu iki boyutlu bir boşlukta üç DOF'ye (Degrees of Freedom – Serbestlik Derecesi) sahiptir; ancak hareketlerini tanımlamak için sadece iki parametre gereklidir. Dolayısıyla, dönüştürme mesajlarının doğrusal hızı ve açısal hızı kullanılır. Bunlar x

yönünde doğrusal bir hız v_D ve z ekseninde etrafında bir açısal hızdır ω_D . Bu tür rotasyon mesajları, diferansiyel konfigürasyon robotları için kullanılması nedeniyle u_D olarak tanımlanır.

$$u_D = \begin{pmatrix} v_D \\ \omega_D \end{pmatrix} \quad (4.2)$$

Diferansiyel tahrik hareketinin lineer hızları ve Ackermann tahrik hareketi eşitliği ile $u_A = u_D$ hız v olarak belirlenir; Daha sonra Ackerman'ın direksiyon açısını hesaplamak için öncelikle diferansiyel konfigürasyon robotlarının hareketine dayalı olarak eğrinin yarıçapını hesaplanmalıdır, bunun aks merkezleri aralığını $w_{\omega b}$ bilmek zorunludur.

$$\varphi_C = \arctan\left(\frac{w_{\omega b}}{R_C}\right) \quad (4.3)$$

Olduğu yerde:

$$R_C = \frac{v_D}{\omega_D} \quad (4.4)$$

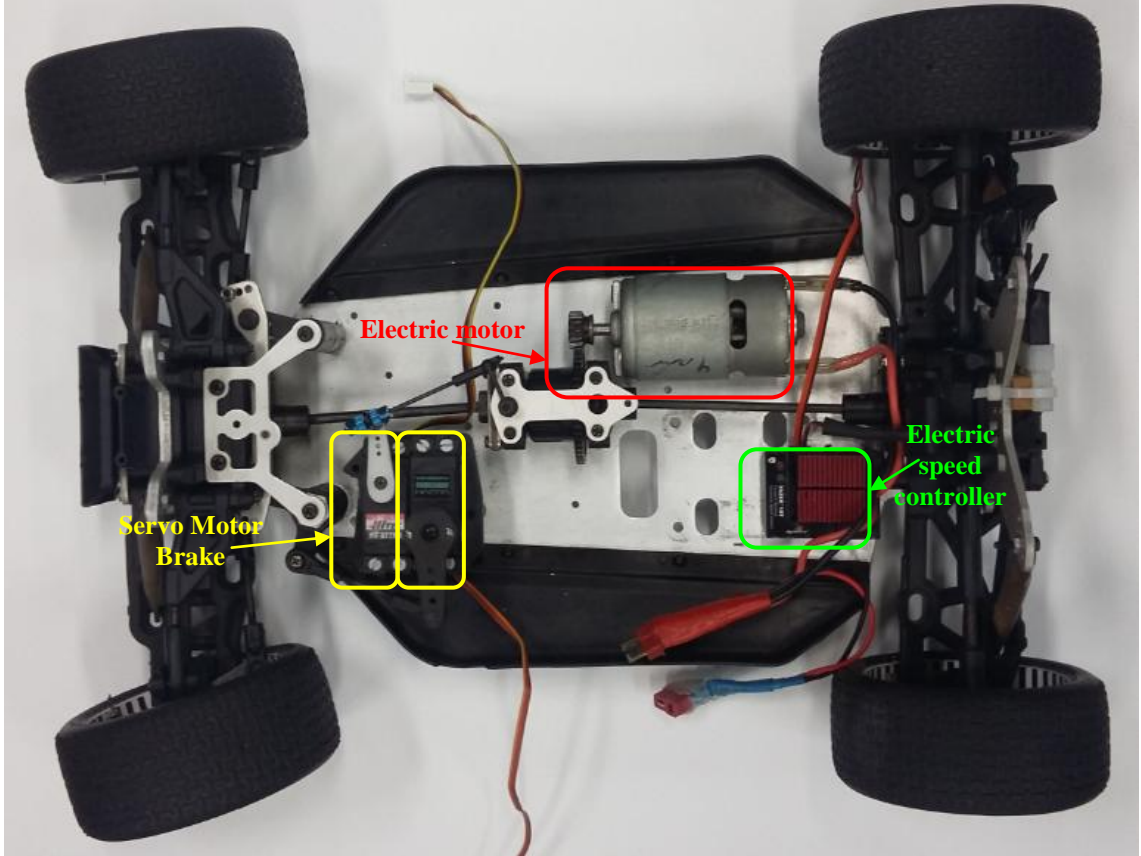
ve

$$-\frac{\pi}{2} < \varphi_C < \frac{\pi}{2} \quad (4.5)$$

Normalde dönüş açısı o kadar geniş değildir; Bu nedenle eğer dönüş açısı direksiyon aracının verdiği maksimum değerden daha büyükse veya direksiyonlu araç tarafından verilen minimum değerden düşükse, φ_C maksimum veya mümkün olan en düşük değere ayarlanır.

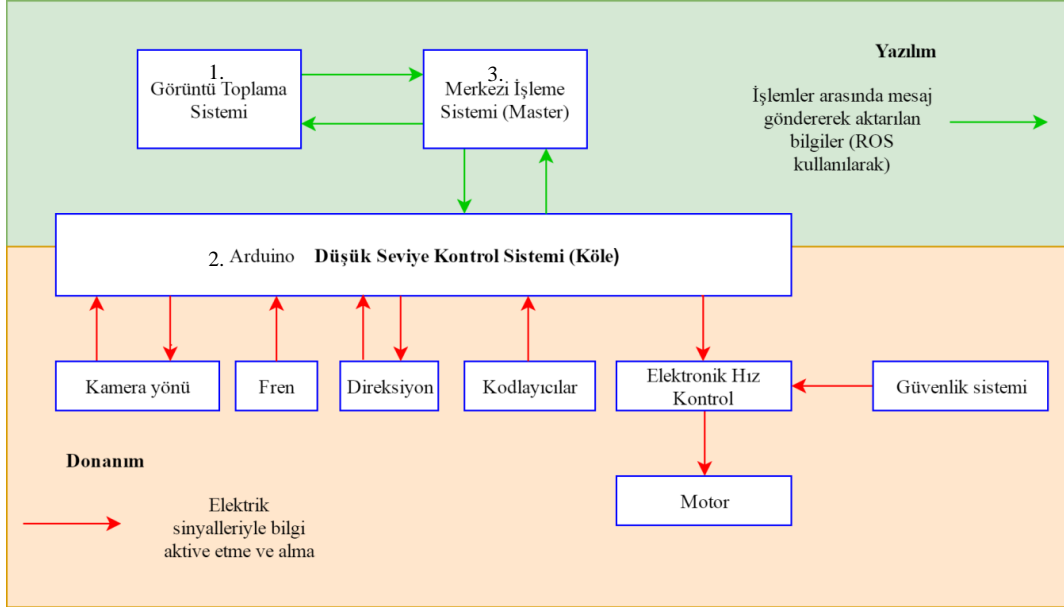
4.1. Robot Tabanı

Kullanılan temel araç, 1:8 ölçekli içten yanmalı motor kullanan bir Buggy modelinin şasisidir. Bu araç, orijinal şeklinde, elektrik tahrik sistemine geçirilen bir yanma itme sistemi içermektedir. Bkz. Şekil 4.3.



Şekil 4.3. Bileşenleri olan aracın modeli

Araç sisteminin genel kontrolü, birbirleriyle iletilen üç alt sisteme bölünmüştür; bunlardan ilki, yalnızca araçtaki iki kamera hakkında bilgi yakalama (bir zemin görünümü kamera ve görüntü sinyalleri kamerası) ile ilişkilidir; İkincisi mekanik ünitelere gönderim ve kontrol sinyalleri ile ilişkilidir, sonuncusu üst düzey karar verme ve kontrolden sorumlu merkezi sistemdir Bu üç sistem arasındaki iletişim, bağımsız işlemler arasında mesajlar yoluyla iletişimi sağlayan ve farklı cihazlarda farklı programlama dillerine izin veren araçları içeren bir sistem olan ROS (Robot Operating System) sistemini temel almaktadır. (Neto, 2014). Bkz. Şekil 4.4.



Şekil 4.4. Genel kontrol ve bilgi akış sistemi

4.1.1 Donanım

Diyagramdan, sistemin iki seviyede bölündüğü görülmektedir. Daha düşük seviyede, hız, yön ve fren gibi mekanik sistemlerin birimleri ve bazılarının geribildirim bilgileri bulunmaktadır. Ayrıca, bu sistemin yönetimi, tahrik motorları ve servo motorlar vasıtasıyla, yukarıda belirtilen tüm mekanik üniteleri kontrol eden, doğrulama kafesinin mekanik geri besleme sistemini toplayan ve Kontrol ünitesinin bir kısmını kapatan düşük seviye bir kontrol sisteminden oluşmaktadır. Bu kontrol seviyesinin üstünde, aracın üst düzey kontrolü olan merkezi işlem sistemi (master) vardır.

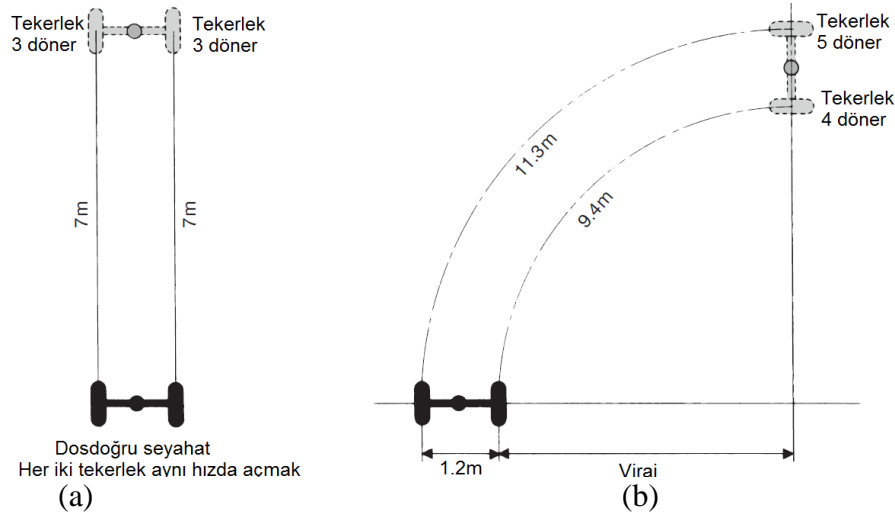
Ana alt sistem almaya ilişkin tüm bilgileri içerir ve bu alt sistem bilgiyi düşük seviye kontrol sistemine gönderir. Görüntü toplama sistemi ana alt sistem ile ilişkilidir. Bu fiziksel sistem, yakalanan bilgilerin tümünü kontrol etmeye izin verir. Ayrıca, olası kazaları önlemek için tasarlanmış bir güvenlik sistemi bulunmaktadır.

4.1.2 Mekanik sürme sistemi

- **Hız kontrolcüsü**

Hız kontrol cihazı, 7.2v'lik bir elektrik motoru ve bir Elektrikli Hız Kontrol Cihazı ESC kullanılarak uygulanmaktadır. Bu kontrol cihazı, diferansiyel bir sistemle motordan güç aldığıında, direksiyon simidinin farklı devirlerde döndürülmesine izin vermek için kullanılır.

Viraja girilmesi durumunda farklı hızlarda hareket eden tekerlekleri dengelemek için Şekil 4.5'de gösterilen diferansiyel sistemi kullanılmaktadır, çünkü viraja girme sırasında virajın dış tarafında kalan tekerler iç tarafında kalan tekerleklere göre daha uzun bir mesafe katetmektedir, hareketin sorunsuz bir şekilde tamamlanması için dış tarafta olan tekerleklerin iç taraftaki tekerleklere göre daha hızlı olması gerekmektedir. (Walter Hillier & Thornes, 2012)



Şekil 4.5. Konumlarına göre tekerleklerin taşınması (a) Düz yol (b) Eğik yol

- **Direksiyon kumandası**

Bu kontrol cihazı, tekerleklere yöneltilen dönme dairesi merkezinin merkezine dayalı Ackerman mantıksal aracını kullanarak, her bir tekerleği eğriye teğet doğru ile

tutarak araçla eğri yapmaya olanak tanır. İç tekerlek (eğrinin yarıçapına göre) tekerleklerin eğrilik merkezine göre aynı açığa sahip olmasını sağlayan dış tekerleğe göre daha büyük bir eğim açısına sahiptir.



Şekil 4.6. Servomotor S60MH

Direksiyon kontrol sistemi, S60MH servo motorunun 3 kg / cm'lik bir torkla kullanıldığı Darbe Genişlik Modülasyonu (Pulse Width Modulation,PWM) tarafından kontrol edilen açısal konuma dayanır. Teknik özellikler Tablo 2'de açıklanmaktadır. Bkz. Şekil 4.6.

Çizelge 4.1'deki Teknik Özellikler:

Çizelge 4.1 Servomotor S60MH Teknik Özellikleri

Cihaz	S60MH
Boyut	Araba Standartı
Tip	Analog
Tork	6,2kg @ 4,8V
Hız	0.15 Sec @ 4,8V
Diş	Metal
Zahnkranz	25 Z
Jak Takımı	JR
Uzunluk	40,6 mm

Yükseklik	40,8 mm
Genişlik	20,5 mm
Delik Mesafesi	49,7mm
Gerilim	4,8V - 6,6V
Ağırlık	52g

- **Fren kontrolü**



Şekil 4.7. Servomotor HS-322HD

Temel fren sistemi için, iki aksın ayrı bir kilitleme diski üzerinden ayrı ayrı çalışan, servomotor tarafından tahrik edilen bir kol sistemi vasıtasıyla durdurma gerçekleştiren orijinal araç freni kullanılır. Tipik sistemlerde, bu formattaki araç freni, aracın ön aksı ve arka aksı bağımsız olarak iki kilit içerir. Bu durumda, aracın hareketsiz kılınması için yeterli olduğu için sadece ön aks uygulanmaktadır. Ön diski kilitleyen kolun konumunu kontrol etmek için kullanılan servo motor HS-322HD'dir. Bkz. Şekil 4.7.

