

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**OTONOM ARAÇLAR İÇİN 2B LAZER TARAYICI KULLANILARAK YENİ
3B LIDAR SİSTEMİ ELDE EDİLMESİ VE ENGEL TESPİTİ**

YÜKSEK LİSANS TEZİ

Ahmet KAĞIZMAN

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Erdinç ALTUĞ

TEMMUZ 2018

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**OTONOM ARAÇLAR İÇİN 2B LAZER TARAYICI KULLANILARAK YENİ
3B LIDAR SİSTEMİ ELDE EDİLMESİ VE ENGEL TESPİTİ**

YÜKSEK LİSANS TEZİ

**Ahmet KAĞIZMAN
(518141003)**

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Erdinç ALTUĞ

TEMMUZ 2018

İTÜ, Fen Bilimleri Enstitüsü'nün 518141003 numaralı Yüksek Lisans Öğrencisi Ahmet KAĞIZMAN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "OTONOM ARAÇLAR İÇİN 2B LAZER TARAYICI KULLANILARAK 3D LIDAR SİSTEMİ ELDE EDİLMESİ VE ENGEL TESPİTİ" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Erdiñç ALTUĞ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Şeniz ERTUĞRUL**
İstanbul Teknik Üniversitesi

Prof. Dr. Haluk KÜÇÜK
Marmara Üniversitesi

Teslim Tarihi : **20 Temmuz 2018**
Savunma Tarihi : **23 Temmuz 2018**





Eşime,



ÖNSÖZ

Otonom araç teknolojisi üzerinde yapılan çalışmalar son yıllarda artış göstermektedir ve buna bağlı olarak engel tespiti, engelden kaçınma, konum tespiti, haritalama ve yol planlaması gibi birçok farklı alanda çalışmalar yürütülmektedir. Bu tez çalışmasında 2 boyutlu lazer tarayıcı (LIDAR) kullanılarak kapalı bir ortamda 360⁰ görüş alanına sahip 3 boyutlu nokta bulutu verisi elde edilebilen yeni bir LIDAR sistemi geliştirilmiştir. Yapılmaya çalışılan bu sistem, otonom araçlarda engel tespiti için ekonomik ve etkili bir çözüm sunmaktadır.

Bu tez çalışmasının her aşamasında benden ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübeleriyle bana yol gösteren danışman hocam Sayın Doç. Dr. Erdinç ALTUĞ'a ve kullanılan malzemelerin temininde maddi destek sağlayan İTÜ BAP birimine sonsuz teşekkürlerimi sunarım. Ayrıca, bu zorlu tez sürecinde gösterdiği fedakârlık ve sabrıyla en büyük destekçim olan kıymetli eşim Tuğba KAĞIZMAN'a ve bu günlere gelmemde büyük pay sahibi olan sevgili anne ve babama teşekkürü bir borç bilirim.

Temmuz 2018

Ahmet KAĞIZMAN
(Mekatronik Mühendisi)



İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
SEMBOLLER	xiii
ÇİZELGE LİSTESİ	xv
ŞEKİL LİSTESİ	xvii
ÖZET	xix
SUMMARY	xxiii
1. GİRİŞ	1
1.1 Otonom Araçların Gelişim Süreçleri.....	2
1.2 Otonom Araçların Algılayıcı Sistemleri.....	6
1.2.1 GPS	7
1.2.2 IMU	7
1.2.3 Kamera	8
1.2.4 Tekerlek enkoderi	8
1.2.5 Ultrasonik ve kızılötesi sensörler	9
1.2.6 Radar	9
1.2.7 LIDAR	10
1.3 Tezin Amacı	11
1.4 Literatür Taraması	11
2. 3B LIDAR SİSTEMİ TASARIMI	15
2.1 Mekanik Tasarım.....	17
2.1.1 Enkoderli dc motor.....	18
2.1.2 Slip-ring	19
2.1.3 Mekanik LIDAR platformu	19
2.2 Elektronik Tasarım	19
2.2.1 Hokuyo LIDAR.....	20
2.2.2 Arduino Uno	23
2.2.3 L298N motor sürücü	23
2.1.4 Jetson TX1	24
2.1.5 Pil	25
3. MOBİL ARAÇ TASARIMI	27
3.1 BNO055 IMU	29
3.2 Arduino Mega	30
3.3 BTS7960 Motor Sürücü	30
3.4 LM2596 Voltaj Düşürücü Regülatör Modülü	31
3.5 Pil	31
4. YAZILIM	33
4.1 Nokta Bulutu ve Geometrik Dönüşümler.....	33
4.1.1 Nokta bulutu.....	33

4.1.2 Geometrik dönüşümler	34
4.2 ROS	35
4.2.1 Dosya sistemi	37
4.2.1.1 Paketler (packages) ve yığınlar (stacks)	37
4.2.1.2 Bildiriler (manifests)	37
4.2.1.3 Mesajlar (messages) ve servisler (services)	38
4.2.2 Hesaplama grafiği	38
4.2.2.1 Ana sunucu (master) ve değişken sunucusu (parameter server)	38
4.2.2.2 Düğüm (node)	39
4.2.2.3 Başlık (topic)	39
4.2.2.4 Mesaj (message) ve servis (service)	39
4.2.2.5 Çanta (bag)	39
4.2.3 URDF (evrensel robot tanım dosyası) paketi	40
4.2.4 Dönüşüm çerçevesi (transform frame) paketi	40
4.3 3 Boyutlu Nokta Bulutu Görüntüsünün Elde Edilme Aşamaları	41
4.4 3B Haritalama	45
4.3.1 Octree	46
4.5 Navigasyon	48
4.5.1 Navigasyon yığını (navigation stack)	48
4.5.1.1 Sensör dönüşümleri (sensor transforms)	49
4.5.1.2 Sensör bilgileri (sensor sources)	49
4.5.1.3 Odometri bilgileri (odometry sources)	49
4.5.1.4 Taban kontrolörü (base controller)	50
4.5.1.5 Move_base paketi	50
4.5.2 robot_localization paketi	51
4.6 Deneyler	52
4.6.1 Bilgisayar deneyleri	53
4.6.2 Jetson tx1 deneyleri	56
5. SONUÇ VE ÖNERİLER	63
5.1 Sonuçlar	63
5.2 Öneriler	65
5.3 Gelecekteki Çalışmalar	66
KAYNAKLAR	67
ÖZGEÇMİŞ	71

KISALTMALAR

ROS	: Robot İşletim Sistemi
GPS	: Küresel Konumlandırma Sistemi
IMU	: Ataletsel Ölçüm Ünitesi
RADAR	: Radyo Sinyali Tespit Etme ve Mesafe Tayini
LIDAR	: Işık Tespiti ve Ölçümü
RVIZ	: ROS Görselleştirici
UGV	: İnsansız Kara Aracı
İKA	: İnsansız Kara Aracı
INS	: Ataletsel Navigasyon Sistemi
USB	: Evrensel Seri Veriyolu
PWM	: Darbe Genişlik Modülasyonu
URDF	: Evrensel Robot Tanım Dosyası
TF	: Dönüşüm Çerçevesi
GUI	: Grafiksel Kullanıcı Arayüzü
DWA	: Dinamik Pencere Yaklaşımı
AMCL	: Adaptif Monte Carlo Yerelleştirmesi



SEMBOLLER

- X,Y,Z** : Kartezyen koordinatlar
 ρ : Lazer tarayıcı ile ölçülen nesne arasındaki eğik mesafe
 α : Ölçüm doğrusunun x eksenine ile yatay düzlemde yaptığı açı
 θ : Ölçüm doğrusunun yatay düzlemle yaptığı eğim açısı





ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : 3B LIDAR sistemi bileşenleri ve adetleri.	16
Çizelge 2.2 : Kullanılan enkoderli dc motorun özellikleri.	18
Çizelge 2.3 : Lazer-doğrusal hassasiyet tablosu.	22
Çizelge 3.1 : DAGU 6WD araç platformu özellikleri.	27
Çizelge 3.2 : Motor özellikleri.	29





ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : IRobot firması tarafından üretilen uzaktan kontrollü İKA olan PackBot.	1
Şekil 1.2 : GM firması tarafından deneysel amaçlı üretilen otonom araç	2
Şekil 1.3 : Stanford cart.	3
Şekil 1.4 : Dünyanın ilk otonom aracı olarak kabul edilen VaMoRs Mercedes	4
Şekil 1.5 : Navlab5.	4
Şekil 1.6 : DARPA tarafından düzenlenen yarışmayı kazanan Stanley.	5
Şekil 1.7 : Google tarafından üretilen Koala car.	5
Şekil 1.8 : Otonom araç donanımı.	6
Şekil 1.9 : GPS.	7
Şekil 1.10 : IMU	8
Şekil 1.11 : 3B Velodyne LIDAR.	10
Şekil 2.1 : 3B LIDAR platformu.	16
Şekil 2.2 : Portatif 3B LIDAR platformu.	17
Şekil 2.3 : 3B LIDAR prototipi.	17
Şekil 2.4 : Mekanik LIDAR platformu.....	20
Şekil 2.5 : HOKUYO-URG-04LX-UG01.	21
Şekil 2.6 : Hokuyo LIDAR doğrusal ölçüm aralığı.	22
Şekil 2.7 : Hokuyo lazerin doğrusal ölçüm hassasiyeti.	24
Şekil 2.8 : Jetson TX1	24
Şekil 2.9 : Elektronik donanımın akış diyagramı.	25
Şekil 3.1 : DAGU araç platformu.	28
Şekil 3.2 : IMU verisi.	29
Şekil 3.3 : Enkoder konum verisi.	30
Şekil 3.4 : Mobil araç.....	31
Şekil 3.5 : 3B LIDAR sistemi ve araç entegrasyonu.	32
Şekil 4.1 : Velodyne LIDAR kullanılarak üretilen nokta bulutu.....	34
Şekil 4.2 : Nokta bulutu dönüşümleri.....	34
Şekil 4.3 : Dosya sistemi grafiği.....	37
Şekil 4.4 : ROS hesaplama grafiği.....	38
Şekil 4.5 : Düğüm-Başlık iletişim grafiği.....	40
Şekil 4.6 : Mobil araç dönüşüm çerçevesi (TF).....	41
Şekil 4.7 : Sapma üst tarama (Yawing scan top) metodu	42
Şekil 4.8 : Mesafe ve nokta bulutu verileri.....	42
Şekil 4.9 : Mesafe ve nokta bulutu verilerinin rviz'deki görüntüleri	43
Şekil 4.10 : ROS-Enkoderli dc motor senkronizasyon sorunu	44
Şekil 4.11 : Kapalı bir ortamın 3 boyutlu görüntüsü	45
Şekil 4.12 : Octree modelinin yapısı.....	46
Şekil 4.13 : Bir aracın farklı voxel çözünürlüklerindeki hali.	46
Şekil 4.14 : Nokta bulutundan doluluk haritalarına dönüşüm aşaması	47

Şekil 4.15 : Laboratuvar ortamının 3B haritadaki görüntüsü (OctoMap).....	47
Şekil 4.16 : ROS navigasyon yığını.....	49
Şekil 4.17 : Sensör tf dönüşümü	51
Şekil 4.18 : Navigasyon akış şeması.....	52
Şekil 4.19 : Otonom golf aracı üzerine entegre edilen sistem	53
Şekil 4.20 : En yakın engel tespiti (Mor top en yakın engeli gösterir)	53
Şekil 4.21 : Engel tespit edilmediğinde PointStamp noktasının konumu.....	54
Şekil 4.22 : Hareketli sistemde engel tespiti deneyinde kullanılan mobil araç.	55
Şekil 4.23 : Move base kullanılarak oluşturulan maliyet haritası (local costmap)	56
Şekil 4.24 : Haritada nokta bulutlarının engeli tespit ettiğinde oluşan görüntü	57
Şekil 4.25 : Lazer tarayıcının dönmesiyle haritada tespit edilen engeller.	57
Şekil 4.26 : Araç yerelleştirme deneyi.....	58
Şekil 4.27 : Move base kullanılarak araç yerelleştirme deneyi	59
Şekil 4.28 : Navigasyon deneyi 1	60
Şekil 4.29 : Navigasyon deneyi 2	61



OTONOM ARAÇLAR İÇİN 2B LAZER TARAYICI KULLANILARAK YENİ 3D LIDAR SİSTEMİ ELDE EDİLMESİ VE ENGEL TESPİTİ

ÖZET

Son yıllarda otonom araçlar üzerinde yapılan çalışmalar gittikçe artmaktadır ve özellikle engel tespiti, haritalama, konum tespiti, rota planlaması, çarpışmadan kaçınma gibi alanlarda yoğun çalışmalar yapılmaktadır. Otonom araçların bu görevleri yerine getirebilmesi için LIDAR, GPS, IMU, RADAR, kamera, mesafe sayacı (odometre), kızıl ötesi ve ultrasonik sensörler gibi algılama sistemleri kullanılır. Bu algılayıcılardan biri olan ve 3 boyutlu lazer tarama tekniği ile çalışan LIDAR, engelleri tespit etmek, görselleştirmek ve çevresel bilgileri elde etmek amacıyla kullanılır. 3 boyutlu lazer tarayıcılar olarak da adlandırılan bu algılayıcılar, yüksek çözünürlük, geniş ölçüm aralığı (250 metreye kadar), yansıtıcı ve zor yüzeylerle başa çıkabilme ve küçük objeleri tespit edebilme gibi özelliklere sahiptir. Fakat oldukça yüksek maliyetli ve büyük boyutlarda olmaları bu algılayıcıların kullanımının yaygınlaşmasına engel teşkil etmektedir. Bu çalışmada yüksek maliyetli ve büyük boyutlu 3B LIDAR'lar yerine, 2B lazer tarayıcı kullanılarak çok daha düşük maliyetli, portatif ve 360⁰ görüş alanına sahip yeni bir 3B LIDAR sistemi geliştirilmesi amaçlanmıştır.

3B LIDAR'ın elde edilmesi temel olarak 2B lazer tarayıcının belli bir eksen etrafında dönmesine ve bu lazer tarayıcı tarafından yayınlanan LaserScan verilerinin bir yazılım platformunda nokta bulutu verilerine dönüştürülmesine dayanmaktadır. Nokta bulutu, tespit edilecek nesnenin yatay ve dikey yönde belirli bir açı altında nokta dizileri şeklinde taranmasının sonucunda elde edilen noktasal görüntüye verilen bir isimdir. Elde edilmeye çalışılan yeni LIDAR sistemi bir bütün olarak incelendiğinde, istenilen verimlilikte çalışabilmesi için üzerinde barındırdığı bileşenleri koordineli olarak haberleşirmesi gerekir. Bu nedenle, bu yazılımsal ve donanımsal haberleşmeyi yönetmek amacıyla açık kaynaklı, ücretsiz ve robot bileşenlerinin bilgisayar ile kontrol edilmesini sağlayan bir yazılım platformu olan ROS (Robot Operating System) kullanılmıştır. ROS platformunda oluşturulan algoritmaların LIDAR sistemiyle bütünleşmesi sağlanarak daha güvenli ve istikrarlı çalışan bir sistem elde edilmesi amaçlanmıştır.

3 boyutlu taramanın yapılabilmesi için standart bir 2B lazer tarayıcı ve bir dönme elemanı gereklidir. 2B lazer tarayıcı olarak HOKUYO-URG-04LX-UG01, dönme elemanı olarak da ROS platformunda hazır kütüphanesi bulunan servo motor yerine sistemin çalışma hızını arttırmak amacıyla enkoderli dc motor kullanılmıştır. Normalde ROS platformunda hazır kütüphanesi bulunan servo motorun ROS'dan yayınlanan açı verisine abone olmasıyla ROS- Servo haberleşmesi sağlanmaktadır. Fakat enkoderli dc motorun ROS platformunda kütüphanesinin bulunmaması, ROS'dan gelen açı ile enkoderden gelen açının eşleşmemesi ve veriler arasındaki frekans farklılıklarından dolayı ROS-Servo haberleşmesinde uygulanan yöntemin kullanılması imkânsız hale gelmiştir. Bu nedenle ROS-Enkoderli dc motor

haberleşmesinde farklı bir metod denenmiştir. Bu metoda göre enkoderden elde edilen açı verileri seri haberleşme yolu ile ROS platformuna gönderilmiş ve gönderilen veri seri iletişimden (serial) okunduktan sonra ROS platformuna uygun hale getirilerek yayın yapılmıştır. Böylelikle ROS-Enkoder haberleşmesi sağlanmıştır.

2B lazer tarayıcı ve dönme elemanının kombinasyonu oldukça önemlidir. Çünkü farklı tarama düzlemleri ve dönme eksenleri farklı görüş alanlarının oluşmasına sebep olur. Bu çalışmada sapma üst taraması (yawing scan top) olarak adlandırılan dikey tarama düzlemi ve z dönme eksenine sahip tarama metodu kullanılmıştır. Ayrıca, 3 boyutlu görüntü elde etme çalışmalarının yürütülebilmesi için, lazer tarayıcı, dc motor, pileksiglas masası ve lazer tarayıcı ile dc motor arasındaki veri bağlantısını sağlayan Slip-ring kullanılarak bir test platformu oluşturulmuştur.

Bir sensör sisteminden elde edilen çevre bilgisi engel tespitinin en önemli aşamalarından biridir. Bundan dolayı öncelikle i5 işlemciye sahip dizüstü bilgisayar kullanılarak oluşturulan 3B LIDAR sistemiyle kapalı bir ortamın 3 boyutlu görüntüsü elde edilerek çevre hakkında bilgi edinilmiştir. Ardından bu sistem bir otonom golf aracına entegre edilerek hareketsiz olan bu araca en yakın engelin mesafesi ve açısı tespit edilmiştir. LIDAR'ın en yakın engel olarak zemini ve otonom aracı göstermemesi için belirlenen mesafe ve açı değerlerine çeşitli filtrelemeler uygulandı. Yapılan bu filtrelemelerden sonra hem lazer tarayıcının bir tam turu tarama süresi azaltıldı hem de araca en yakın engelin bulunduğu konum hatasız bir şekilde elde edildi.

Geliştirilen 3B LIDAR sistemi, hareketsiz golf aracında test edildikten sonra 3B tarama kabiliyetlerinin hareketli bir platformda denenmesi için bir mobil robota entegre edildi. Bu mobil robot vasıtasıyla sisteme hareket kazandırılarak en yakın engelin konumunun hareketli bir ortamda tespit edilmesi amaçlanmıştır. Tespit edilen engelin mesafe ve açısı ROS görselleştiricisi olarak adlandırılan RVIZ platformunda PointStamped noktası olarak gösterilmiştir. Belirlenen mesafe aralıklarında hiçbir engel tespit edilmediğinde PointStamped noktası sistemin merkezinde konumlanmıştır.

Bilgisayar kullanılarak elde edilen 3 boyutlu nokta bulutu verisi ve yapılan deneylerden sonra en az bilgisayar kadar yüksek performans sağlayan ve çok daha küçük boyutlu olmasından dolayı sistemin daha portatif hale gelmesine imkân sağlayan Jetson TX1 geliştirme kartı kullanıldı. Jetson TX1 geliştirme kartını ve 3 boyutlu görüntü için gereken bütün komponentleri içinde barındıran bir mekanik model tasarlanarak 3B LIDAR sisteminine portatif özellik kazandırıldı. Bilgisayarı taşımak için kullanılan mobil araçtan sonra yeni oluşturulan portatif LIDAR platformuna hareket kazandırmak ve haritalama, lokalizasyon ve navigasyon işlemlerinde kullanmak amacıyla 6 tekerlekli diferansiyel sürüş modeline sahip ve ortadaki iki tekerinde enkoderli dc motor bulunan yeni bir mobil araç tasarlandı. Tasarlanan mobil araç üzerine LIDAR platformu entegre edildi ve oluşturulan sistem navigasyon işlemine hazır hale getirildi. Navigasyon için Gezinme Yığını (Navigation Stack) olarak adlandırılan, aracın bir noktadan diğerine otonom olarak hareket etmesi ve hareket esnasında önüne çıkan bütün engellerden kaçınması için kullanılan bir dizi algoritma ve ros düğümü kullanıldı. Mobil araca navigasyon işleminin yaptırılabilmesi için öncelikle araç üzerinde bulunan sensörlerden elde edilen veriler kullanılarak aracın bulunduğu ortamın bir haritası oluşturuldu. Harita oluşturulduktan sonra aracın bu harita üzerinde yerinin belirlenmesi gerekir. Çünkü

uygun bir navigasyon gerçekleřtirmek için aracın haritaya göre hangi konumda olduđunun ve hangi yöne baktıđının bilinmesi gerekir. Bu yüzden haritalama işleminden sonra ROS platformunda yerelleřtirme olarak ifade edilen konum kestirimi işlemi gerçekleřtirildi. Daha sonra aracın mevcut yerini ve gitmek istediđi konumu girdi olarak alan ve belirlenen noktaya en iyi ve en hızlı bir şekilde gidilmesini sađlayan yolu çıktı olarak veren yol planlaması işlemi gerçekleřtirildi. Son olarak kullanılan gezinme yığını algoritması sayesinde aracın odometri verilerini (IMU ve Enkoder) ve lazer tarayıcı verisini kullanarak engellere çarpmadan ve kaybolmadan belirlenen hedefe ulaşması sađlanmıştır.