Çizelge 4.2'deki Teknik Özellikler:

Çizelge 4.2 Servomotor HS-322HD Teknik Özellikleri

Cihaz	HS-322HD
Boyutlar	1.57" x 0.78" x 1.43" (39.88 x 19.81 x 36.32mm)
Ürün Ağırlığı	1.52oz (43g)
Çıkış Mili Stili	24 tooth

Gerilim Aralığı	4.8V - 6.0V
Yük Olmayan Hız (4.8V)	0.19sec/60°
Yük Olmayan Hız (6.0V)	0.15sec/60°
Durma Torku (4.8V)	42 oz/in (3.0 kg/cm)
Durma Torku (6.0V)	51 oz/in (3.7 kg/cm)
Maks. PWM Sinyal Aralığı(Standart)	553-2450µsec
µs Başına Seyehat (kutudan çıktığında)	.106°/µsec
Maks. Seyehat (kutudan çıktığında)	201°
Darbe Genliđii	3-5V
Çalışma Sıcaklığı	-20°C to +60°C
Akım Boşaltımı- bořta (4.8V)	7.4mA
Akım Boşaltımı- bořta (6.0V)	7.7mA
Akım Boşaltımı- yüksüz(4.8V)	160mA
Akım Boşaltımı- yüksüz(6V)	180mA
Sürekli Deđiřtirilebilir Döndürme	Evet
PWM Sinyalini Arttırma Yönü	Saat yönü
Deband Genişlik	5µs
Motor Tipi	Cored Metal Brush
Potansiyometre Tahrik	4 Slider/Direct Drive
Geri Bilgilendirme Stili	5KΩ Potentiometer
Çıkış Mili Destek	Top Resin Bushing
Diřli Tipi	Straight Cut Spur
Diřli Malzemesi	Karbonite
Tel Uzunluđu	11.81" (300mm)
Tel Ölçer	25AWG

4.1.3 Düşük seviye kontrol sistemi

Düşük seviyeli sistem, ana ünite ile fiziki ünite arasında çalışır. Bu sistem geçerli değerleri araçtan toplar (hız, tekerlek dönüş açısı, kamera direksiyon açısı ve fren), sürücülerin kontrolünü yapar ve aktüatör durum bilgisini (hız kontrol cihazı, servo motor açısı kontrol tekerleđi, kamera ve fren) periyodik olarak döndürür.

Mikro denetleyici (Arduino) ile bağımlı birim sistemi, dört adet tekerleğin enkoder darbelerinin alınabilmesi için kullanılır, bu nedenle cihaz dört giriş sinyalini işleme kabiliyetine sahip olmalıdır. Bu çalışmada, altı giriş sağlayan "Mega 2560" modeli kullanılır. Mega 2560, aşağıdaki teknik özelliklere sahip ATmega2560'a dayanan bir mikro denetleyici kartıdır. Bkz. Şekil 4.8.

Çizelge 4.3'deki Teknik Özellikler:

Çizelge 4.3 Arduino Mega 2560 Teknik Özellikleri

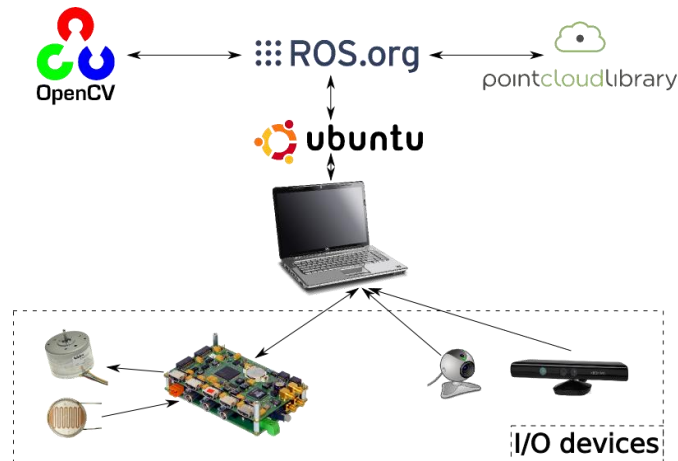
Mikro Kontrol Denetleyici	ATmega2560
Çalışma Gerilimi	5V
Giriş Voltajı(Önerilen)	7-12V
Giriş Voltajı (limit)	6-20V
Dijital I/O Pimleri	54 (bunlardan 15 tanesi PWM çıkışı sağlamaktadır)
Analog Giriş Pimleri	16
I/O Pimi Başına DC Akımı	20 mA
3.3V Pin İçin DC Akımı	50 mA
Flash Bellek	256 KB 8 KB bootloader tarafından kullanılır
SRAM	8 KB
EEPROM	4 KB
Saat Hızı	16 MHz
LED_BUILTIN	13
Uzunluk	101.52 mm
Genişlik	53.3 mm
Ağırlık	37 g



Şekil 4.8. Arduino Mega 2560

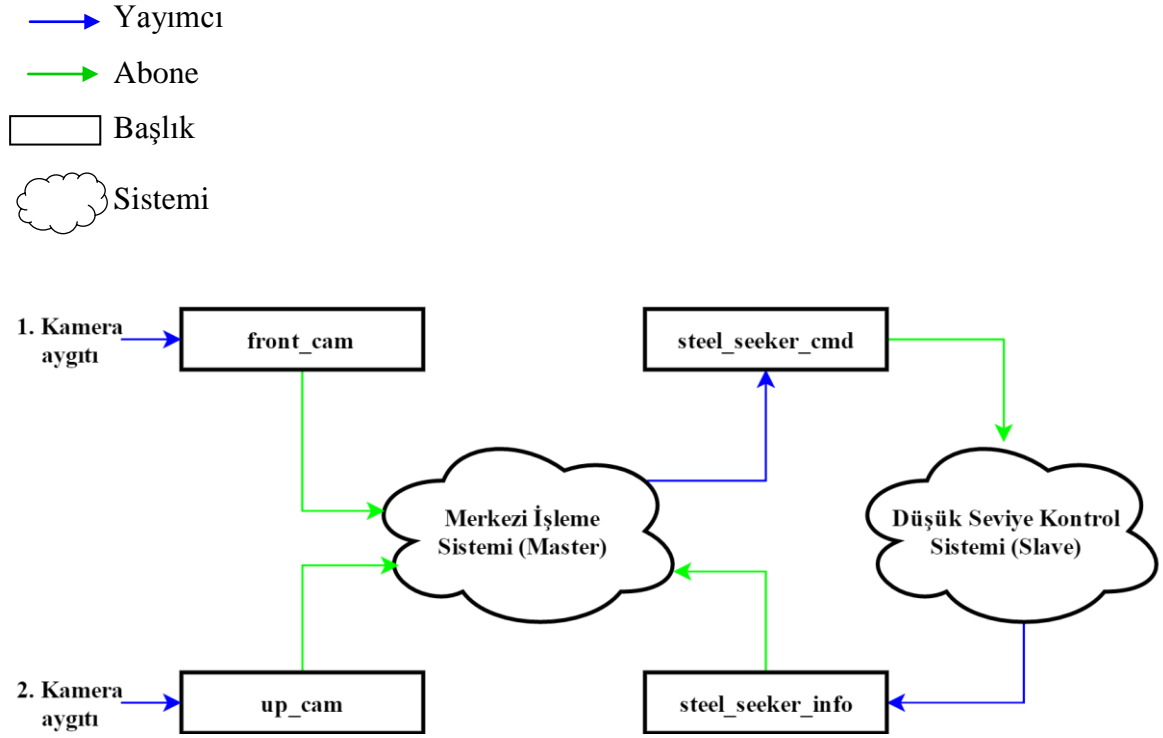
4.1.4 Düşük seviye kontrol yazılımı

Alt düzey kontrol yazılımını ana bileşen olup, bağımlı olarak yürütülen program, yani Arduino'dur. Genel olarak bu program, her girdi için ana kayıt tarafından istenen durumla ilgili bir dizi veri alır, karar bilgilerini temel alarak birimlerin durumunu değiştirir ve kontrol eder ve mevcut bilgileri geri bildirir. Bu iletişim ROS mesajlaşma sistemi ile yapılır. ROS, donanım soyutlaması, düşük seviye kontrol ekipmanları, işlemler arasındaki mesaj yönetimi ve paket kontrolü gibi standart işletim sistemi servislerini desteklemektedir. (ROS Organization) Bkz. Şekil 4.9.



Şekil 4.9. Bilgisayar görüntüsü amacıyla bir masaüstü ROS robotunun kavramsal taslağı (Into Robotics, 2014)

İşlemler, algılayıcılar, çalıştırıcılar, kontroller vb. ile ilgili bilgi alan, ileti alabilen veya gönderebilen bağımsız uygulamalara dayalıdır. İletiler(mesajlar) konular halinde mevcuttur dolayısı ile aboneler tarafından (kayıtlı kullanıcılar) okunabilirler. Bu sistemin önemli bir avantajı, mesajların ayrı makinelerde ve donanımda çalışan süreçler de dahil olmak üzere tamamen bağımsız süreçlerle değiştirilebilmesidir. Bu bölümde kullanılan sistem, mesaj göndermek ve ROS seri bağlantıları yoluyla mesajlar almak için kullanılan protokol olan roserial'in uygulanmasına dayanıyor. Şekil 4.10, bilgi değişiminin temel bir planının yanı sıra kendisine yayın yapan / abone olan kurumları ve kullanılan konuları göstermektedir.



Şekil 4.10. Bilgi değişim Şeması

- **Hız kontrolü**

Hız kontrolü, düşük seviye kontrol sisteminde daha fazla kaynak harcayan şeydir. Her bir tekerlek bir kare dalga ile ilişkilendirildiğinden, sistemin her bir tekerleğe uygulanabilen en yüksek hızda bir dönüş esnasında üretilen tüm geçişleri yakalaması gereklidir. Bunu yapmak için sistem, durum sinyalinin 0'dan 1'e (0 V'dan 5 V'a) veya 1'den

0'a deđiřtiđini gstermek iin program gncellemelerine yanıt vermelidir. Bylece, bir sre boyunca ka atımın sayıldıđını da kontrol etmek mmkn olur. Ayrıca, belirli bir zaman zarfında ka atımın sayıldıđını kontrol ederek her tekerleđin hızını elde edebilirsiniz.

Fark sistemi temel alınarak, direksiyon hesaplama modeli, iki tekerlek arasındaki ortalama dnř aısını ve tahrik tekerleklerinin ortalama hızını kullanan dođrusal hızı gznnde bulundurur. Kinetik modeli kullanarak, aracın aısal ve dođrusal yer deđiřtirmesine karřılık gelen ařađıdaki denklemler elde edilebilir (Dudek & Jenkin, 2011):

$$\tan(\alpha_{obt}) = \frac{d_{btw}}{I_{cc}} \quad (4.6)$$

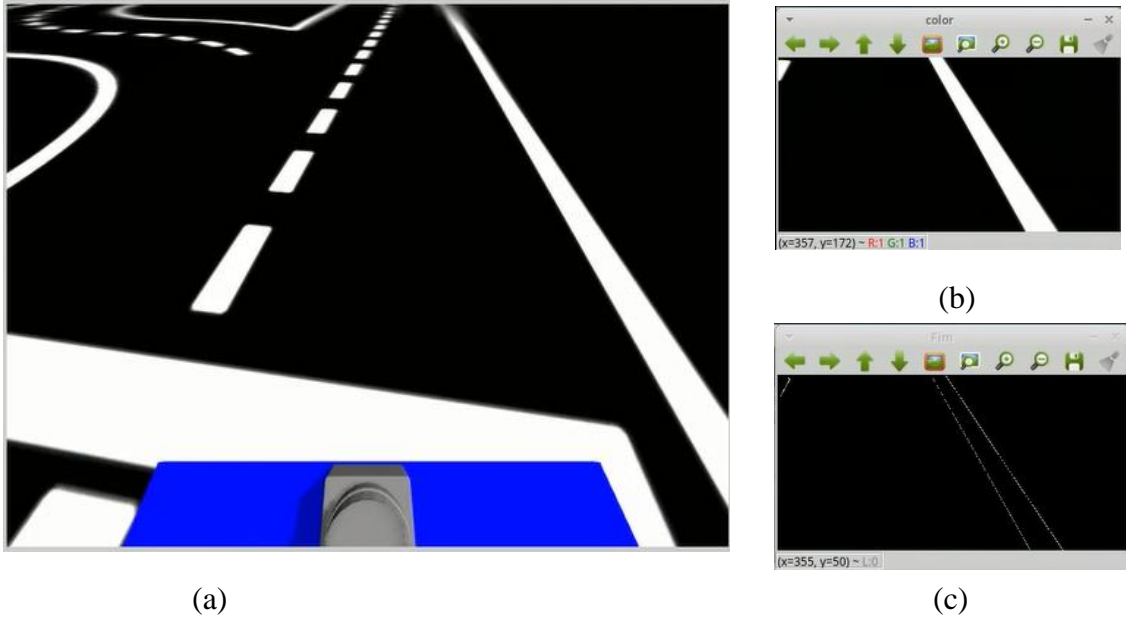
$$\left\{ \begin{array}{l} \Delta_{\theta} = \frac{1}{d_{btw}} \int_0^t v_{obt}(t) * \tan(\alpha_{obt}) dt \\ \Delta_x = \int_0^t v_{obt}(t) * \cos(\theta) dt \\ \Delta_y = \int_0^t v_{obt}(t) * \sin(\theta) dt \end{array} \right\} \quad (4.7)$$

$$\downarrow$$

$$\left\{ \begin{array}{l} \Delta_{\theta} = \frac{v_{obt}(t) * \Delta_t * \tan(\alpha_{obt})}{d_{btw}} \\ \Delta_x = v_{obt}(t) * \Delta_t * \cos(\theta) \\ \Delta_y = v_{obt}(t) * \Delta_t * \sin(\theta) \end{array} \right\} \quad (4.8)$$

Yukarıdaki denklemlerde v_{obt} ve α_{obt} sırasıyla, aracın duysal sistemi tarafından elde edilen dođrusal hıza ve yne karřılık gelir; d_{btw} eksenler arasındaki mesafeye karřılık gelir; I_{cc} arka tekerleđin sanal merkezinden Anlık Eđrilik Merkezine (ICC - Center of Curvature) uzaklıktır; Δ_{θ} , Δ_x ve Δ_y , θ 'daki aısal ve dođrusal yer deđiřimi olup, burada X ve Y, sırasıyla global ekseninde en son gncellenen mesafe karřılařtırmalarıdır ve buradaki Δt ise her bilgi alımında geen zamanı gstermektedir.