Son olarak, bu çalışmada ortamın 3 boyutlu haritasının temsil edilmesi için oktektörlere dayanan, özellikle C++ için veri yapıları ve eřleme algoritmaları sunan ve ros içerisinde bulunan açık kaynaklı bir çerçeve olan OctoMap kütüphanesi kullanıldı. Birçok robotik sisteme kolayca entegre edilebilen OctoMap sayesinde nokta bulutu verileri kullanılarak ortamın 3 boyutlu haritası elde edildi.





OBTAINING A NEW TYPE 3D LIDAR SYSTEM USING 2D LASER SCANNER FOR AUTONOMOUS VEHICLES AND OBSTACLE DETECTION

SUMMARY

In recent years, studies on autonomous vehicle technology have increased. Autonomous vehicles, also called UGV (Unmanned Ground Vehicles), are vehicles that have the ability to move on their own without the need for human intervention. These vehicles are able to detect objects in their surroundings thanks to the various sensors they carry. Obstacle distance detection, crash monitoring and reporting, accident avoidance, path planning, situation awareness, localization, mapping, and navigation are basic research topics for autonomous vehicles. To carry out the aforementioned tasks sensing systems –RADAR, IMU, GPS, LIDAR, camera, distance meter (odometer) - are used. Distance, velocity, and the angle of the objects are detected by the usage of Radio Detection and Ranging systems (RADAR), which can adapt to almost all weather conditions. However, RADAR systems cannot be used to detect the objects. IMU (Inertial Measurement Unit), one of the sensors used in autonomous vehicles, is a sensor that provides angular velocity and acceleration data to monitor the dynamically changing motion of the vehicle. It is commonly used with devices that provide location detection called GPS (Global Positioning System). While IMU provides relative position and motion information, the GPS provides absolute position and motion information. That is, when GPS determines the absolute position of the vehicle, the IMU gives information about how far the vehicle is from a position. The odometer is also used to measure the distance traveled by the vehicle. This sensor has a crucial role in the localization process. The cameras known as the eye of autonomous vehicles are used to detect obstacles, signs and traffic signals by collecting color information about surfaces within its field of view. Finally, LIDAR, one of the sensors used in autonomous vehicles and working with 3D laser scanning technique, is used to detect obstacles, visualize and obtain environmental information. These sensors, also called 3D laser scanners, have features such as high resolution, wide measurement range (up to 250 meters), detect small objects and ability to cope with reflective/difficult surfaces. At the same time, it can provide a 360⁰ 3D view of all obstacles around a vehicle. For this reason, LIDAR is the most suitable sensor for estimating the shape of objects according to RADAR and cameras. However, disadvantages such as high cost, being affected by weather conditions and having large dimensions prevent to their become widespread. In this study, it was aimed to develop a new type 3D LIDAR system with much lower cost, portable and 360⁰ fields of view by using 2D laser scanner instead of high costing and large dimension 3D LIDARs and determine the closest obstacle with this new LIDAR system.

The acquisition of 3D LIDAR is mainly based on the rotation of the 2D laser scanner around a certain axis and the fact that data of LaserScan, which is published by this laser scanner, is converted into point cloud data on a software platform. When the new LIDAR system to be obtained is examined as a whole, it needs to communicate

in coordinately the components it contains in order to operate at the desired efficiency. In order to manage this software and hardware communication, ROS (Robot Operating System), which is a software platform that enables open source, free, multi-lingual, tools-based and robot components that can be controlled by the computer, is used. A ROS system consists of nodes, which performs computation, communicating with other nodes using Publisher/Subscriber messaging model. A system is comprised of many nodes and every node is responsible for one task. These nodes communicate with each other through messages published by topics. Topics carry messages that send information to other nodes. They are also called buses that used for moving messages from the node to the node. When a node sends a message via a topic, it means that the node is publishing a topic. In a similar vein, when a node receives a message via a topic, it means that the node is subscribing to a topic. Therefore, a node can publish many topics and subscribe to many topics. This study aims to obtain a more secure and stable system by integrating the algorithms created in this platform with the LIDAR system.

A standard 2D laser scanner and mechanical rotational element are required for 3D scanning. In this study, HOKUYO-URG-04LX-UG01 was used as a 2D laser scanner and dc motor with encoder was used instead of a servo motor as the rotational element with a ready-made library on ROS platform. This 2D laser scanner has compact dimensions and its weight is only 160gr. It has a field of view 240° and an angular resolution of $0,352^{\circ}$. Also, it has a maximum scanning range of 5.6m and minimum scanning range of 0.02m. This sensor is designed for indoor environment use only. Normally, the servo motor with the ready-made library on the ROS platform is subscribed to the angle data published by the ROS and ROS-Servo communication is provided. However, it is impossible to use the method applied in the ROS-Servo communication due to the angle of the ROS and the angle of the encoder do not match, which is also because of the frequency difference between the data provided by ROS and encoder. Also, the absence of an encoder library in the robot operating system is another reason why this method cannot be used. Because of the mentioned reasons, a different method has been tried in the ROS-Encoder communication. According to this method, the angle data obtained from the encoder has been sent to the ROS platform via serial communication and after the transmitted data read out from the serial, it was published in accordance with the ROS platform. So that, ROS-Encoder communication has been provided.

The combination of the 2D laser scanner and the mechanical actuator has a crucial role in the system, because of the fact that different scan planes and rotation axes cause different view fields to be formed. There are four types of laser scanning methods. These scanning methods are called as pitching scan, rolling scan, yawing scan and yawing scan top. Pitching scan has a horizontal scan plane and pitch up and down. Rolling scan is rotating around the center of the laser scanner. The yawing scan and yawing scan top have a vertical scan plane and these are rotating around the upright z-axis. In this study, the yawing scan top scanning method was used. According to this method, the vertical scanning plane was taken as the scanning plane, and z-axis was selected as the rotation axis.

In the mechanical design of the system was used a slip-ring component, which transfers the data on the rotating laser scanner and the dc motor with an encoder to the computer environment was used. This component prevents cables from being twisted and damaged between the rotating system and the fixed system. In addition, a test platform has been created using a laser scanner, dc motor with encoder,

plexiglass table, slip-ring. Simultaneously, the L-shaped aluminum rotary support, called the Gimbal, was used in the mechanical design of the system to provide the ability to rotate around a single axis. And also coupling was used to connect the dc motor and LIDAR.

Environmental information obtained from a sensor system is one of the most important steps in detecting obstacles. In order to be able to detect an obstacle, it is necessary to have information about the environment in which the obstacle is located. Thus, in this study, the 3D image of a closed environment was obtained by using the created 3D LIDAR system, so that the information about the environment was also obtained. After following these steps, the system was integrated into an autonomous golf vehicle and distance and angle of the nearest obstacle to this non-moving vehicle have been determined. Various filters have been applied to the distance and angle values determined so that LIDAR does not show the ground and autonomous vehicle as the closest vehicle. After applying these filters, the full-round scanning time of the 3D laser scanner was reduced and the position where the nearest obstacle was found was detected without error.

After testing on a non-moving golf vehicle, the developed 3D LIDAR system has been integrated into a mobile robot to test 3D scanning capabilities on a moving platform. Through this robot, the system has been moved and it has been aimed to determine the position of the nearest obstacle in a moving environment. The distance and angle of the detected obstacle have been shown as the PointStamped point in the RVIZ platform, which is referred to as the ROS visualizer. The PointStamped point has been located at the center of the system when no obstacle is detected at the specified distance ranges.

After the 3D point cloud image and experiments obtained by using the computer, the Jetson TX1 development card was used which provided at least as high as a computer performance. Since the Jetson TX1 is much smaller in size than the computer, it made it possible to make the 3D LIDAR system more portable. Then, a mechanical model was designed that includes the Jetson TX1 development board and all the components required for 3D point cloud viewing. Thus, the 3D LIDAR system became more portable. After the mobile vehicle used to carry the computer, a new mobile vehicle was designed to give motion to the newly created portable 3D LIDAR system and to be used in mapping, localization, and navigation. This mobile vehicle consists of 6 wheels and an aluminum frame, each driven by a separate motor. This vehicle, which works according to the differential driving method, has the dc motor with encoders attached to the medium wheels. Thanks to these encoders, it is provided to the movement control of the vehicle's wheels which based on the amount of rotation. The LIDAR platform was integrated on the designed mobile vehicle and the system was made ready for navigation. A number of algorithms and ros nodes, called Navigation Stack for navigation, are used to autonomously move the vehicle from one point to another and avoid all obstacles that may occur during movement. In order to be able to navigation with the mobile vehicle, a map of the vehicle environment was created by using the data obtained from the sensors on the vehicle. Once the map is created, the location of the vehicle on this map must be determined. Because in order to carry out an appropriate navigation, it is necessary to know which position the vehicle is in relation to the map and which direction it is looking at. So, after the mapping process, on the ROS platform performed location estimation, which is expressed as localization. Then, the path planning process was performed, taking as input the current position of the

vehicle and the position it's wanted to go. Thanks to this path planning, the mobile vehicle goes to the designated spot in the fastest and best way. Finally, thanks to the navigation stack algorithm used, it was provided the vehicle is able to reach the specified target without collision to the obstacles by using odometry data (IMU and Encoder) and laser scanner data.

Finally, in this study, it was used the OctoMap library, an open source framework within ROS, that provides a 3D map of the environment, specifically for C++ data structures and mapping algorithms. Thanks to the OctoMap, which can be easily integrated into many robotic systems, the 3D map of the environment was achieved using point cloud data.