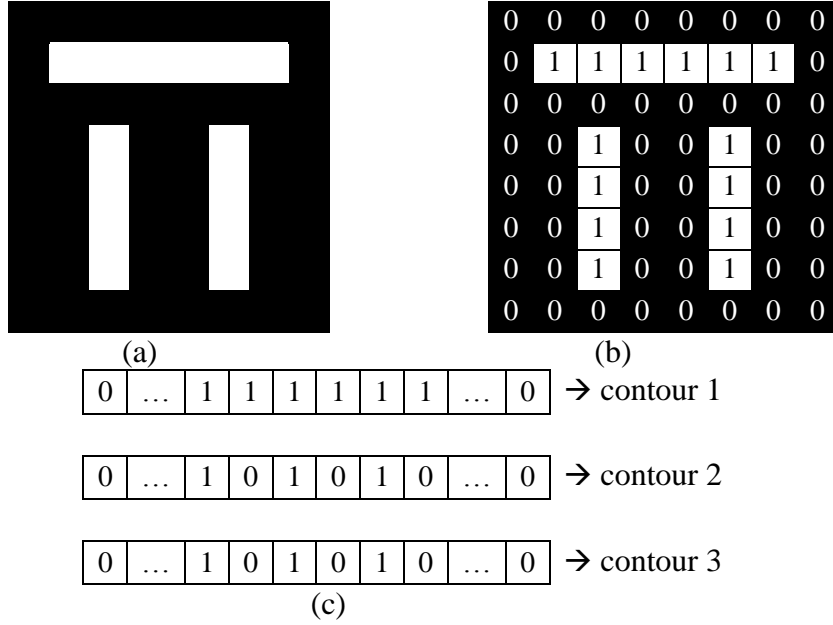
Ara direksiyonu iin tekerleklerin sola ve sađa maksimum dnř deđeri tespit edildi, burada bir referans deđeri dnř aısı 0 olarak alındı, mevcut dnme aısının deđeri maksimum tahmini deđer iinde deđerler alıyor, bu durumda maksimum deđer 45 derece. Bkz. Őekil 4.11.



Şekil 4.13. (a) Orjinal görüntü (b) Seçilen görmüş segment (c) İşlem görmüş segment

Beklenen tepki, aracın yolun ortasında tutulması ve önceden belirlenmiş bir değerle (piksel cinsinden) kontrol edilir, eğer değer daha düşükse, aracın rotayı sağ çizgiye en yakın yaptığı anlamına gelir ancak daha yüksekse çizgiden uzaklaşacaktır, bu nedenle en iyi sonucun elde edildiği ortalama değeri belirlemek için farklı değerlerle test yapılması gereklidir. Bkz. Şekil 4.13.

Görüntüdeki çizgileri algılamak için, OpenCV'de kontur takip algoritması (Suzuki & Abe, 1985) uygulanmaktadır. Giriş dizisi, sıfır olmayan piksellerin 1 olarak kabul edildiği ve sıfır piksellerin 0'ın kaldığı ikili bir görüntü kullanılır ve her konturu bir noktanın depolanmış bir vektörü olarak döndürür ve örnek, şekil 4.14'de gösterilmiştir.



Şekil 4.14. Kontur takip eden algoritma örneği (a) Orijinal resim (b) İşlem görmüş olan görüntü (c) Noktaların kaydedilmiş vektörü

- **Ön kamera yön kontrolü**

Kontrolörlerin geribildirim sistemi olmasına rağmen, sistem bir açık döngü kontrol sistemi görevi görür. Sistem istenen yönü alır ve Darbe Genişlik Modülasyonuna (PWM - Pulse Width Modulation) karşılık gelen sinyali ilgili servomotora verir. Tüm bu işlem gerçekleştirilir ve ana program döngüsü herhangi bir kesintiye bağlı değildir. Aynı döngü servomotorun mevcut yöneliminin analog değerini lineer bir ilişki yoluyla toplamaktadır.

- **Fren kontrolü**

Fren servo motorunun kontrolü için, aktif ve aktif olmayan fren olmak üzere yalnızca iki pozisyon tespit edilmiştir. Master tarafından gönderilen mesaj, yalnızca aktif (değer 1) veya devre dışı (değer 0) freni tanımlar. Bağımlı servomotorun istenen konuma karşılık gelen PWM sinyalini gönderir ve gönderilen mesajdaki bu değeri master'a ekler, değişiklik talebi mevcut durum dikkate alınmamasına rağmen alınmıştır, yani mevcut durum gereksinimle aynı olmasına rağmen bilgi gönderilir.

4.2. Üst Düzey Denetim

Master'ın ana işlevi kameralar ve aktuatörlerin durumu hakkında bilgi toplamak ve otonom sürüşe yönelik algoritmalar uygulayarak yörüngenin değişiklikleriyle ilgili kararlar almaktır.

4.2.1 Görüş sistemi – takip

Sürüş kararlarının alınması için bir referans noktası olan görüş sistemi, etrafındaki hattın sağ çizgisine dayanır. Bu şekilde, araç referans çizgisinin bulunduğu mesafeyi doğrulamak için kamerasını kullanır; böylece tekerleklerin yönünü kontrol eder.

Sistem, hattı belirlemek ve bilgi işlemeyi daha basit yapmak için çekme alanını ana ilgi alanına odaklanacak şekilde sınırlar. Görüntü direkt olarak parlaklık bileşeni içine alınır ve daha sonra siyah yol üzerindeki beyaz işaretleri tanımak için ana unsur olarak yerleştirilir.

Görüntü yakalama işleminde konturların toplanması işlemi uygulanır; bu işlem, renk sınırlarının değiştiği bölgeler olan pikselleri tanımlar; bu, yoğunluk varyasyonlarını algıladığında, varyasyon daha büyük olduğunda değerinin daha büyük olmasını sağlar.

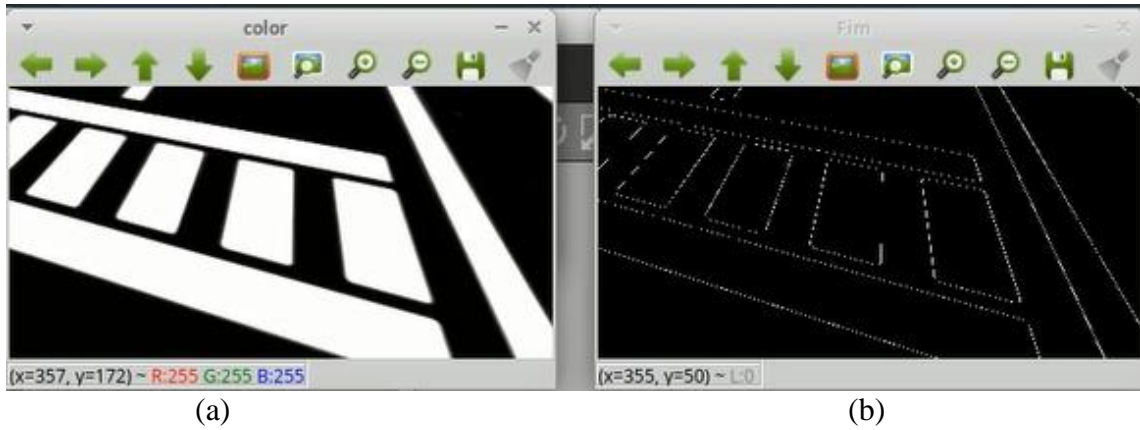
Daha sonra, ana ilgi bilgilerini içerdiğinden, resmin sağ alt köşesindeki bilgiler dikkate alınır, böylece bir eşik değeri belirlenir. Ardından konturun kabul edilebilir bir minimum uzunluğa sahip olup olmadığını, yani bu uzunluğu bir kılavuz olarak kabul edilebilir bir minimum değer olup olmadığını kontrol eder.

Son olarak, kameranın merkezinden seçilen kenarın ortalama değerine olan uzaklığı doğrulanır (başlangıç ve bitiş sütunları arasında). Mevcut sapmayı telafi etmek için direksiyon açısını değiştirecek arzulanan mesafe ile gerçek mesafe arasındaki fark hatayla ilişkilendirilir.

4.2.2 Görüş sistemi - yaya geçidi (Zebra) tespiti

Zebra tanıma için kullanılan kamera, izleme yolunu tanımlayan kamerayla aynıdır. Bu nedenle, yüksek aydınlık alanlarını tanıma süreci esas alınır ve elde edilen görüntü, yolların tanımlanması için kullanılır. Zebra görüntülendiğinde, resimdeki mevcut beyaz piksellerin sayısı önemli derecede arttığı için tanımlanması mümkündür.

Görüntüye beyaz piksel sayısı uygulanır ve zebanın mevcut olduğu düşünülen bir değer atanır, böylece çevrimlerin sayısı, bir zebanın tanınması doğru olduğunu saptamak için sayılabilir ve bir zebra ile karıştırılabilen dağınık beyaz piksellerin tanımlanması mümkün değildir, sonunda zebanın görünür olmadığını ve referans çizgisinin tekrar başlatılabileceğini teyit eder. Bkz. Şekil 4.15.



Şekil 4.15. (a) Zebanın saptanması (b) Zebanın uygulanması

4.2.3 Görüş sistemi - tünel

Varolan nesnelerin seçilmesi, kamera tarafından çekilen görüntüde yüksek bir parlaklığa sahip olduğu ve daha sonra, zeminin görünümü, aracın rotayı tanımlamasına ve onu tünelde izlemesine olanak tanıdığından, tünelin iç kısmı için önemli farklılıklar yoksa çok fazla değişiklik gerekli değildir. Bkz. Şekil 4.16.

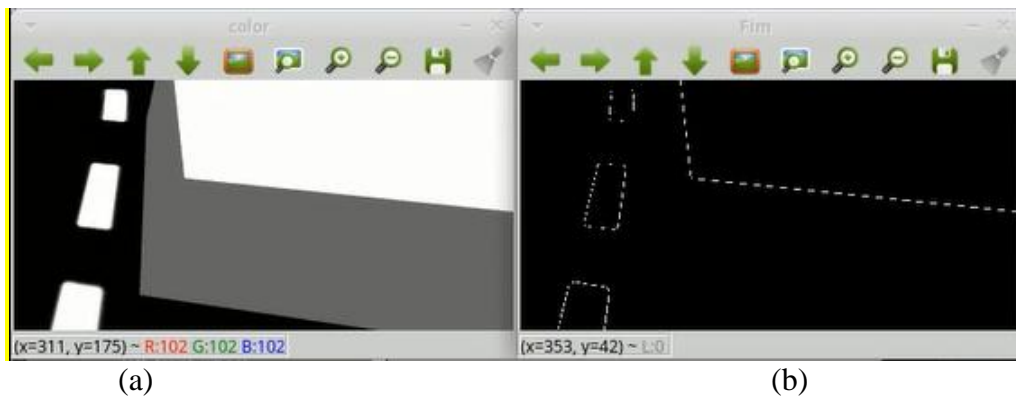


Şekil 4.16. Tünel içinde görüş. Şekil 4.3. (a) Tünelnin saptanması (b) Tünelnin uygulanması

4.2.4 Görüş sistemi - engel tanıma

Engel belirleme sisteminde renk temel alınmıştır. Bir engelin varlığı, yol boyunca gösterilen parlaklıktan daha yüksek bir parlaklık olan bir nesne belirlendiğinde, yani siyah beyaz resimde çok daha fazla sayıda beyaz pikselin bulunduğu zaman belirlenmiştir. Aynı zamanda zebra varlığını belirlemede gerekli olduğu için, piksel eşikleri testlerle göz önüne alınmıştır. Bkz. Şekil 4.17

Engeli tanımlarken engeldeki hatlar bulunmuştur ve engelden geçmek için kontur takip algoritması yeniden uygulanmıştır.