1. GİRİŞ

İnsansız kara araçları (İKA), içinde insan unsurunun bulunmasına gerek duymayan, yerle temas halinde olup karada hareket edebilme özelliğine sahip olan ve insan kontrollü, yarı otonom ve otonom olma gibi farklı fonksiyonları içinde barındıran araçlardır. İKA'lar son yıllarda askeri, sivil ve bilimsel araştırma amaçlı birçok alanda kullanılmaya başlanmıştır. Askeri alanda, bomba, mayın gibi tehdit unsurlarının saptanması ve imhası, tespit edilen nesnelere uzaktan müdahale, lojistik destek, istihbarat, keşif, gözetleme ve tehlikeli nesne takip görevlerinde can kaybının önüne geçerek nitelikli insan gücünü koruma gibi amaçları gerçekleştirmek için kullanılmaktadır. Sivil alanda ise arama kurtarma, yangın söndürme, tarım, yük taşıma, kütüphane ve hastane gibi kapalı alanlarda insanlara rehberlik etme gibi birçok amaç doğrultusunda insanlığın hizmetine sunulmuştur. Genel olarak, uzaktan kontrollü insansız kara aracı ve otonom kara aracı olmak üzere iki temel sınıfta incelenirler. Uzaktan kontrollü insansız kara aracı, bir insan operatör tarafından, üzerinde taşıdığı algılayıcılar kullanılarak kontrol edilen araçlardır. Askeri alanda en yaygın olarak kullanılan sürücüsüz araç modelidir. Hem askeri hem de sivil alanda kullanılan bir uzaktan kontrollü kara aracı olan PackBot Şekil 1.1' de gösterilmiştir [1].



Şekil 1.1 : IRobot firması tarafından üretilen uzaktan kontrollü İKA olan PackBot [1].

Otonom araçlar ise hiçbir insan müdahalesine gerek duymadan üzerinde taşıdığı algılayıcılar vasıtasıyla çevresel bilgi edinip kendi kontrol mekanizmasını geliştirebilen ve kendi aldığı kararları uygulayabilen sistemlerdir. Yapılan bu tanım tam otonom araçlar için geçerlidir. Çünkü yarı otonom olup sürücü ile hareket eden ve gerektiğinde sürücünün müdahale edebildiği araçlar da otonom araç olarak adlandırılabilir. Son yıllarda bu araç teknolojisi üzerinde yapılan bilimsel araştırmalarda ulaşılmak istenen temel hedef, insan faktöründen kaynaklanan kaza riskini tamamen ortadan kaldırmak, trafik sorununu minimize etmek, zaman ve yakıt tasarrufu sağlamak, park sorununu ortadan kaldırmak ve kazalara sebebiyet veren kör nokta gibi problemlerin üstesinden gelmektir. Bu alandaki ilk denemeler 1920'li yıllara, daha ileri seviyedeki çalışmalar ise 1990'lı yıllara dayansa da sensör ve kontrol teknolojisinin yeterli düzeyde olmamasından dolayı istenilen seviyeye ulaşılamamıştır.

1.1 Otonom Araçların Gelişim Süreçleri

Otonom araçlar bugünkü aşamaya gelene kadar çeşitli süreçlerden geçmişlerdir. Yapılan her araştırma, otonom araç teknolojisine ciddi katkılar sağlamıştır. Bu alanda yapılan uygulama, 1958 yılında General Motor firması tarafından deneysel amaçlı üretilen Chevrolet marka araçtır [2]. Yapılan bu çalışmada aracın direksiyonu yönetme kolu (joystick) ile, pedalları ise bir acil durum freni ile değiştirilmiş ve araca otonom olarak frenleme ve gerektiğinde şerit değiştirebilme özelliği kazandırılmıştır. Şekil 1.2 'de gösterilen aracın önünde bulunan gösterge, otomobilin hızı ve öndeki araca olan uzaklığı hakkında bilgi vermektedir.



Şekil 1.2 : GM firması tarafından deneysel amaçlı üretilen otonom araç [2].

1960 ve 1970'li yıllarda ise başlangıçta roketler için bir platform olarak tasarlanan fakat 1979 yılında Hans Moravec tarafından geliştirilerek otonom özellik kazandırılan bir araç geliştirildi. [3]. Stanford Cart olarak adlandırılan bu araç 5 saat boyunca hiçbir insan müdahalesi olmadan sandalyelerle dolu bir odayı başarılı bir şekilde geçebilmesiyle bilinir (Şekil 1.3).



Şekil 1.3 : Stanford Cart [3].

1987 yılına gelindiğinde otonom araç teknolojisine çok büyük katkı sağlayan ve otonom araçların öncüsü olarak kabul edilen alman mühendis Ernst Dickmanns, üzerinde kamera, mikroişlemci ve birçok sensör barındıran ve güçlü bir yazılımla donatılan VaMoR'lar üzerinde yeni bir araç sistemi geliştirdi (Şekil 1.4) [4]. Geliştirilen bu araçlar, yol üzerinde aracın önündeki ve arkasındaki engelleri tespit edebilme özelliğine sahiptir. Dickmann'ın bu alandaki en büyük katkısı görüntüleme sisteminin yabancı bir gürültüyü filtreleyip sadece ilgili nesnelere odaklanmasını sağlayan bir sistem geliştirmesidir. Ortaya çıkardığı bu yenilik, bugün dahi otonom araçların engel ve tehlikeleri tespit etmesine yardımcı olmaktadır. Fakat günümüzde kullanılan sensör ve kontrol teknolojisinin bu dönemde yeterli düzeyde olmamasından dolayı çukur ve buna benzer yol üzerindeki bazı engellerin tespit edilmesinde çeşitli sorunlar yaşanmıştır. Bu yüzden güvenli yolculuk yapılması için yeterli düzeyde değildi.



Şekil 1.4 : Dünyanın ilk gerçek otonom aracı olarak kabul edilen VaMoRs Mercedes [4].

1995 yılında Carnegie Mellon Üniversitesinden araştırmacı Dean Pomerleau ve Todd Jochem tarafından % 98,2'lik bir otonom sürüş yüzdesiyle Navlab5 adında bir otonom araç üretildi (Şekil 1.5) [5]. Taşınabilir bilgisayar, ön cama montajlı kamera, GPS alıcısı ve diğer bazı yardımcı donanımlara sahip olan bu araç hız ve fren kontrolü dışında tamamen otonomdur ve bu konuda yapılan testleri yüksek başarı yüzdesiyle geçmiştir.



Şekil 1.5 : Navlab5 [5].

2000'li yıllara gelindiğinde algılama ve kontrol sistemlerindeki gelişim otonom araç teknolojisinin de gelişimine büyük katkı sağlamıştır. 2004 yılında Amerika Savunma Bakanlığı İleri Araştırma Projeleri Ajansı (DARPA) tarafından insansız araç çalışmalarını desteklemek amacıyla bir yarışma düzenlenmiştir. Yarışmanın amacı, 175 mil uzunluğundaki bir çölü 10 saatten daha kısa sürede tamamlayan ve tamamiyle otonom olarak hareket eden araçların geliştirilmesidir. 2005 yılında

düzenlenen bu yarış, Stanford üniversitesinin Stanford yarış takımı tarafından geliştirilen ve yazılım sistemi makina öğrenimi ve olasılıksal muhakeme gibi yapay zeka teknolojisine dayanan Stanley (Şekil 1.6) [6] adlı araç kazanmıştır [7].



Şekil 1.6 : DARPA tarafından düzenlenen yarışmayı kazanan Stanley [6].

Günümüzde otonom araçlar üzerine yapılan çalışmalarda Google firması ön plana çıkmıştır. Firma, 2014 yılında Koala car adında hiçbir direksiyon simidi, fren pedalı ve gaz pedalı olmadan tamamen otonom olan bir otomobil üretmiş ve bu otomobil üzerinde başarılı testler gerçekleştirmiştir. Araç, Velodyne LIDAR gibi yüksek çözünürlüklü, geniş ölçüm alanına sahip, çevrenin 3 boyutlu görüntülerini veren nokta bulutu verileri üreten ve radarlara göre çok daha küçük boyutlu engelleri tespit edebilen ileri teknoloji ürünü algılayıcılarla donatılmıştır (Şekil 1.7) [8]. Bu algılayıcıların birçok avantajları olmasına rağmen oldukça yüksek maliyete sahip olmaları kullanımlarının yaygınlaşmasına engel teşkil etmektedir.



Şekil 1.7 : Google tarafından üretilen Koala car [8].

1.2 Otonom Araçların Algılayıcı Sistemleri

Araçlarda tam otonomluğun sağlanabilmesi için birbirleriyle uyumlu bir şekilde çalışan birçok algılayıcıya ihtiyaç vardır. Bu algılayıcılar sayesinde otonom araç 360°'lik bir açıda çevresi hakkında kapsamlı bir bilgiye sahip olur ve bu çevresel bilgiler ışığında engelleri tespit edebilir. Ayrıca bu algılayıcılar kullanılarak aracın dünya üzerinde bulunduğu konumu, hızı, direksiyon pozisyonu, fren durumu ve bulunduğu konumdan ne kadar uzakta olduğuyla alakalı bilgiler de elde edilebilir. Bu yüzden otonom araçlar kamera, LIDAR, Radar gibi birçok algılayıcıyla donatılmıştır. Bir otonom araçta olması gereken algılayıcılar Şekil 1.8 'de gösterilmiştir [9].



Şekil 1.8 : Otonom araç donanımı [9].

1.2.1 GPS

Küresel konum belirleme sistemi olarak adlandırılan GPS, uydulardan yayımlanan radyo sinyalleri vasıtasıyla günün her saatinde ve her türlü hava koşullarında aracın dünya üzerindeki konumunun tespit edilebilmesi için kullanılan bir sensör sistemidir. (Şekil 1.9) [10]. Bu sensörün konum belirleme görevini yerine getirebilmesi için birden çok uydudan haberleşmesi gerekir. 3 uydudan alınan uzaklık verileriyle aracın enlem ve boylam bilgileri elde edilebiliyorken 4 uydudan alınan uzaklık verileriyle aracın enlem ve boylam bilgilerinin yanısıra yükseklik bilgisi de elde edilebilmektedir. Fakat uydudan sinyallerinin alınmasının mümkün olmadığı tüneller, sualtı, kapalı ortamlar ve binaların yoğun olduğu yerleşim yerlerinde GPS kullanımı imkânsız hale gelir. Bu gibi GPS kullanımının mümkün olmadığı durumlarda GPS,

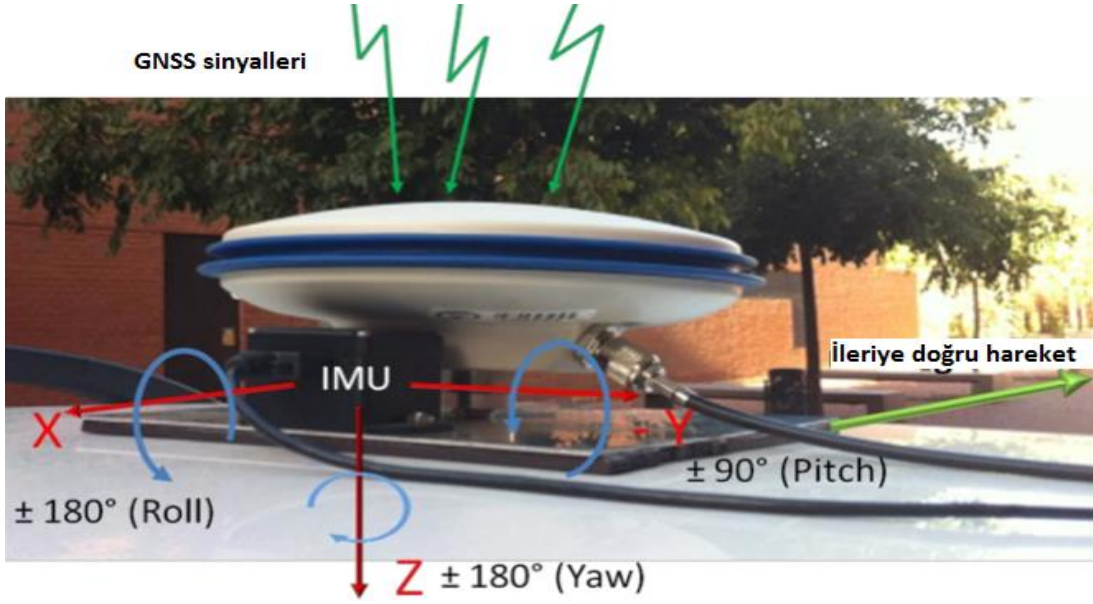
Ataletsel Navigasyon Sistemi (INS) olarak adlandırılan, aracın konumunun tespit edilmesinde GPS'e yardımcı olan ve içerdği jiroskop ve ivmeölçer ile aracın pozisyon, uyum ve hızını da ölçebilen bir sistem ile birlikte kullanılır. Bu iki sensör sisteminin birlikte kullanılmasıyla ortamdaki kaynaklanan sinyal alımı problemleri ortadan kalkarak her türlü alanda aracın konumu mutlak bir şekilde tespit edilebilir.



Şekil 1.9 : GPS [10].

1.2.2 IMU

GPS'in araçların konumunun belirlenmesinde önemli bir görevi olsa da bu algılayıcının yetersiz kaldığı durumlar vardır. Uydu sinyallerinin ulaşmadığı ortamlarda GPS etkisiz olduğu için otonom araçlar konum bilgilerini alabilmek için herhangi bir harici sinyale gerek duymayan Ataletsel Ölçüm Birimi olarak ifade edilen IMU'ları kullanır (Şekil 1.10) [11]. IMU, içinde birbirlerine ortogonal olarak monte edilen ivmeölçer ve jiroskop algılayıcıları bulunan bir platformdan oluşur. Bu sayede aracın hızı ve pozisyonu hakkında da bilgi verir. GPS ile aracın mutlak konumu ve hareketi tespit edilirken, IMU ile aracın bir konumdan ne kadar uzakta olduğu bilgisi elde edilir. Bu iki sistem birbirleriyle uyumlu olarak çalışır.



Şekil 1.10 : IMU [11].

1.2.3 Kamera

Otonom araçlarda kullanılan ve genellikle ön cama monte edilen gelişmiş kamera sistemleri temel olarak çevreyi gözlemleme, engelleri tespit etme, trafik ışıklarını, fren lambalarını, sinyalleri, şerit işaretlerini, hız işaretlerini görme gibi görevleri yerine getirirler. Yol işaretlerini ve şerit işaretlerini renkleriyle birlikte görebilmesi kameraları otonom araçlarda kullanılan diğer algılayıcılardan ayıran önemli özelliklerden biridir. Ayrıca, kameralar Radar ve LIDAR'lara göre çok daha ekonomiktir. Kötü hava koşullarında, karanlık ve yoğun güneşli ortamlarda yetersizdirler. Otonom araçlarda kameraların LIDAR ile birlikte kullanılması araca çevre hakkında daha detaylı bilgi edinme olanağı sunar. Çünkü kamera bir engelin rengini algılayırken LIDAR da o engelin türünü tespit eder. Böylece her iki algılayıcıya sahip olan otonom araçlar, karşılaştığı engel hakkında çok daha kapsamlı bilgiye sahip olduğu için diğer araçlara nazaran daha güvenli bir sürüş gerçekleştirirler.

1.2.4 Tekerlek enkoderi

Otonom araçlarda yaygın olarak kullanılan bir diğer sensör de aracın hız ve mesafe bilgilerini tekerleklerin ne kadar döndüğüne bağlı olarak hesaplayan algılayıcılardır. Bu sensörler ölçülen belli bir süre içinde alınan mesafeyi ifade eden hızı kaydederek elde edilen bu bilgileri elektrik sinyali olarak sürüş güvenlik sistemlerine aktarır. Otonom araçlardaki temel lokalizasyon sistemlerinden biri olan ve aracın başlangıç

konumuna göre konumundaki deęişikliklerin tahmin edilmesine dayanan odometri uygulamalarında kullanılır.

1.2.5 Ultrasonik ve kızılötesi sensörler

Ultrasonik sensörler, nesnelere yansıyan yüksek frekanslı ses dalgalarını kullanarak otonom aracın söz konusu nesnenin ne kadar uzakta olduğunu tespit edebilen, düşük hız ve kısa mesafe gerektiren uygulamalarda kullanılan sensörlerdir. Çalışma prensibi olarak (uçuş süresi ilkesi) radarlara benzemelerine rağmen mesafe belirlemede kullanılan dalga türü, basınç ve sıcaklıktan etkilenme dereceleri, etki ettikleri mesafe aralıkları gibi özellikleriyle radarlardan ayrılırlar. Ayrıca oldukça ekonomik oldukları için ucuz sensör grupları içinde yer alırlar. Kızılötesi sensörler de uçuş süresi prensibiyle çalışan ışıktan kaynaklanan çevresel sınırlamalardan etkilenmeyen ve şerit çizgilerini gece ya da gündüz olmasına bakmaksızın algılayabilen sensörlerdir.

1.2.6 Radar

Radyo dalgalarını kullanarak nesnelere mesafelerini, hızlarını, boyutlarını hesaplayabilen; park etme, şerit deęiştirme, tampona yaklaşan engelle karşı sürücüyü uyarma gibi görevleri yerine getirebildiği için genelde otonom aracın ön ve arka tamponlarına yerleştirilen ve 360⁰ tarama yapabilen sensörlerdir. Çalışma prensibi belli yönde yaydıkları radyo dalgalarının engellere çarpıp geri dönmesiyle elde edilen sürenin ölçülmesine dayanır. Kötü hava koşullarında LIDAR ve kameralara göre daha verimlidir. Karlı, sisli, bulutlu ortamlarda radardan gönderilen sinyal etkilenmez. Nesne tespitinde LIDAR'lardan farklı sinyal türü kullanılırlar. Aynı zamanda, LIDAR'lardan çok daha az veri ürettikleri için nesnelere LIDAR'lar kadar hassas algılayamazlar. Çözünürlükleri LIDAR'lara göre çok düşüktür. Fiyat söz konusu olduğunda ise LIDAR'lara göre nispeten daha ekonomiktirler. Şerit çizgilerini, yayaları ve trafik işaretlerini neredeyse hiç algılayamazlar. Çözümlemekte olduğu nesnenin türü ve rengi hakkında bilgi veremezler. Otonom araçların uzak veya yakın mesafede, iyi veya kötü hava koşullarında minimum hata ile hareket edebilmeleri, bu iki sensörün tek başına kullanılmalarından ziyade otonom araçlar için gerekli dięer sensörlerle birlikte kullanılmalarına bağlıdır. Çünkü hiçbir sensör her koşulda mükemmel sonuç veremez. Bu mükemmelliği algılayıcıların birlikte kullanılması sağlar.

1.2.7 LIDAR

Son yıllarda geleceğin teknolojisi olarak görülen otonom araç teknolojisi üzerinde yapılan çalışmalarda Google firması ön plana çıkmaktadır. Google'ın otonom aracı üzerinde birbirleriyle senkronize bir şekilde çalışan tam 8 adet sensör bulunur. Bu sensörlerden biri olan ve aracın otonomluğuna büyük katkı sağlayan LIDAR, çevrenin yüksek çözünürlüklü 3 boyutlu haritasının elde edilmesinde, insan gözüyle görünmeyen lazer ışığını kullanarak insan, yol işaretleri, araç ayırt etmeksizin bütün engellerin maksimum 250 metreye kadar mesafe, açı gibi özelliklerinin ayrıntılı bir şekilde tespit edilmesinde kullanılır. Işık tabanlı radarlar olarak da adlandırılırlar. Fakat radarlardan farklı olarak engelleri, saniyede milyonlarca kez yaydıkları lazer ışınları sayesinde daha büyük algılama hassasiyeti ve daha yüksek çözünürlükte tespit ederler. Örneğin, Google otonom aracında kullanılan ve 360° görüntü elde etmek için aracın üzerine monte edilen Velodyne LIDAR (Şekil 1.11) [12] saniyede 1,3 milyon gerçek zamanlı olarak yerkürenin 3 boyutlu haritası gibi işleyen nokta bulutu verisi üreterek çok yüksek çözünürlüklü 360° görüntü sağlar. LIDAR'ları diğer sensörlerden ayıran bu avantajlarına rağmen kötü hava koşullarında yetersiz kalmaları ve özellikle çok yüksek bir maliyete sahip olmaları bütün otonom araç projelerinde kullanılmalarının önüne geçmektedir.



Şekil 1.11 : 3D velodyne LIDAR [12].