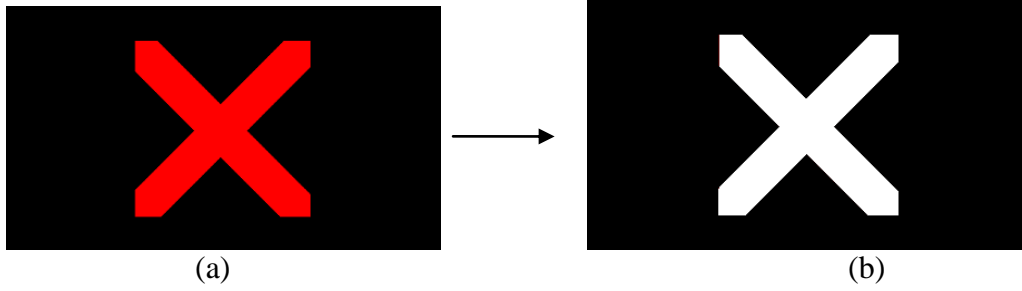
Şekil 4.17. Bir engelden kurtulan görüş alanı. (a) Engelin saptanması (b) Engelin uygulanması

4.2.5 Görüş sistemi - trafik ışığının tanınması

Trafik sinyalinin görüntülenmesi üst kamera ile yapılır. İlk önce çıkış sinyalini tanır, bu nedenle aracın çalıştırılabileceğini gösteren yeşil bir sinyalin algılanması gerekir.

Başlangıç noktasındaki sinyal değişimini tanımlamak için yalnızca trafik ışığı bilgisini içeren görüntü alanı hesaba katılır, ayıklanan görüntünün kırmızı bir bileşeni bulunur ve görüntüdeki ortalama değerleri hesaplar. Görüntü değiştiğinde, daha önce tanımlanan eşikte bir değişiklik olur; bu, sinyalin değiştiği anlamına gelir ve bu nedenle bir yetkilendirme sinyali belirtir.

Sinyal ne olursa olsun, devreye alma sinyalinin analizi için, panelin varlığının konumunu (araç hareket halindeyken marşda bir kez başlatıldıysa bu zaten tanımlanacaktır) parlak olmayan pikselleri araştırmak gerekir; arka plan büyük ölçüde karanlık kalır. Görüntü parametrelerini bileşenleriyle RGB (red blue green - kırmızı mavi yeşil) kullanarak ve her sinyalin renk eşiklerini araştırarak, `avgPixelIntensity` OpenCv aracını kullanarak farklı sinyalleri tanımlamak mümkündür. Bkz. Şekil 4.18.



Şekil 4.18. (a) Orijinal sinyal trafiği (b) Trafik sinyallerinin uygulanması.

4.3. Temsili Robot

Robotun benzetimini kolaylaştırmak için, fiziksel özellikleri 3 Boyutlu bir görüntüye eklemek ve böylece işlenecek dosyaların boyutunu arttıran bir kod satırı kullanarak robotun tüm yönlerini tanımlamak mümkün değildir ve hata olması durumunda gözden geçirmek zordur.

3D tasarım yapmak için birçok basit araç vardır, bu durumda tasarım aracı Tinkercad hayata geçirilmiştir. Bu araç, düzenleyiciye eklenebilen veya içe aktarma görüntüleri kullanabilen önceden tasarlanmış geometrik modüllerin eklenmesine izin verir; Tasarım tamamlandığında, .STL formatında 3D görüntü olarak dışa aktarılabilir. (Gazebosim Organization, 2014)

Robot özelliklerini dikkate alarak, gerçek modelin genişliği, uzunluğu ve yüksekliği (Şekil 4.19), 3D düzenleyicide şasinin oluşturulması için yürütüldü:



Şekil 4.19. Yarışma için kullanılan robot

Tekerlekler de kodu basitleştirildiğinden simülasyona 3D görüntü ile eklendi. Robotun tanımını gerçekleştirmek için URDF (Unified Robot Description Format - Birleşik Robot Tanımlama Formatı) paketi robotun farklı özelliklerini içeren xacro dosyalarını kullanarak oluşturulmuştur. (Gazebo Organization, 2014)

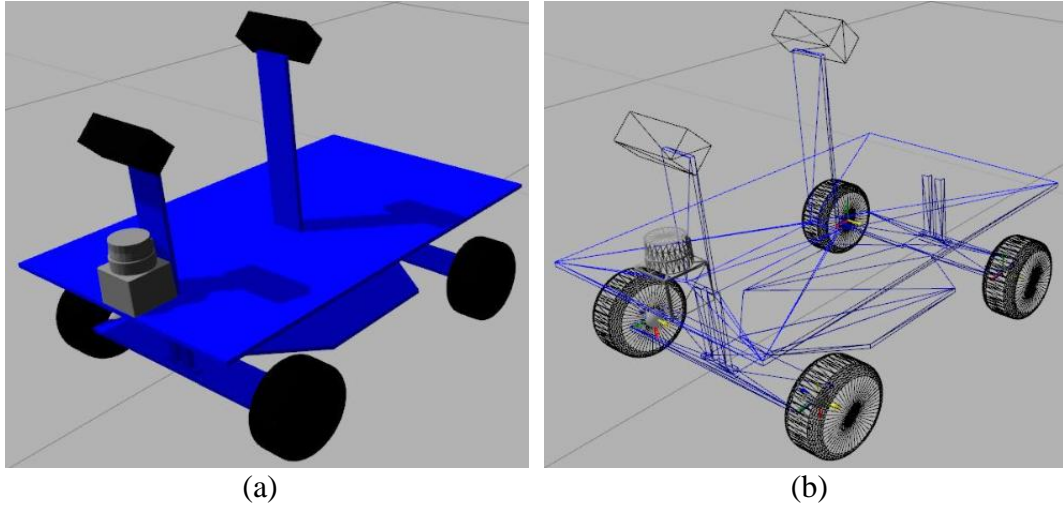
Renklerin tanımı, Gazebo'yu ROS ile arayüz için örnek bir robot ve kod olan materials.xacro dosyasında kullanıldığı için ros-simulation / gazebo_ros_demos paketinde bulunmaktadır.

Ackermann.gazebo, temas yüzeyi boyunca temel temas yönleri için sürtünme katsayıları μ ve temas rijitliği k_p ve rijit vücut temasları için sönümlenme k_d gibi hem

Gazebo referanslarını içerir, hem de ODE (Open Dynamics Engine - Açık Dinamik Motor) içerir. Ayrıca Gazebo, modele daha fazla işlevsellik kazandırır ve sensör, motor ve kameralar için ROS mesajlarında ve servis çağrılarında bağlantı kurar.

Macros.xacro'da 3D görüntülerin dahil edildiği atalet, iletim ve geometri olarak makro parametreleri eklendi.

Sonunda ana dosya ackermann.xacro, robotun tüm özelliklerini ve benzetim ortamının yerinin ayrıntılarını yapılandırır ve kalan xacro dosyalarını çağırır. Sonuç Şekil 4.20'de gösterilmektedir.



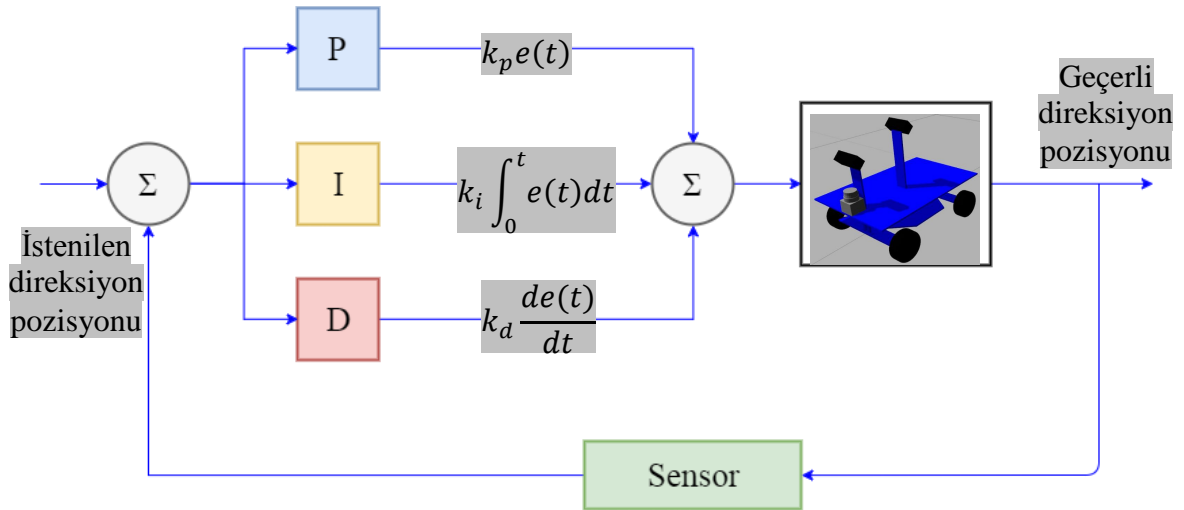
Şekil 4.20. Benzetimli gerçek modelin şeması, (a) Normal görünüm (b) Tel kafes görünümü

Robotu çalıştırmak için bir başlatma dosyası kullanıldı, burada robot dizaynı, ortak devrelere yayınlayan düğümler tarafından kullanıldı ve durum yayıncı düğümünün eklem konumlarını ortak durumlardan okuması için ekledi ve aracın durumunu tf'ye yayınladı.

Gazebo ve ROS arasındaki iletişimi sağlamak için, Effort Ortak Arabirimi, Pozisyon Eklem Arayüzü veya Hız Ortak Arabirimi, gibi ortak kontrol yöntemleri olan libgazebo_ros_control.so eklentisi kullanıldı; bunların tümü PID denetleyicilerini kullandı; Bunların konfigürasyonu .yaml dosyalarında açıklanmıştır; bu dosyalar, tekerlekleri ve

amortisörleri hesaba katabilecek direksiyon aracı kontrolörleri için ortak kumandaları ve parametreleri tanımlamaktadır.

YAML, ROS parametreleri için yaygın olarak kullanılan bir işaretleme dilidir. Parametre sunucusunda ROS parametrelerini ayarlamak için yaml-kodlayıcı dosyalarını kullanmak uygundur, bu uygulama için yaml dosyaları yukarıda açıklandığı gibi ortak denetçi konfigürasyonu için yaratılmış ve bunlar ana başlatma dosyası tarafından yüklenmiştir. Denetleyici türü, birleşim durumu denetleyicisi için olduğu kadar her tekerlek ve direksiyon mafsalı denetleyicisi için de tanımlanmıştır. Tüm denetleyicilerin de PID kazançları tanımlanmıştır. Bu PID kazançları, her tekerleğin ve şanzımanın kontrolüne ayarlanmış olup, bu da şanzıman her tekerlek için uygulanan hızın doğru hesaplanmasını ve dolayısıyla sistemin diferansiyel sistemi olarak işlev görmesini sağlar. Bkz. Şekil 4.21.



Şekil 4.21. Gazebo ile Gerçeklenen PID Kontrolör Şeması

Kullanılan PID denetleyicisi hız ve konum kontrolü döngüsünün üstünde çalışan bir konum kontrol döngüsü düşünür. Varsayılan değerler ROS ile kullanılmıştır: K_p , K_i ve K_d için 1.0, 0 ve 0. Her bileşen yaml dosyasında şu şekilde tanımlanmıştır:

Konum denetleyicileri:

p: $k_p \rightarrow 1.0$

i: $k_i \rightarrow 0.0$

d: $k_d \rightarrow 0.5$

Hız kontrolörleri:

p: $k_p \rightarrow 1.0$

i: $k_i \rightarrow 0.0$

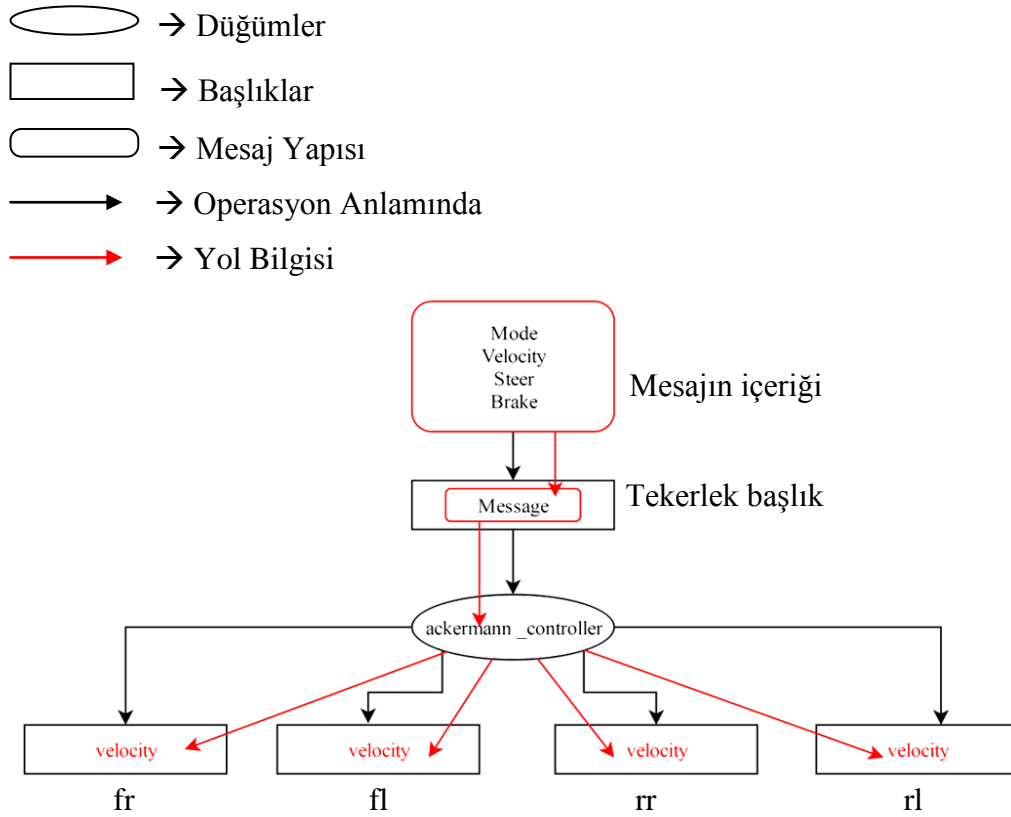
d: $k_d \rightarrow 0.0$

i_clamp: Hız PID döngüsü için I teriminin kısıtlanması için kullanılan değer $\rightarrow 0.0$

4.3.1 Gazebo, ROS'a bağlantı

Bir aracın tekerleklerini robotun fiziksel özelliklerinde Ackerman direksiyon ile kontrol etmek için benzetilmiş aracın orta noktası için kontrolör oluşturmak gereklidir. Bunun için, otonom sürüş kontrolörü ile Gazebo (ackerman kontrol) arasında bir düğüm oluşturulmuştur.

Bu düğümün temel işlevi, hız ve yönlendirme mesajlarını içeren konuyu kaydetmek ve daha sonra bilgiyi her belirli öğeye giden konulara yayınlamaktır. Bkz. Şekil 4.22.



Şekil 4.22. Mesajların dağıtımında kontrolörün gösterilmesi

Bu bölümde, bir ackermann otonom platformunun benzetimi için kullanılan ortak kontrolör parametre değerlerini tanımlayan .yaml dosyaları da eklenmiştir;

4.3.2 Benzetilmiş robot kontrol

Bir simülasyonun amacı, gerçek sistemde aşırı pahalı ve tehlikeli olabilmesi mümkün olduğu için kurgulama ya da denemeye gerek duymadan bir sistemin çalıştığını bilmek, analiz etmek ve değerlendirmektir. Glass-Husain'e göre, her simülatörün üç özelliği olmalıdır: gerçekliği taklit eder (bir benzetimi bir oyundan ayıran şeydir), kendi başına gerçekçi değildir (genellikle gerçek hayatın basitleştirilmesi) ve kullanıcı tarafından değiştirilebilir.

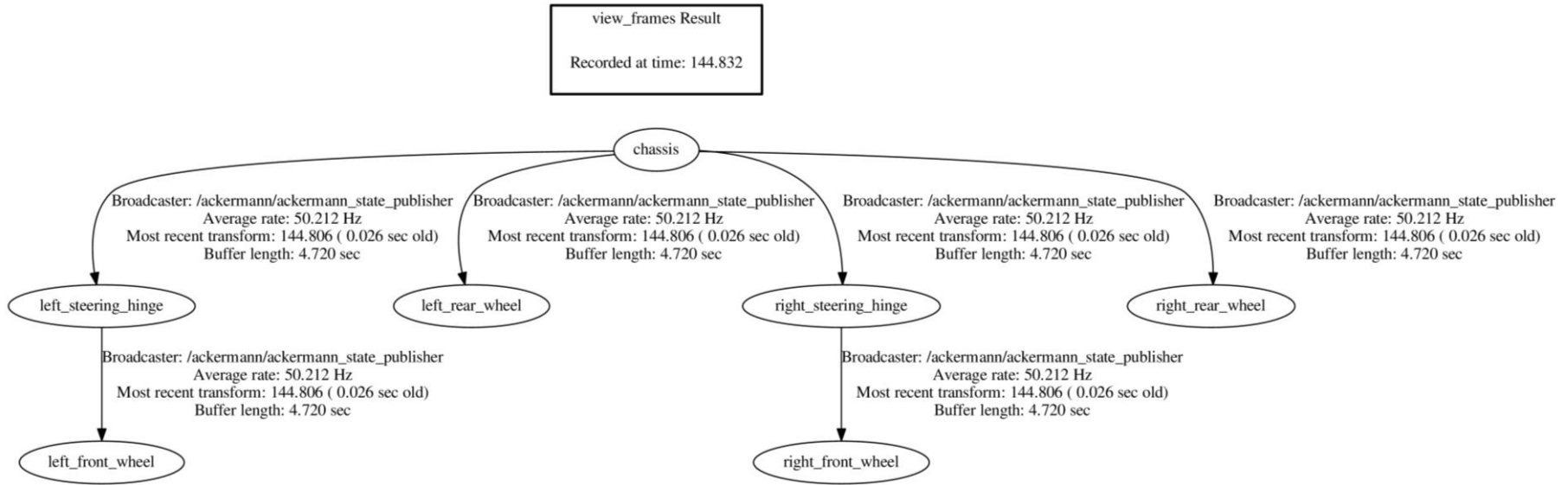
Bir benzetim, gerçek bir sistemin nasıl çalıştığını gösterebilen bir modelin inşasından oluşur. Bununla birlikte, gerçek dünyadaki ilgi sorunları genellikle çok karmaşıktır, böylece basit bir matematiksel model onları temsil edecek şekilde yapılamaz ve bu nedenle sistemin davranışı tahmin edilmelidir. Bir modelde doğru gösterim nadiren mümkündür; Araştırma amaçları için kabul edilebilir derecede sadakatle yapılan yaklaşık değerlerle sınırlı olmalıdır. (Glass-Husain, 1998)

Bir benzetim çalışması aracılığıyla, sistemin iç ve dış değişikliklerinin etkisi incelenebilir, sistem modelinde değişiklikler yapılır ve bu değişikliklerin sistemin davranışındaki etkileri gözlemlenebilir.

Benzetilen sistemin detaylı gözlemlenmesi sistemin daha iyi anlaşılmasına ve dolayısıyla sistemin çalışmasını ve verimliliğini arttıran stratejileri ortaya koymayı mümkün kılabilir.

Gazebo-Ros'un özel durumu için, farklı cihazlar arasındaki bağlantılar, her birine bir port atayan IP adresleri aracılığıyla yapılır; bu şekilde, her robota ayrı ayrı sinyal gönderilebilir. Bu, programların TCP/IP'yi destekleyen herhangi bir dilde yazılmasını sağlar.

Temel bağlantı ilişkisi, kullanıcıya zaman içinde çoklu koordinat karelerini takip etmesini sağlayan bir paket olarak ifade edilir. TF, zaman içinde arabelleğe alınan bir ağaç yapısındaki koordinat kareleri arasındaki ilişkiyi korur ve kullanıcının herhangi bir istenen noktadaki, herhangi bir iki koordinat çerçevesi arasındaki noktaları, vektörleri vb. dönüştürmesini sağlar; rqt tf ağacı, ROS TF çerçeve ağacını görselleştirmek için bir GUI eklentisi sağlar. Bkz. Şekil 4.23.



Şekil 4.23. ROS TF çerçeve ağacı

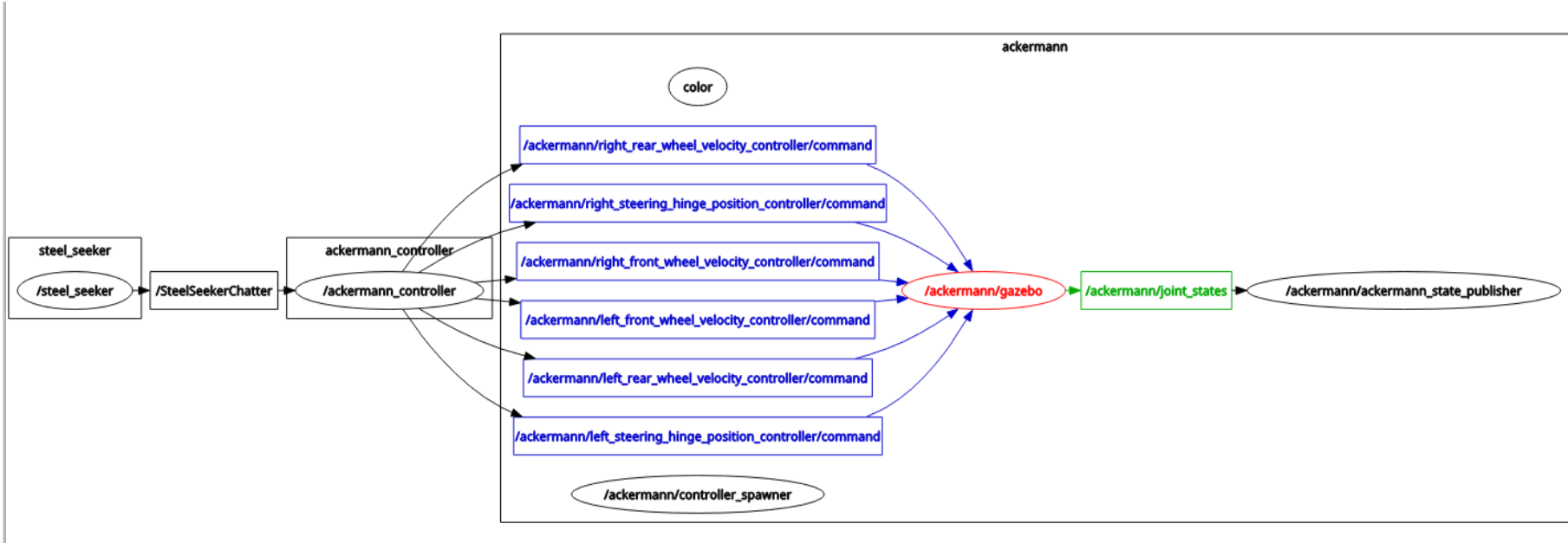
Gerçek model dört tekerleğinin tamamı için tek bir motora sahip olmasına rağmen, simülasyonda bağımsız birimler olarak eklenmelidir.

Spesifik unsurlar URDF modeline eklenmelidir, böylece Gazebo simülasyon ortamında kontrol edilebilir olur. Şanzıman, robot eklemleri ve aktüatörler arasındaki ilişkiyi tanımlamak için kullanılır; bu nedenle bunlar, paralel bağlantılar ve dişli oranı gibi operasyonel ayrıntıları ekleyen açıklayıcı öğelere sahip olabilir. Bu özel durum için basit bir mekanik bağlantı ve basit bir şanzıman kullanılır.

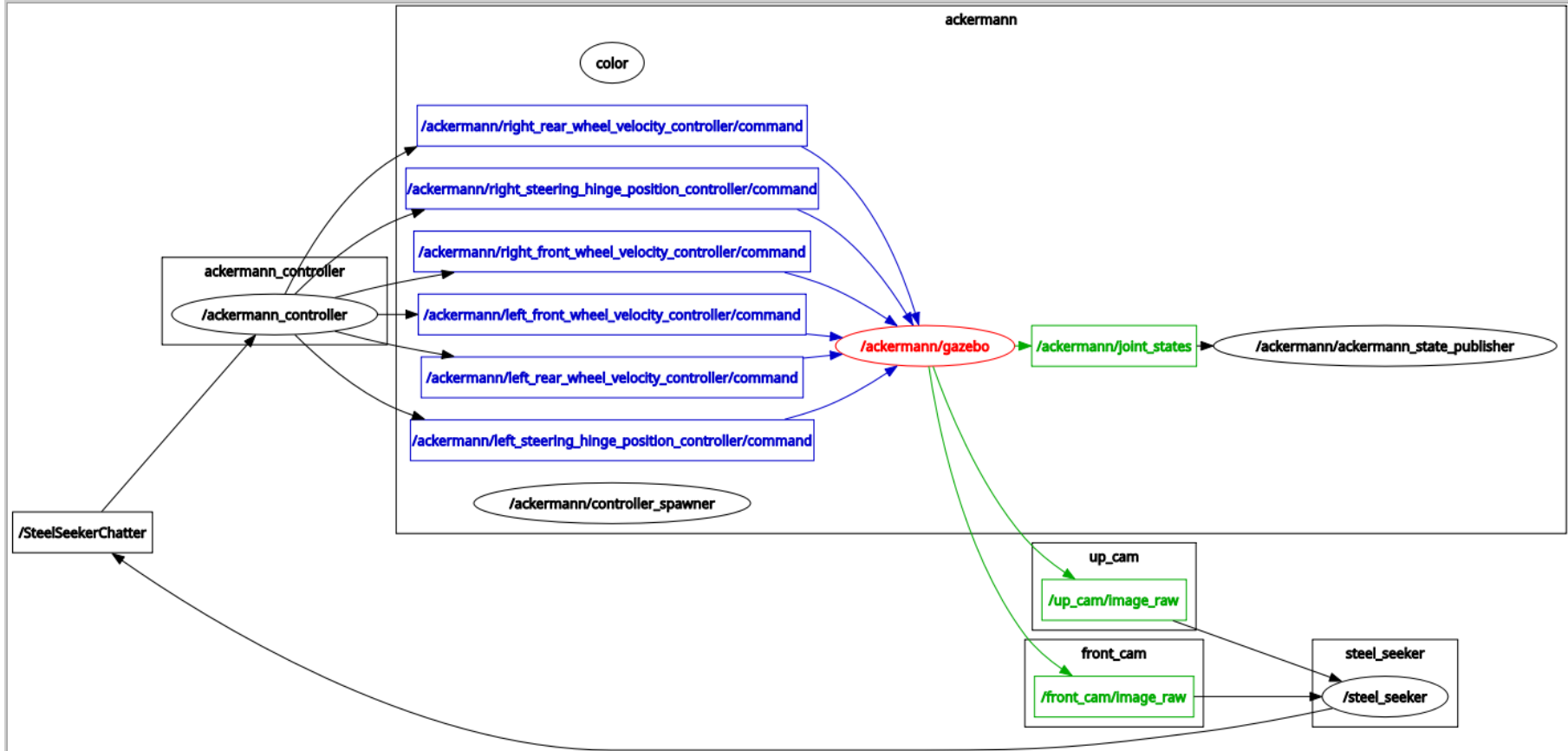
ROS'da, tf, koordinat karelerini takip eden ve birbirleriyle nasıl ilişkili olduklarını gösteren özel bir konudur. Benzetim, dünya koordinat çerçevesinin belirli bir noktasında başlar. Araç hareket ettiğinde, kendi koordinat çerçevesi değişir. Her bir tekerleğin, nasıl döndüğünü ve nerede olduğunu izleyen bir koordinat çerçevesi vardır. Genellikle, robot üzerinde olan dünyada herhangi bir karşılığı olmayan herhangi bir şeyi tf açıklar. Rqt_graph bölümünde, / tf konusunun farklı düğümler tarafından yayınlandığı ve onaylandığı görülebilir.