1.3 Tezin Amacı

Bu çalışmada öncelikle otonom araçlarda kullanılmasıyla aracın çevre ile olan haberleşmesine çok büyük avantaj sağlayan fakat yüksek maliyeti nedeniyle yaygınlaşamayan büyük boyutlu 3B LIDAR'ların yerine 2 boyutlu lazer tarayıcılar kullanılarak çok daha düşük maliyetli, daha portatif ve **360⁰** görüş alanına sahip yeni bir 3B LIDAR sistemi geliştirilmesi amaçlanmıştır. Bu sistemin elde edilmesi, sabitken belli bir tarama aralığına sahip olan 2 boyutlu lazer tarayıcının bir dönme elemanı vasıtasıyla belirlenen bir eksen etrafında dönmesine ve tarayıcı tarafından yayınlanan 2 boyutlu LaserScan verilerinin nokta bulutu verilerine dönüştürülmesine dayanmaktadır. Saniyede yüzbinlerce nokta bulutu verisi üretildiği ve üretilen bu verilerin akış hızının yüksek olması için hem bilgisayar hem de Jetson TX1 adında yüksek kapasiteli bilgisayar sistemi kullanılmıştır. Yapılan çalışmanın ikincil amacı ise geliştirilen bu yeni sistemin otonom mobil araca monte edilerek belli mesafe aralığındaki engelleri 3 boyutlu olarak tespit etmesi ve tespit ettiği engelin mesafe ve açısı hakkında bilgi vermesidir. Mobil araçla yapılan çalışmalardan sonra tasarlanan sistem otonom kapalı ortam aracına kurulacaktır. Mevcut aracın ön tarafında 5,6 metreye kadar **240⁰** 'lik bir açıda engelleri tespit edebilen iki tane 2B lazer tarayıcı vardır. Fakat ne yazık ki araç dönerken veya yanlardan herhangi bir engel çıktığında bu engelleri algılayamaz. Bu araç, geliştirilen bu yeni 3B lazer tarayıcı sisteminin yardımıyla bu nesnelere tüm yönlü olarak tespit edebilecektir. Geliştirilen bu yeni sistemin otonom araç araştırmalarında yüksek maliyetinden dolayı temin edilmesi mümkün olmayan 3B LIDAR'lar yerine kullanımının yaygınlaşması öngörülmektedir.

1.4 Literatür Taraması

LIDAR, lazer ışığını kullanarak nesneye olan mesafeyi ölçüp çevrenin 3 boyutlu haritasını çıkarabilen hızlı, yüksek çözünürlük ve doğruluk oranına sahip bir uzaktan algılama teknolojisidir. Birçok farklı çeşide sahip olan bu algılayıcılar, otonom araç ve mobil robot uygulamalarının yanı sıra askeri, tarım, arkeoloji, jeoloji, meteoroloji, atmosferik ortamlar, astronomi, yer ölçümü gibi birçok alanda da kullanılmaktadır [13]. Otonom araç ve mobil robot uygulamalarında ise 3D lazer tarayıcılar kullanılarak engel tespiti [14-18], çarpışmadan veya engelden kaçınma [19-21], eşzamanlı konum tespiti ve haritalama [22-25], navigasyon [26-28], nesne

sınıflandırma [29], rota planlaması [30] gibi farklı alanlarda yoğun arařtırmalar yapılmaktadır. Otonom araç ve robot teknolojisinde en yaygın kullanılan 3 boyutlu lazer tarayıcılar Velodyne firması tarafından üretilen **360⁰** görüş alanına sahip çok ışınlı HDL-64E [31], HDL-32E [32], ve VLP-16 [33] modelleridir. Bu 3B lazer tarayıcılar yüksek çözünürlük, geniş ölçüm aralığı (200 metreye kadar), onları yüksek hızda hareket eden araçlar için uygun hale getiren yüksek edinim hızı ve zor yüzeylerle başa çıkabilme gibi özelliklere sahiptir. Fakat lazer tarayıcıların kullanıldığı çalışmalar oldukça maliyetlidir. Ayrıca kullanılan bazı 3B lazer tarayıcılarının ölçüleri ve ağırlıkları oldukça büyüktür. Örneğin, Google firmasının otonom araç çalışmalarında kullandığı LIDAR sistemi 80 kg ağırlığında ve yaklaşık olarak 75.000 dolardır [34]. Bu nedenle birçok arařtırmacı çalışmalarında standart 2B lazer tarayıcıları belli bir eksen etrafında döndürerek ve elde edilen verileri nokta bulutu verisine dönüřtürerek kendi 3B lazer tarayıcılarını oluşturmuşlardır [35-39]. Belirtilen çalışmalardan farklı olarak bu çalışmada daha ekonomik, küçük boyutlu ve otonom araç çalışmalarının önünü açabilecek bir 3B lazer tarayıcı sistem tasarımı amaçlanmıştır. Ayrıca tasarlanan sistemin ek bilgisayar kullanımı gerektirmemesi de yapılan çalışmayı diğer çalışmalardan ayıran özelliklerdendir. Yani tasarlanan sistemde bilgisayara bağımlı olmadan bilgisayar görevi üstlenen yüksek performansa sahip daha küçük geliştirme kitleleri kullanılarak sistem daha portatif hale getirilmiştir. Yapılan çalışmayı diğer çalışmalardan ayıran bir özellik de tasarlanan sistemin yer ve hava gibi farklı platformlarda kullanılabilmesidir. Son olarak bu çalışmada diğer arařtırmalardan farklı olarak ROS (Robot İşletim Sistemi) [40] olarak adlandırılan bir yazılım platformu üzerinde hazır kütüphaneleri bulunan servo motor yerine enkoderli dc motor kullanılmıştır [41-43].

Bu tezin ilk bölümünde insansız kara araçları ve otonom araçların gelişim süreci hakkında bilgi verilmiş ve otonom araçlar üzerinde kullanılan algılayıcı sistemleri tanıtılmıştır. Aynı zamanda tezin amacıyla birlikte otonom araçlarda kullanılan 3B LIDAR sistemleri ve 2B lazer tarayıcılar kullanılarak oluşturulan yeni LIDAR sistemleriyle ilgili literatür arařtırmasına yer verilmiştir.

İkinci bölümde sırasıyla sistemin mekanik ve elektriksel tasarımı ele alınmış ve kullanılan komponentler tanıtılmıştır.

Üçüncü bölümde geliştirilen LIDAR ile hareketli bir ortamda engel tespiti yapmak için tasarlanan mobil araç hakkında bilgi verilmiştir.

Dördüncü bölümde nokta bulutu kavramı, 3B LIDAR sisteminin elde elde edilmesinde kullanılan yazılım platformu, 2B lazer tarayıcıdan 3B görüntü elde edilmiş aşamaları ve tasarlanan mobil araçla gerçekleştirilen navigasyon işleminden bahsedilmiştir. Ayrıca tasarlanan 3 boyutlu LIDAR'ın tarama özelliklerini ve çevresindeki engelleri tespit edebilme kabiliyetini test etmek amacıyla hem bilgisayar hem de Jetson TX1 geliştirme kartı üzerinde yapılan deneyler açıklanmıştır.

Beşinci ve son bölümde ise sonuçlar, öneriler ve gelecekte yapılması planlanan çalışmalar hakkında bilgi verilmiştir.





2. 3B LIDAR SİSTEMİ TASARIMI

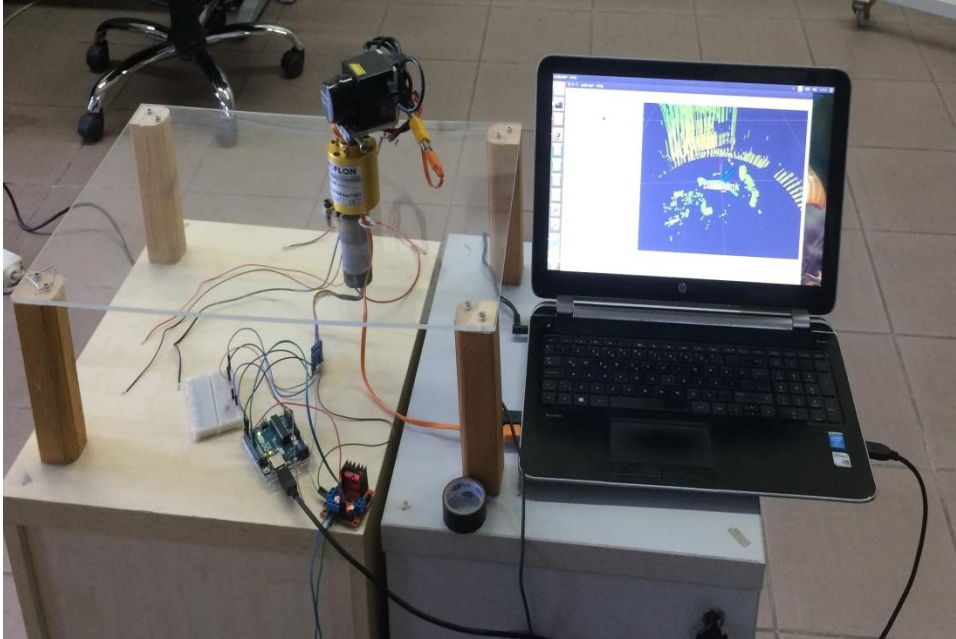
Bu tezde hem otonom araçlarda hem de mobil robotlarda uygulanmak üzere çevrenin 3 boyutlu geometrik görüntüsünü elde edebilen bir sensör sistemi tasarlanmıştır. Bu 3 boyutlu görüntünün elde edilebilmesi için temel olarak uçuş süresi ölçüm prensibine göre çalışan standart 2B lazer tarayıcıya ve bu lazer tarayıcıyı belli bir ekseninde döndürecek mekanik bir dönme elemanına ihtiyaç vardır. Tarayıcıyı döndürmek için servo motor veya dc motor gibi mekanik aktüatörler kullanılabilir. Bu çalışmada hem tarama hızını hem de hassasiyeti arttırmak için yüksek çözünürlüklü bir enkoderli dc motor kullanılmıştır. Ayrıca, olası kalibrasyon problemlerinin önüne geçmek için 2B lazer tarayıcının tarama ekseninin merkezi ile dönme eksenini aynı hizaya getirilmiştir. Farklı tarama düzlemleri ve farklı dönme eksenleri farklı görüş alanlarına sebebiyet verdiği için 2B lazer tarayıcı mekanik aktüatöre 4 temel tarama metodundan biri olan ve dikey tarama düzlemi ve z dönme eksenine sahip sapma üst taraması (yawing scan top) yapabilecek şekilde entegre edilmiştir. Lazer tarayıcı ile dc motor arasında sisteme sınırsız dönüş özelliği kazandırmak, lazer tarayıcının dönüşü esnasında kabloların birbirine dolanmasına engel olmak ve dönen bu sisteme güç ve sinyal iletimini sağlamak için elektromekanik bir cihaz da eklenmiştir. Sistemdeki dc motoru istenen verimde tahrik etmek için bir motor sürücü entegresi ve bu sistemi kontrol edip enkoder verilerini okumak için ise bir geliştirme kartı kullanıldı. 2B lazer tarayıcıdan gönderilen laserscan verileri 3 boyutlu çevre bilgisi veren nokta bulutu verilerine dönüştürülür. Elde edilen bu 3B nokta bulutu verileri çok yoğun bir veriler topluluğudur. Bu veriler topluluğunun ortamın 3 boyutlu görüntüsünü detaylı bir şekilde gösterebilmeleri, akış hızlarının yüksek olmasına bağlıdır. Bundan dolayı donanımsal olarak çok güçlü bilgisayarlara ihtiyaç duyulur. Tasarlanan sistemde önce normal dizüstü bilgisayar vasıtasıyla 3 boyutlu görüntü elde edilmiş daha sonra ise sistemi daha portatif hale getirmek için küçük ve yüksek hesaplama performansı sağlayan bir geliştirme kiti kullanılmıştır.