Düğümleri başlattıktan ve mesajların yayınlanması için gerekli konuları postaladıktan sonra; Grafikler sonuçları Şekil 4.24 ve 4.25'te gösterilmektedir.

Şu an için grafik, ana denetleyiciden Gazebo'ya hız ve yön bilgisi gönderilmesini ve robotun durumunun yayımlanmasını göstermektedir.



Şekil 4.24. Düğümler ve önce yayınlanan başlık döngüsü



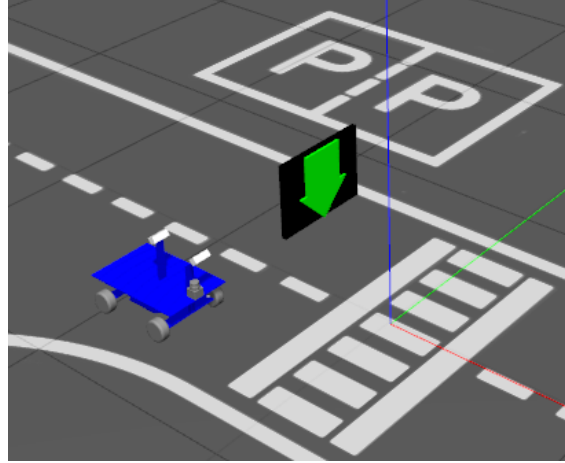
Şekil 4.25. Döngüyü kapatıldıktan sonra yayınlanan düğümler ve başlık

4.3.3 Benzetim ortamı

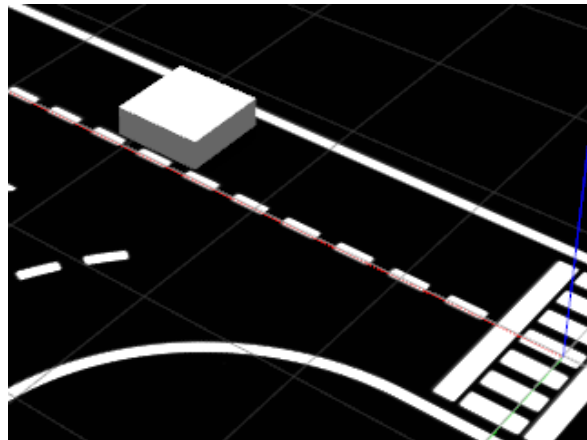
Takip ve 3D görüntüleri içeren detaylar, ikaz bantlı inşaat alanı, tünel ve bir engel gibi simülasyon ortamına eklenir.

3D görüntülerin eklenmesi, simülasyonu daha yavaş hale getirir, çünkü her bir öge birer birer eklenmelidir.

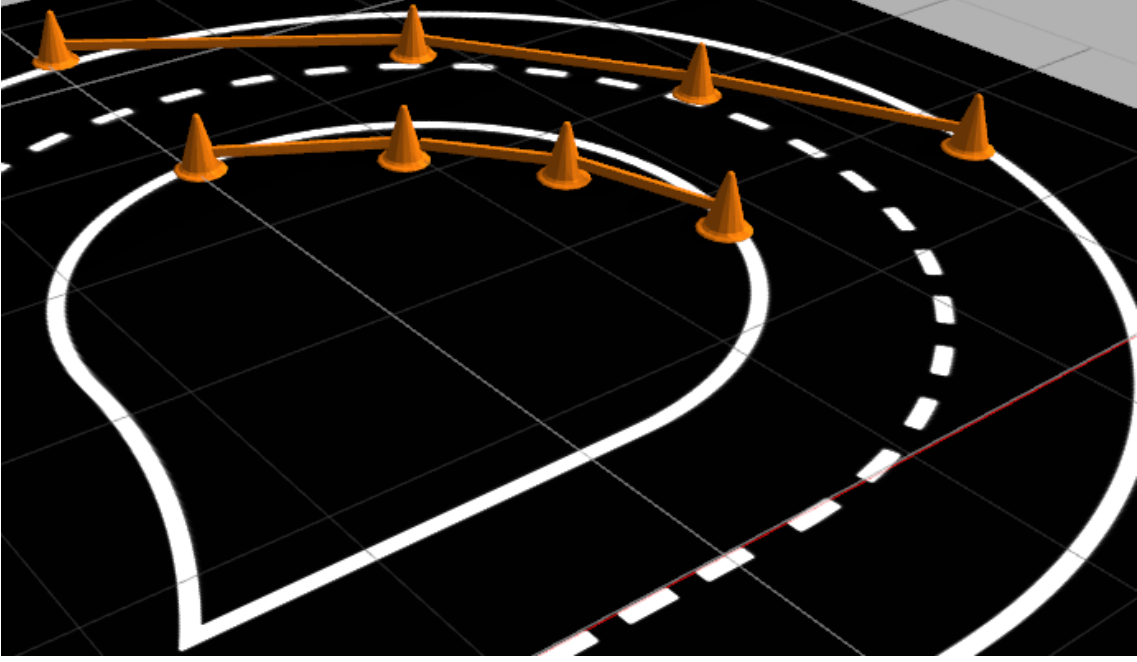
4.26, 4.27, 4.28, 4.29, 4.30 ve 4.31 nolu şekillerde, benzetilmiş ortam modelinin ek nesnelerle birlikte görüntülerinin bazıları verilmektedir:



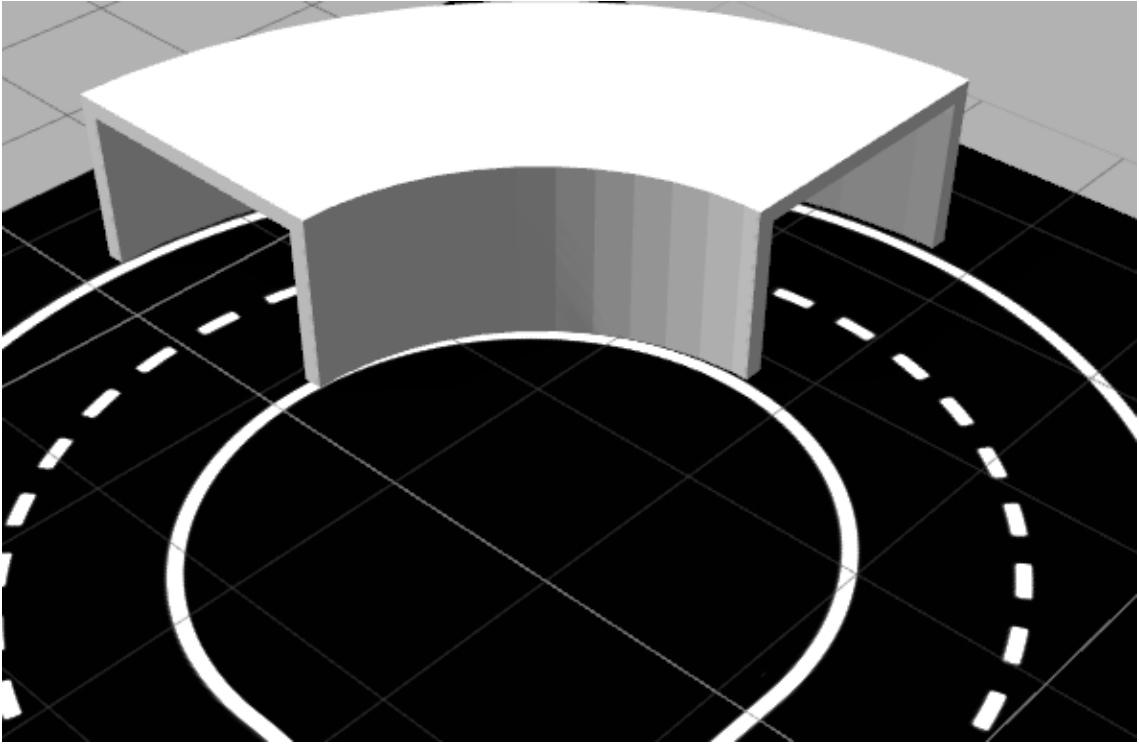
Şekil 4.26. Rota ve trafik ışığı üzerindeki robot



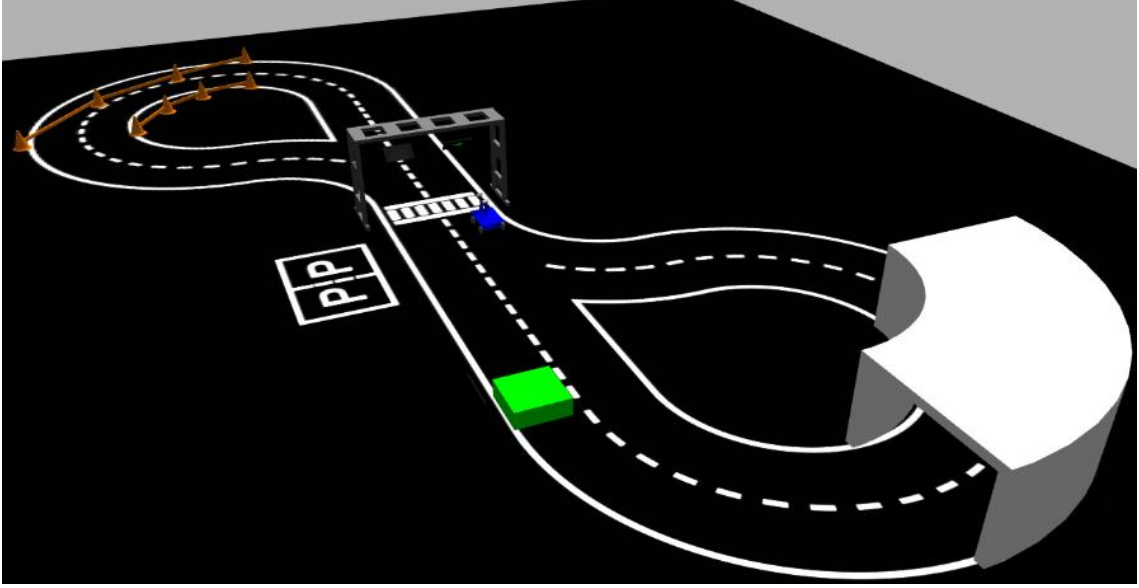
Şekil 4.27. Pistteki engel örneği. Engel, ray üzerinde herhangi bir yere yerleştirilebilir



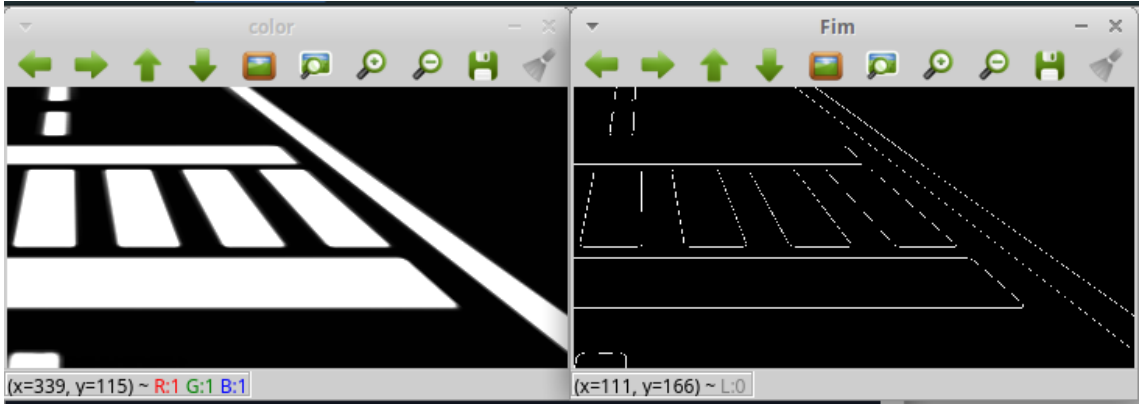
Şekil 4.28. Çalışma alanı. Konikler ve kısıtlama bandı



Şekil 4.29. Pistteki tünel yaklaşık 90 derecelik bir şekle sahiptir



Şekil 4.30. Parçanın robot ve ilave elemanlarla perspektif görünümü

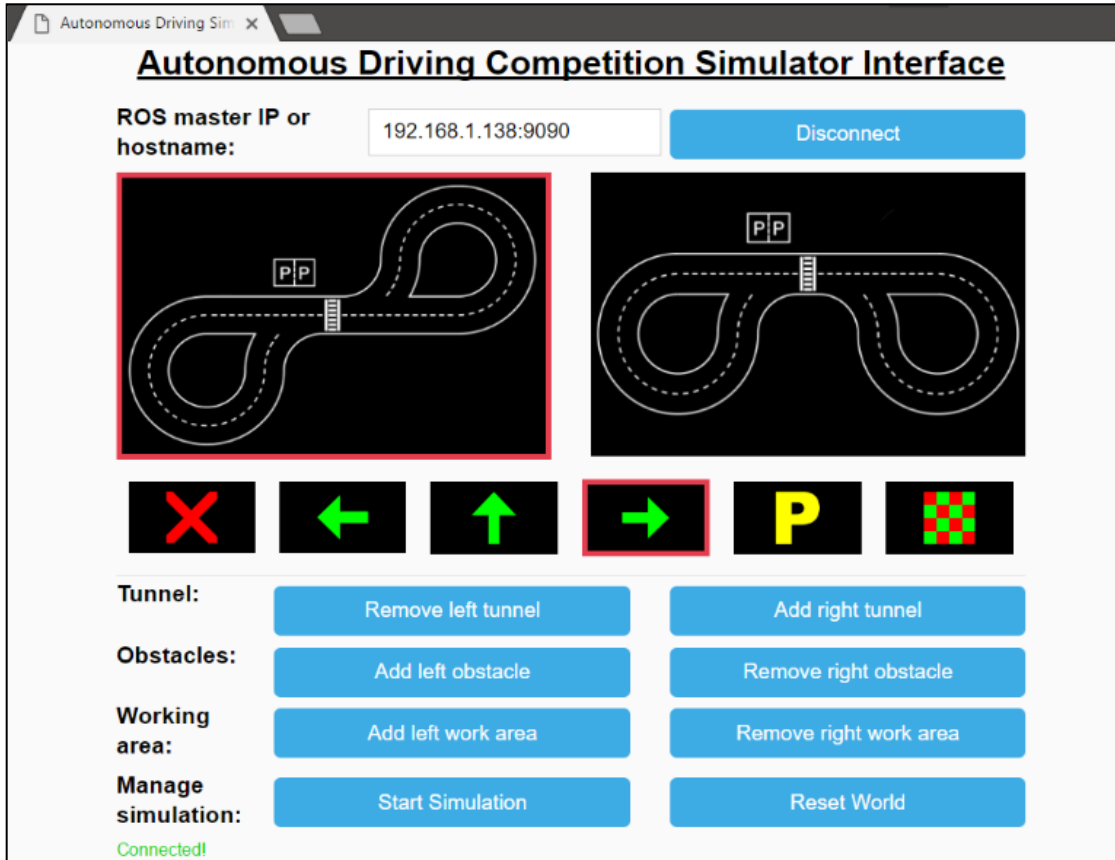


Şekil 4.31. Kontur bulma işleminden sonra, ilgi alanındaki parçanın ve görüntü izinin alınması

4.3.4 Simülasyon Arayüzü

Otonom sürüş yarışında mevcut olan çeşitli zorlukların test edilmesini kolaylaştırmak için bir arayüz geliştirilmiştir. Arayüz, Robot Web Tools projesinden. Roslibjs'e dayalı bir HTML web sayfası olarak uygulanmıştır. (Toris, et al., 2015). Roslibjs, ROS ile arabirim oluşturmada JavaScript tabanlı uygulamaların geliştirilmesini sağlayan bir Node.js kütüphanesinden oluşur. Arabirimi çalıştırmak için modern bir tarayıcı yeterlidir ve roslibjs'i desteklediği sürece farklı işletim sistemleri ve donanım platformlarında çalışabilir - Windows 10 ve Ubuntu 16.04'te Chrome'u kullanarak test

edilmiştir. Şekil 4.32, Şekil 4.30'de gösterilen senaryonun arayüzünün bir fotoğrafını göstermektedir. Şekil 4.32'dan görülebileceği gibi, arabirim parça seçimine, sinyalleme panellerinin güncellenmesine ve tünellerin, engellerin ve çalışma alanlarının bağımsız olarak eklenmesine / kaldırılmasına izin vermektedir. Ayrıca, benzetimin duraklatılmasını / çalıştırılmasını ve dünyanın sıfırlanmasına izin verir. ROS bağlantısı, Web Sunucusu kullanılarak rosbridge sunucusuna yapılır; bu, arayüzün farklı bir PC veya mobil cihaz üzerinde uzaktan çalıştırabileceği anlamına gelir. Bu arayüzle, bir operatör, bir mouse tıklamasıyla çeşitli zorluklara nesnel ekleyerek / kaldırarak rekabetin gerçek koşullarını simüle edebilir. Ayrıca, robot zorlukları yerine getirirken, sinyalleme panelleri gerçek rekabetteki gibi güncellenebilir ve daha gerçekçi bir teste izin verilebilir.



Şekil 4.32. Şekil 4.30 teki durum için arayüz görüntüsü

5. SONUÇ VE ÖNERİLER

Bu bölümde, önerilen amaçları göz önünde bulundurarak, bu çalışma boyunca yürütülen görevlerin ardından sonuçlar sunulmuştur. Ve daha sonra bu projenin devam etmesi için gelecekteki çalışmalar için öneriler sunulmuştur.

5.1. Sonuçlar

Robot, ROS ve Gazebo araçlarını kullanarak benzetilmiş bir çevrede inşa edilmiştir, böylece projenin özerk denetleyicisi tarafından oluşturulan yol kontrolörü değiştirmeden hem gerçek hem de benzetilmiş araçta kullanılabilir.

Aynı denetleyiciyi benzetilmiş ve gerçek platform için kullanmak, ortamın robotun davranışını nasıl etkilediğini daha açık bir şekilde tanımlamasına aynı zamanda benzetilmiş bir modeldeki ve kontrollerde ki muhtemel hataları tanımlamak için gerekli temelleri sağladığı için büyük önem taşır. Denetleyicinin tepkisi, gerçek ve simüle edilmiş ortamda benzer olmasına rağmen, performansı etkileyebilecek birçok özellik bulunduğu için gerçek ortamın tüm özellikleri bir benzetimde toplanamaz.

Gerçek dünya çok karmaşıktır, bu yüzden basit algoritmalarla ifade edilemez, ancak eğer çevrenin ana yönlerini dikkate alarak gerçek dünyanın soyutlanması yapılabiliyorsa doğrudur; Bununla birlikte, gerçek dünyanın her yönü benzetilemez, bu nedenle beklenen ve mevcut sonuçlar arasında bir miktar hata payı kabul etmek gerekir, ancak bunun sonucu olarak doğrudan veya dolaylı olarak etkilenen diğer faktörler kullanılarak da çözülebilir. Ancak hem benzetilmiş hem de gerçek ortamda, robotun etkileşim biçimini çok iyi bilmek gerekir.

Gözlemlenebilen temel bir husus, çevrenin ışığının, bilgi toplamanın önemli bir şekilde etkilenmesi ve bu nedenle gece Otonom Sürüş gibi çok parlaklığın olmadığı ortamlarda alınan görüntülerin işlenmesini etkilemektedir. Renklerin ve parlaklığın

farkında olan farklı aralıkların dikkate alınması gerektiği için, görüntülerin toplanmasının ve işlenmesinin farklı bir işlemle yapılması gerektiğini dikkate almak gerekir.

5.2. Öneriler

Gelecekte, benzetimden gerçek zamanlı olarak güncellenen parçaların en üst görünümüne sahip olmak ve böylece benzetilen yarışmayı daha sezgisel bir şekilde uzaktan test edip izleyebilmek ilginç olacaktır. Geliştirilen araçlar aynı zamanda benzetime dayalı yarışma için de kullanılabilir.

Şu anda en yakın sağdaki çizgiyi referans noktası olarak aldığı için denetleyicinin, bir engelden kaçındıktan sonra parçanın daha iyi izlenmesine olanak tanınması için geliştirilmesi gerekir. Sonrasında robotun başlangıçta takip ettiği çizgiyi bir referans noktası olarak alması gereklidir.

KAYNAKLAR DİZİNİ

2017 Georgia Tech AutoRally. (2016), AutoRally Robot, <https://autorally.github.io/>, erişim tarihi: 26.09.2016.

Ackerman, E., 2016, IEEE Spectrum, <http://spectrum.ieee.org/cars-that-think/transportation/self-driving/autonomous-mini-rally-car-teaches-itself-to-powerslide> adresinden alındı, erişim tarihi: 16.12.2016.

Audi Corporation, 2017, Audi Autonomous Driving Cup, Audi Autonomous Driving Cup 2017, <https://www.audi-autonomous-driving-cup.com/wettbewerb/bewerbung/>, erişim tarihi: 25.02.2017.

AVC.Sparkfun, 2016, Avc.sparkfun.com, <https://avc.sparkfun.com/2016/rules#AVC>, erişim tarihi: 01.02.2017.

Costa, V., Rossetti, R. J., & Sousa, A., 2016, Autonomous Driving Simulator for Educational Purposes, Information Systems and Technologies. IEEE Xplore, doi: 10.1109/CISTI.2016.7521461.

DARPA Robotics Challenge, 2015, DARPA Robotics., <http://archive.darpa.mil/roboticschallenge/index.htm>, erişim tarihi: 06.02.2017.

Defense Advanced Research Projects Agency DARPA, DARPA News and Events, <http://www.darpa.mil/news-events/2013-06-27>, erişim tarihi: 07.02.2017

Dudek, G., & Jenkin, M., 2011, Computational Principles of Mobile Robotics, Cilt 1, Cambridge University Press.

Gazebo Organization, 2014, Gazebo.org, http://gazebo.org/tutorials/?tut=ros_urdf#, erişim tarihi: 11.10.2016

KAYNAKLAR DİZİNİ (devam)

GazeboSim Organization, 2014, GazeboSim.org, http://gazeboSim.org/tutorials/?tut=ros_gzplugins#Tutorial:UsingGazeboPl, erişim tarihi: 11.10.2016

Georgia Institute of Technology, 2017, AutoRally platform documentation, https://github.com/AutoRally/autorally_platform_instructions/blob/master/autorally_chassis_instructions.pdf, erişim tarihi: 27.02.2017

Glass-Husain, W, 1998, Forio.com, Foro Business Simulations, Gilles Brougère: <http://forio.com/about/blog/>, erişim tarihi: 20.10.2016

Into Robotics 2014, Intorobotics. A blog about robots, technologies and work loop, <https://www.intorobotics.com/ros-tutorials-start-working-arduino-raspberry-pi/>, erişim tarihi: 28.09.2016

KPMG LLP ve Center for Automotive Research (CAR), 2012, KPMG, https://assets.kpmg.com/content/dam/kpmg/pdf/2015/10/self-driving-cars-next-revolution_new.pdf, erişim tarihi: 10.10.2016

National Council on Disability, 2015, United States Government Publishing Office, https://permanent.access.gpo.gov/gpo66235/NCD_AutomatedVehiclesReport_508-PDF.pdf, erişim tarihi: 07.10.2016

Neto, B. B, 2014, Condução Autónoma – Desenvolvimento de um robô para navegação num ambiente rodoviário à escala. Yüksek Lisans Tezi, Polytechnic Institute of Leiria, Leiria.

ROS Organization, ROS.org, <http://www.ros.org/about-ros/>, erişim tarihi: 26.09.2016

KAYNAKLAR DİZİNİ (devam)

- Ross, P. E., 2017, IEEE Spectrum, <http://spectrum.ieee.org/cars-that-think/transportation/self-driving/nvidia-ceo-announces>, erişim tarihi: 12.02.2017
- Sousa, A., & Pereira, A., 2017, Festival Nacional de Robotica - Coimbra 2017, <http://robotica2017.isr.uc.pt/index.php/en/competitions/major/autonomous-driving>, erişim tarihi: 01.02.2017
- Suzuki, S., & Abe, K., 1985, Topological structural analysis of digitized binary images by border following, Science Direct, s. 32-46.
- Toris, R., Kammerl, J., Lu, D. V., Lee, J., Osentoski, S., Wills, M., Jenkins, O. C., 2015, Robot Web Tools: Efficient messaging for cloud robotics, IEEE Xplore, doi: 10.1109/IROS.2015.7354021.
- Walter Hillier, V. A., & Thornes, N., 2012, Hillier's fundamentals of motor vehicle technology, Cilt 6, United Kingdom: Nelson Thornes Ltd.
- Williams, G., Aldrich, A., & Theodorou, E. A., 2017, "Model Predictive Path Integral Control: From Theory to Parallel Computation", Journal of Guidance, Control, and Dynamics, Cilt 40, Atlanta, Georgia: American Institute of Aeronautics and Astronautics, Inc.
- World Health Organization, 2015, WHO World Health Organization, <http://www.who.int/mediacentre/factsheets/fs358/en/>, erişim tarihi: 01.11.2016
- World Health Organization, 2016, WHO World Health Organization, Map production: Information Evidence and Research (IER), http://gamapserv.who.int/mapLibrary/Files/Maps/Global_RoadTraffic_Mortality_2013.pngWo, erişim tarihi: 01.11.2016

EK AÇIKLAMALAR

Ek Açıklamalar – A: Benzetilmiş robot kontrolü

Steer_seeker projesini adım adım açıklayan belge.

A.1. . Kurulum komutları

1. KURULUM ÖNŞARTLARI

1.1. Ubuntu 16.04 64-bit kur. İşletim sistemini kurduktan sonra yeni güncellemeleri kontrol et. Aşağıdaki programları kur.