Tasarlanan 3B LIDAR sisteminde kullanılan mekanik ve elektronik parçalar Çizelge 2.1’ de gösterilmiştir.

Çizelge 2.1 : 3B LIDAR sistemi bileşenleri ve adetleri.

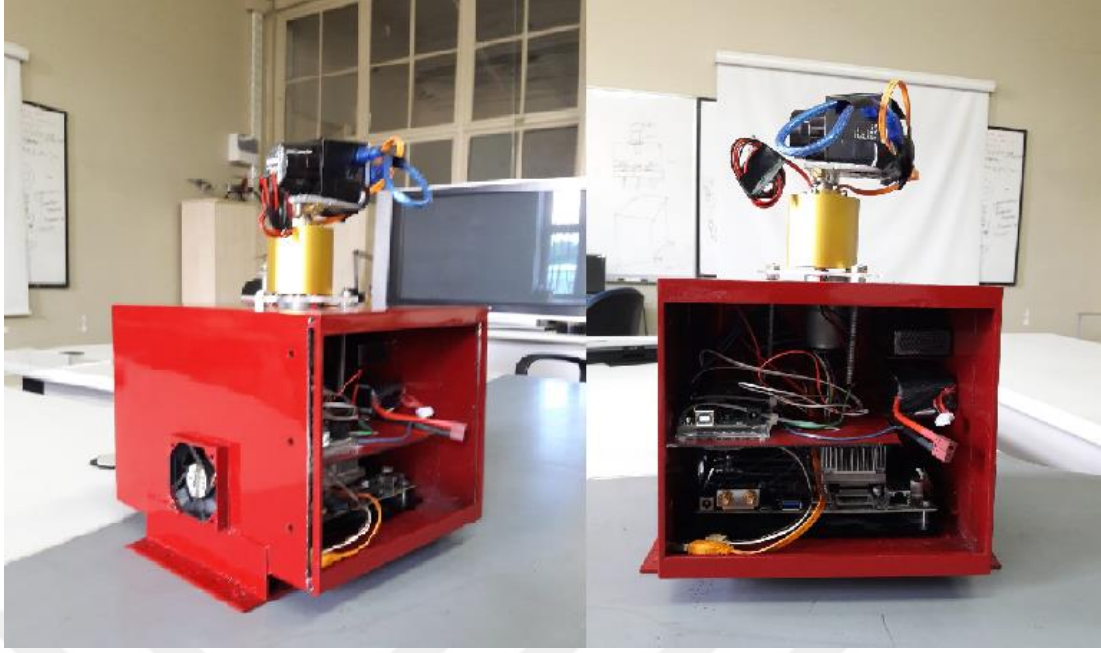
Parça adı	Adet
Hokuyo-URG-04LX-UG01 Lazer tarayıcı	1
USB Slip-Ring	1
Enkoderli Dc Motor	1
Arduino UNO	1
L298N Motor Sürücü	1
Pil	1
Jetson TX1	1
Alüminyum plaka	1
PC	1
Mekanik LIDAR Platformu	1

Bu bölümde yukarıda belirtilen tüm bileşenler detaylı bir şekilde açıklanacaktır. Bilgisayar kullanılarak elde edilen 3B LIDAR platformu şekil 2.1’de gösterilmiştir.



Şekil 2.1 : 3B LIDAR platformu.

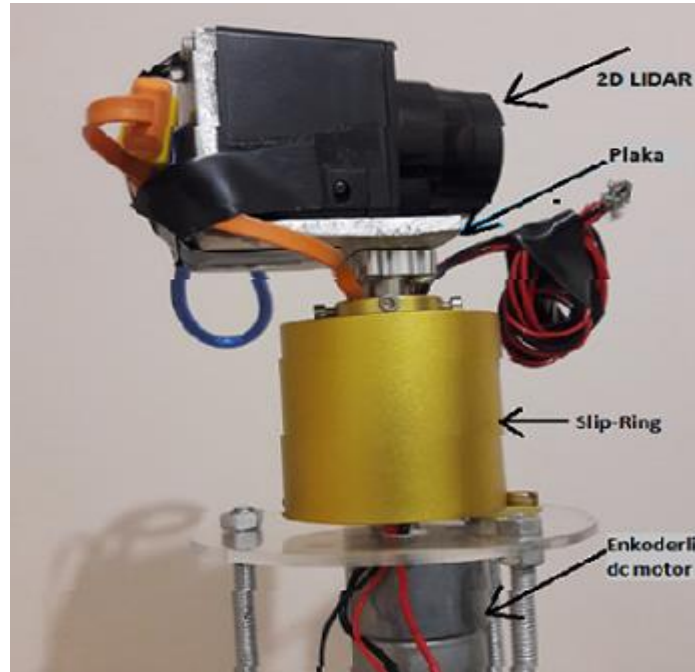
Jetson TX1 geliştirme kartı kullanıldığında ise 3B LIDAR sistemini daha portatif hale getirmek ve hem mobil araçlarda hem de hava araçlarında kullanmak amacıyla 810 gr ağırlığında, 0.36 cm kalınlığında ve 21x22x17 cm boyutlarında bir platform tasarlanmıştır. Tasarlanan platform kullanılarak elde edilen yeni portatif 3B LIDAR sistemi ise şekil 2.2’de gösterilmiştir.



Şekil 2.2 : Portatif 3B LIDAR platformu.

2.1 Mekanik Tasarım

2B lazer tarayıcı, dc motor, slip ring, arduino ve motor sürücünden oluşan sistem, test edilme aşamasında pleksiglass malzemesinden oluşan bir plakaya bağlanmıştır.



Şekil 2.3 : 3B LIDAR prototipi.

Ayrıca, lazer tarayıcı ile enkoderli dc motorun birbirine bağlanması için L şeklinde alüminyum bir plaka tasarlanmış ve üretilmiştir (Şekil 2.3). Lazer tarayıcının ölçüm

düzleminin kendi merkezi etrafında döndüğünden emin olunması gerekir. Aksi halde yapılan ölçümlerde hatalı veriler üretilir ve çeşitli yazılımsal hatalar ortaya çıkabilir. Bu yüzden hem lazer tarayıcıyı hem de dc motoru istenen pozisyona getirebilmek için plaka L şeklinde tasarlanmıştır.

2.1.1 Enkoderli dc motor

Kullanılan dc motorun ana görevi LIDAR'ı belirlenen bir eksen etrafında döndürerek 3 boyutlu görüntü elde edilmesini sağlamaktır. Daha önce yapılan birçok çalışmada mekanik aktüatör olarak servo motor kullanılmasına rağmen sistemin çalışma hızını arttırmak amacıyla dc motor tercih edilmiştir. Motorun enkoderinden elde edilen pozisyon verileri açı verilerine çevrilerek seri haberleşme yoluyla sistemin yazılım platformu olan ROS'a gönderilmiş ve seri iletişimden (serial) okunarak bu platforma uygun hale getirilerek yayınlanmışlardır. Fakat servo motor kullanılmış olsaydı ROS platformundan gönderilen açı verilerine servonun abone olması gerekecek ve böylelikle iki ortam arası veri geçişlerinden dolayı sistem enkoderli dc motora göre daha yavaş çalışacaktı. Benzer şekilde çalışan yazıcılarda da servo motor yerine enkoderli dc motor kullanılmasının sebebi yukarıda belirtildiği gibi tarama hızını arttırmaktır.

Tasarlanan 3B LIDAR platformunda kullanılan enkoderli dc motorun özellikleri çizelge 2.2'de gösterilmiştir.

Çizelge 2.2 : Kullanılan enkoderli dc motorun özellikleri.

Parametre	Değer
Çalışma voltajı	0-12V DC
Yüksüz motor hızı	80 dev/dak
Dişli dönüştürme oranı (redüksiyon)	131:1
12V Durma akımı	5A
12V Durma torku	18 kg/cm
Enkoder çalışma voltajı	3.5V-20V
Enkoder çalışma akımı	10mA
Enkoder pulse sayısı	64 pulse/motor devri
Enkoder çıkışı	Quadrator 2 tel kare dalga
Max çap	36.8mm
Motor uzunluğu	31.7mm
Enkoder uzunluğu	15.4mm
Redüktör uzunluğu	26.5mm
Mil ve mafsalsal uzunluğu	22mm
Toplam uzunluk	94.6mm

Çizelgeye göre motorun dişli oranı 131:1 olarak belirtilmiştir. Böyle yüksek bir dönüştürme oranının seçilmesinin nedeni üzerindeki bileşenleri zorlanmadan taşıyabilmesinin istenmesidir. Bu motorlar 12V 'da kullanılmak üzere tasarlanmasına rağmen nominal gerilimin üstündeki ve altındaki gerilim değerlerinde de çalışabilirler. Bu dc motorun tercih edilmesinin bir diğer sebebi ise uygulamanın hassasiyeti için mm başına üretilen sinyal sayısının yani enkoder darbe sayısının yüksek olmasıdır.