1.2. Ros-kinetic-desktop-full.

1.3. Gazebo7.

1.4. Gazebo_ros_pkgs

ROS paketleri ROS ve Gazebo arasındaki arayüzlerdir. Gazeboyu aşağıdaki komutu kullanarak ROS ile kurmak mümkündür:

```
$ sudo apt-get install ros-kinetic-gazebo-ros-pkgs ros-kinetic-gazebo-ros-control
```

https://github.com/ros-simulation/gazebo_ros_pkgs url'yi kullanarak ta kurulum mümkündür. git önceden kurulmadıysa. **\$ sudo apt-get install git** komutunu kullanın

Aşağıdaki komutları kullanarak herşeyin çalıştığını kontrol edebilirsiniz (farklı terminallerde):

```
$ roscore
```

```
$ rosrun gazebo_ros gazebo
```

Başlıkların çalıştığını aşağıdaki örnek komutlarla kontrol edebilirsiniz.:

\$ rostopic list

/clock

/gazebo/link_states

/gazebo/model_states

/gazebo/parameter_descriptions

/gazebo/parameter_updates

/gazebo/set_link_state

/gazebo/set_model_state

/rosout

/rosout_agg

2. YENİ ÇALIŞMA ORTAMI KURULUMU

2.1. ROS Kinetik ortamı kaynaktır:

\$ source /opt/ros/kinetic/setup.bash

2.2. Çalışma ortamını ve src dosyasını tutacak dosyayı oluştur

\$ mkdir -p ~/catkin_ws/src

2.3. Kaynağa git ve çalışma ortamını başlat:

\$ cd ~/catkin_ws/src

\$ catkin_init_workspace

2.4. İlk çalışma alanına inşa edilebilir, sadece bu durumda boş olacak dosyaları üretmek için:


```
$ cd..
```

```
$ catkin_make
```

3. KLON VEYA FİLTELERİN KORUNMASI

3.1. Steel_seeker depolarını bir catkin çalışma alanından alın. Önerilen yer, önceden oluşturulmuş alan olan `~/catkin_ws/src/` dir.

```
$ git clone https://github.com/ipleiria-robotics/steel_seeker.git
```

```
$ git clone https://github.com/ipleiria-robotics/adc_gazebo.git
```

4. DERLEME VE ÇALIŞMA

4.1. Derlemek ve kurmak `catkin_make` catkin çalışma alanı klasöründen. `Setup.bash` kaynağını hatırla:

```
$ cd ..
```

```
$ catkin_make
```

```
$ source /home/USER/catkin_ws/devel/setup.bash
```

4.2. Yükselmiş dosyayı çalıştırın, engeller ve çalışma alanı olan ortamı başlatabilir: `Adc`'yi (Autonomous driving competition) başlatmak isterseniz yapın:

```
$ roslaunch adc_gazebo adc.launch
```

Tünel ve çalışma alanını zaten çalışan simülasyona eklemek istiyorsanız:

```
$ roslaunch adc_gazebo spawn_models.launch
```

Eğer robotu halihazırda çalışan simülasyona eklemek istiyorsanız:

```
$ roslaunch steel_seeker_gazebo spawn_steel_seeker.launch
```

Simülasyonu robot ile açmak isterseniz:

```
$ roslaunch steel_seeker_gazebo adc_steel_seeker.launch
```

Robotun çalışmasını istiyorsanız (robot modeli olan simülatörü zaten bulundurun):

```
$ roslaunch steel_seeker_controller steel_seeker.launch
```

Ek Açıklamalar – B: Basic Characteristics of ROS and Gazebo

The content of this appendix is focused on review a robot control framework, for this purpose are used the Gazebo and ROS interfaces. The workspace was prepared to simulate a robot, controlling behavior and collect information from it. Aims to learn how to use the basic concepts of interfaces work, as well as the applicability of control and visualization tools also understand the connection between ROS and Gazebo.

B.1. Gazebo, connect to ROS

To understand and acquainted with the ROS and Gazebo platforms, the simulation of a car-like robot (Ackerman) was performed, in which the vehicle control was implemented and connected to the image viewing platform Rviz, hence the following procedure was done:

- Setup a ROS workspace
- Create the projects
- Create a world in Gazebo
- Create robot model
- Connect robot model to ROS
- Control the robot by keyboard
- Add a camera
- Implement Rviz to visualize the robot information

Important processes

Gazebo simulator can be run as stand-alone, but it is used with ROS to consult the information from the created nodes in the sensors and to send different specification commands. To run the simulator first has to boot into a terminal ROS:

roscore: Is a specialization of the roslaunch tool for bringing up the "core" ROS system. For more information, please see the roscore documentation.

roslaunch gazebo_ros gazebo

roslaunch package_name file.launch: Is a tool used to run several ROS nodes of a package automatically and with a specific configuration imposed by an XML file with extension .launch.

URDF (Universal Robotic Description Format): It consists of a tree of geometric elements (links) connected by joints that determine the relationship between them; these joints can be fixed or mobile. XML file format.

Xacro (XML Macros): Xacro is an XML macro language. With xacro, you can construct shorter and more readable XML files by using macros that expand to larger XML expressions. This package is most useful when working with large XML documents such as robot descriptions. It is heavily used in packages such as the urdf.

B.2. Robot description

In order to learn how Gazebo and ROS tools are implemented, a simulation of an Ackerman was done, in this case a chassis with four wheels and a camera.

In Figure 1 the characteristics of the implemented car is shown and the result of simulation using Gazebo Wireframe View, this in order to understand its design.

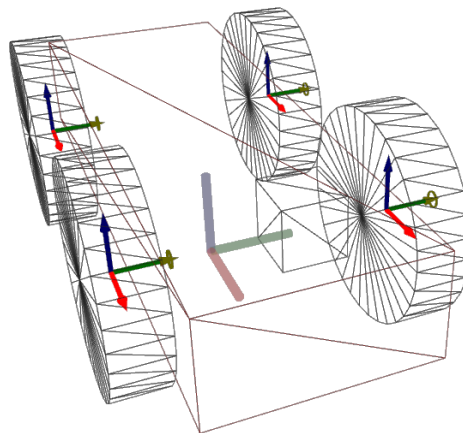


Figure 1 – Car model in Wireframe View

For implementation was used a URDF file generated by Xacro and put it the description package.

The Universal Robotic Description Format (URDF) is an XML file format used in ROS to describe all elements of a robot. To use a URDF file in Gazebo, some additional simulation-specific tags must be added to work properly with Gazebo.²

For the implementation of the robot, the some of the steps followed according to the available documentation:

Required

An `<inertia>` element within each `<link>` element must be properly specified and configured.

Optional

- Add a `<gazebo>` element for every `<link>`
- Convert visual colors to Gazebo format
- Add sensor plugins
- Add a `<gazebo>` element for every `<joint>`

² [2] GAZEBO Robot simulator “ROS - URDF” “http://gazebosim.org/tutorials?tut=ros_urdf”, [Online]

- Add actuator control Plug-in
- Add a <gazebo> element for the <robot> element
- Add a <link name="world"/> link if the robot should be rigidly attached to the world/base_link

Gazebo implementation allows adding some features to display the item and make the simulation closer to the reality. It can be added features of color, shadows and movement in real time, as is illustrated in Figure 2.

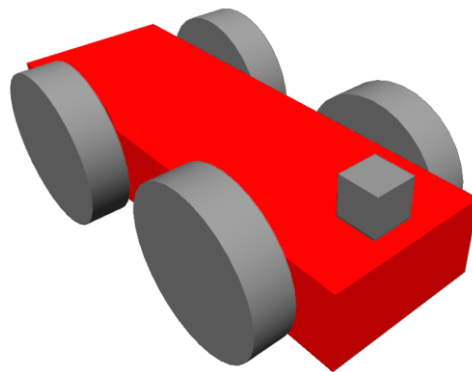


Figure 2 – Car model with colors and shadows

Additionally, it joined in the world the ground plane of the track which will be implemented in the Autonomous Driving competition at the Portuguese Robotics Open. It was made using the image of the treads in .png format and scaling the measures according with the real track. Figure 3

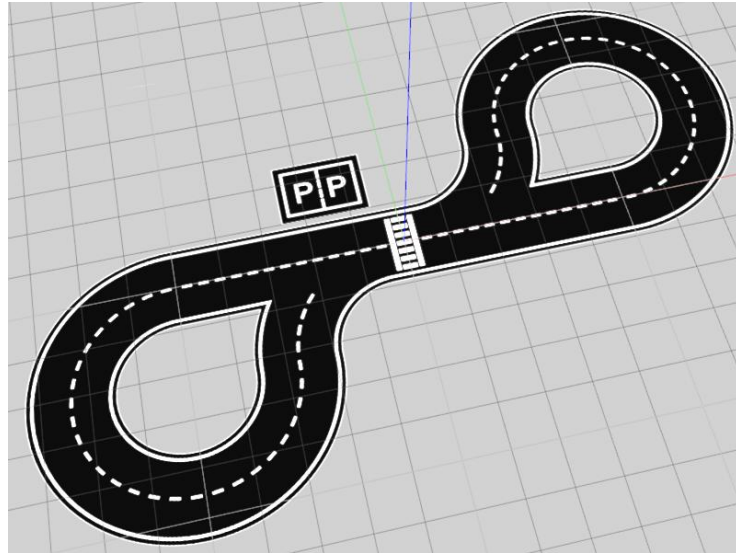


Figure 3 – Ground plane: Track

B.3. Robot control - Gazebo

Taking advantage Gazebo already has some Plugins to drive a robot, was used the plugin Differential Drive.

Differential_drive_controller can change the robot's direction by varying the relative rate of rotation of its wheels and does not require an additional steering motion, but the movement is based on two wheels placed on either side of the robot body. Because in this case the robot has four wheels, it is necessary to allow the rear wheels to roll freely, so that the carriage control is done by the front wheels.

This plugin will be subscribed by the `cmd_vel` topic specified with the «commandTopic» tag and convert the messages to the proper commands on the wheels. It also provides some odometry data. And to teleoperate the robot with the keyboard it is used the node `turtlebot`, remapping the topic name to connect it with the robot.

Some of the considerations for the control should be the configuration of the PID (Proportional-Integral-Derivative) controller in a `yaml` file that gets loaded to the parameter server via the `roslaunch` file for starting the `ros_control` controllers.

B.4 RViz

RViz is a robot visualization tool. It provides a convenient GUI to visualize sensor data, robot models, environment maps, which is useful for developing and debugging your robot controllers. This is configured in the main Launch file.

In RViz you can view the simulation with different details, and visualize the added sensors to the robot as un the Figure 4 respect to the current simulation Figure 5.

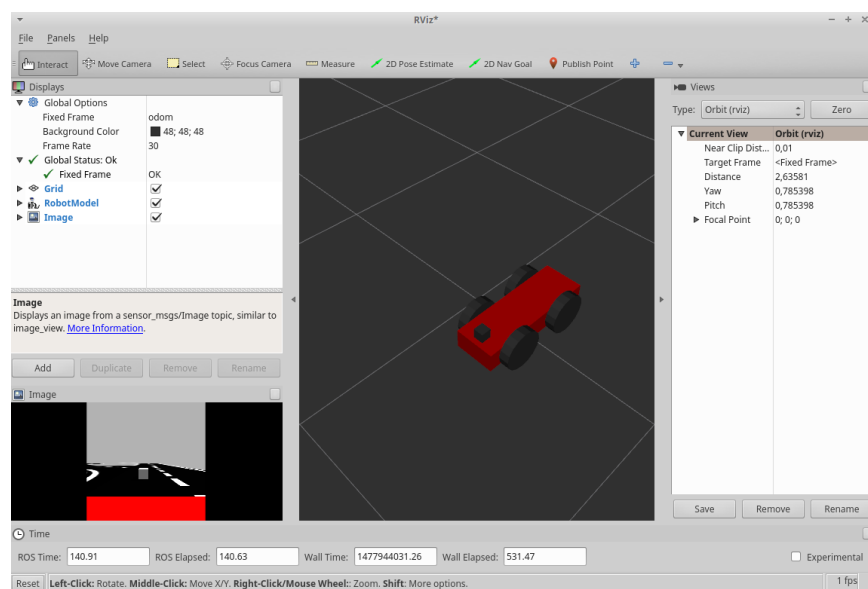


Figure 4 – RViz view

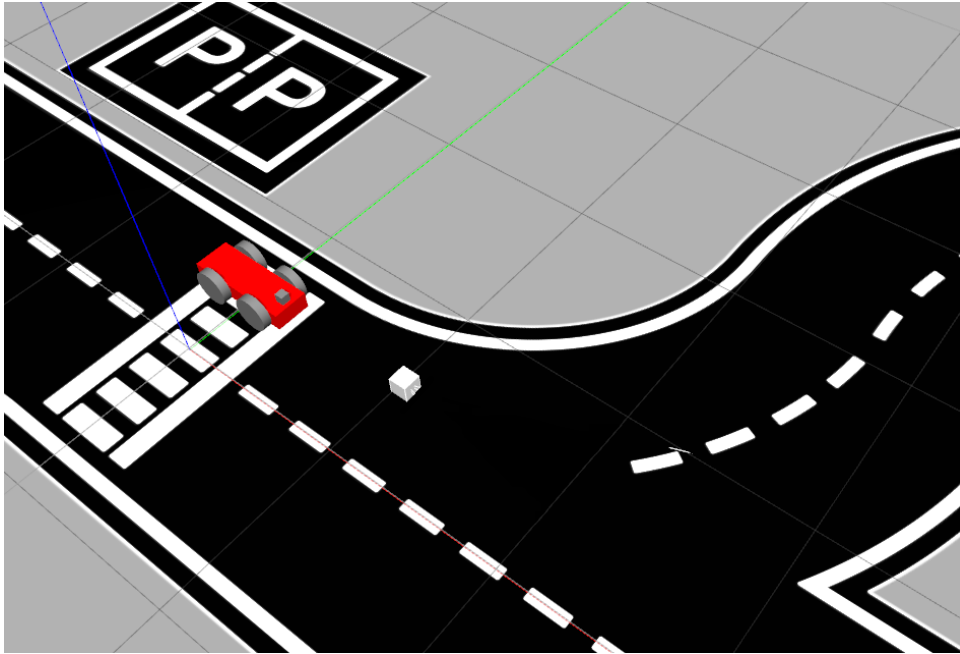


Figure 5 – Gazebo - current simulation