2.1.2 Slip-ring

Lazer tarayıcı ile dc motor arasına slip-ring olarak adlandırılan güç ve elektrik sinyali iletiminde kullanılan ve hareketli bir bileşen ile sabit bir bileşen arasındaki veri bağlantısını sağlayan bir elektromekanik kayma halkası cihazı konulmuştur. Bu çalışmada Moflon firmasının MUSB2121-S02-FL01 model kayma halkası kullanılmıştır. Delik çapı 12.7 mm, dış çapı ise 56mm olan bu kayma halkasının USB1.0/USB2.0'ı destekleyen bir usb kanalı vardır. Bu kayma halkası sayesinde hiçbir kod kaybı olmaksızın sinyal eksiksiz bir şekilde iletilir. Projede bu kayma halkasının kullanılmasıyla sisteme yüksek mekanik performans, hiçbir veri kaybı olmaksızın sinyal ve güç iletimi ve dönen lazer tarayıcıdan sarkan kabloların birbirine dolanmasına engel olmak gibi özellikler kazandırılmıştır.

2.1.3 Mekanik LIDAR platformu

Bilgisayar kullanılarak yapılan deneylerde 3B Lidar sistemini taşımak için kullanılan alüminyum plaka hem çok büyük hem de taşınabilme özelliğinden yoksundu. Fakat bilgisayara göre çok daha küçük ve hafif olan Jetson TX1 geliştirme kartı kullanılmaya başlandığında bu 3B Lidar sistemine daha kompakt ve taşınabilir özellik kazandırmak için yeni bir platform tasarlanmıştır. 810 gr ağırlığında, 0.36 cm kalınlığında ve 21x22x17 cm boyutlarında olan bu platform sayesinde 3B Lidar sistemi hem kara hem de hava araçlarında kullanılabilme özelliğine sahip olmuştur. İçinde her komponent için belirlenen haznelere mevcuttur. SolidWorks programında tasarlanan bu platform, hafif, dayanıklı, kolay işlenebilme ve boyanabilme özelliklerine sahip Akrilonitril Bütadien Stiren (ABS) malzemesinden üretilmiştir. Üretilen mekanik platform şekil 2.4'de gösterilmektedir.



Şekil 2.4 : Mekanik LIDAR platformu.

2.2 Elektronik Tasarım

Elektronik tasarımda tek düzlem üzerinde tarama gerçekleştirerek nesnelerin mesafesini ölçüp açı ve uzaklık bilgisi veren 2B lazer tarayıcı, kullanılan dc motoru kontrol edip enkoder verilerini okumak ve çalışmada kullanılan yazılım platformu ile haberleşmek için mikrodenetleyici ve dc motora istenen seviyede tahrik vermek için motor sürücü kullanılmıştır. Ayrıca sistemin geliştirilme aşamasında kullanılan dizüstü bilgisayar yerine 3 boyutlu görüntü elde etmek için laserscan verilerinden dönüştürülen nokta bulutu verilerinin akış hızlarının yüksek olmasını sağlayan küçük, portatif ve donanımsal olarak güçlü bir geliştirme kiti kullanılmıştır. Sistemin güç ihtiyacı pillerden sağlanmıştır.

2.2.1 Hokuyo LIDAR

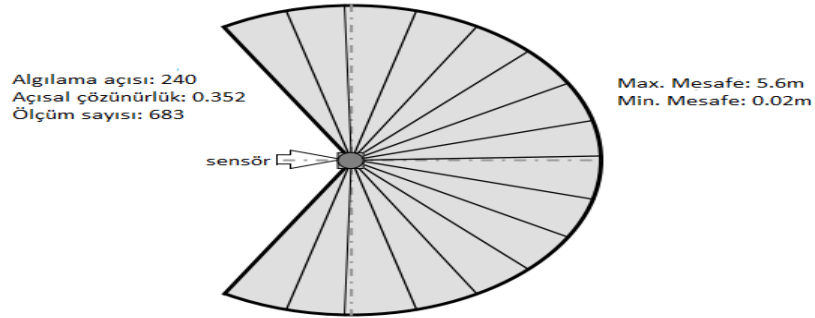
LIDAR bir nesneyi tespit etmek için lazer ışınları kullanan ve nesneden yansıyan ışınları analiz ederek nesnenin mesafesini uzaktan ölçebilen bir algılama sensörüdür. Bu tezde HOKUYO firmasının URG-04LX-UG01 model 2 boyutlu lazer tarayıcısı kullanılmıştır (Şekil 2.5) [44]. Genelde otonom mobil robotların ve otomatik malzeme taşıma sistemlerinin alan taraması ve lokalizasyonu gibi uygulamalarda kullanılan bu tarayıcı, küçük, uygun fiyatlı, kompakt ve hassas ölçüm özelliklerine

sahiptir. USB veri yolu gücünü kullanan bu cihaz akademik ve Ar-Ge çalışmaları için ideal bir algılayıcıdır.



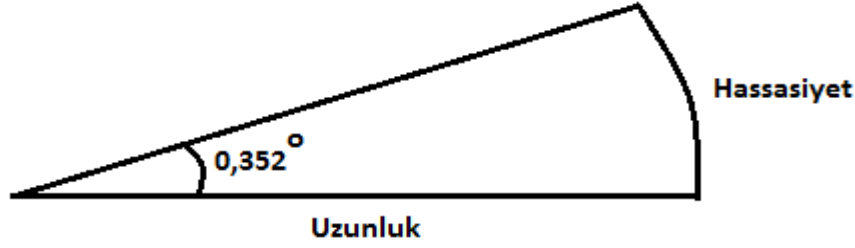
Şekil 2.5 : HOKUYO-URG-04LX-UG01 [44].

Bu lazer tarayıcı HOKUYO'nun diğer lazer tarayıcılarına göre daha düşük tarama hızı ve açısal çözünürlüğe sahiptir. 50x50x70mm boyutunda 160 gr ağırlığında olan bu tarayıcı 0.02 metreden 5,6 metreye kadar olan bir ölçüm aralığına sahiptir. 240°'lik geniş tarama aralığı ve 0,352°'lik açısal hassasiyeti sayesinde yaklaşık 683 ölçüm yapabilir. Bu ölçümler lazer tarayıcının 600 rpm'deki devir sayısı ile 100ms'de alınır. Yani her 100ms'de 0,352° açısal çözünürlükle tarama yapabilir. Şekil 2.6 ölçüm aralığının bir temsilini göstermektedir [45].



Şekil 2.6 : Hokuyo LIDAR ölçüm aralığı [45].

Gerekli tüm ölçümleri birkaç saniye içinde almak için yeterli bir veri toplama hızına (100ms) sahip olmasa da kısa mesafedeki nesnelere tespit edilmesi için uygundur. Şekil 2.7'ye göre açısal kesinlik denklemini kullanarak lazer tarayıcının doğrusal hassasiyeti tespit edilebilir. Yani belirli bir mesafenin ölçülmesi esnasında oluşan sapma miktarı elde edilebilir.



Şekil 2.7 : Hokuyo lazerin doğrusal ölçüm hassasiyeti.

$$\text{Hassasiyet} = \text{radyan değeri} * \text{uzaklık} = (2\pi * 0,352)/360 * \text{uzaklık} \quad (2.1)$$

Yukarıdaki denkleme göre 3 metre uzaklıktaki bir nesnenin sapma miktarı yaklaşık olarak 1.8cm'dir. Farklı mesafe değerleri kullanılarak oluşturulan doğrusal ölçüm hassasiyeti sonuçları çizelge 2.3'de gösterilmiştir. Lazerin minimum ölçüm mesafesininin doğrusal hassasiyeti hesaplandığında yaklaşık olarak 0.0123cm, maksimum ölçüm mesafesi ele alındığında ise doğrusal hassasiyet yaklaşık olarak 3.4 cm olarak hesaplanmaktadır. Bu değerlere göre yakın mesafedeki çok küçük nesnelere dahi yüksek doğruluk hassasiyetiyle tespit edilebilmektedir.

Çizelge 2.3 : Lazer -doğrusal hassasiyet tablosu.

Mesafe (m)	Doğrusal hassasiyet (cm)
0.02	0.0123
0.1	0.0614
1	0.61
3	1.8
5	3
5.6	3.4

Genel olarak kullanılan lazer tarayıcı iç mekân ortamında minimum hata ve doğru aralık verileriyle nesnelere tespit edebilir. Minimum güç tüketimine, yüksek tarama aralığına ve yüksek hassasiyet oranı gibi avantajlarına rağmen uzun tarama süresi, düşük tarama mesafesi ve dış mekânda kullanım eksikliği gibi dezavantajlara da sahiptir. Dış mekânda kullanılamamasının sebebi yüksek ortam ışıktan etkilenmesi ve düşük tarama mesafesine sahip olmasından dolayıdır. Bu yüzden iç mekân

algılayıcısı olarak nitelendirilir. Projemizde kullanılan otonom golf aracı iç mekân uygulamaları için tasarlandığından lazer tarayıcının dış mekânda çalışmaması sistemimiz için herhangi bir soruna sebep olmaz. Bu sensörün kullanılmasının sebebi prototip geliştirmek içindir. Gerçek dünyadaki uygulamalar için dışarıda çalışabilen, uzun tarama mesafesine sahip ve güvenlik cihazı olarak belirtilmiş bir LIDAR gerekli olacaktır.

2.2.2 Arduino Uno

Arduino, açık kaynak kodlu bir elektronik geliştirme platformudur. Birçok farklı uygulama özelliği sunan bu giriş çıkış kartı processing/wiring tabanlı bir programlama dili içerir. Farklı özelliklere sahip birçok çeşidi bulunur. Bu çalışmada Arduino modellerinden biri olan ve içerisinde Atmega328 mikrodenetleyicisi bulunan Arduino UNO kullanılmıştır. 5V çalışma voltajı ve 7-12V arasında besleme voltaj değerine sahip olan bu platformun 6 tanesi PWM pini olmak üzere toplam 14 tane digital giriş/çıkış pini bulunur. Ayrıca 6 adet analog girişi bulunmaktadır. Haberleşmesinde USB veri yolunu kullanır. Bu çalışmada kullanılan arduinonun görevi dc motor kontrolünü sağlayıp enkoder verilerini okumaktır.

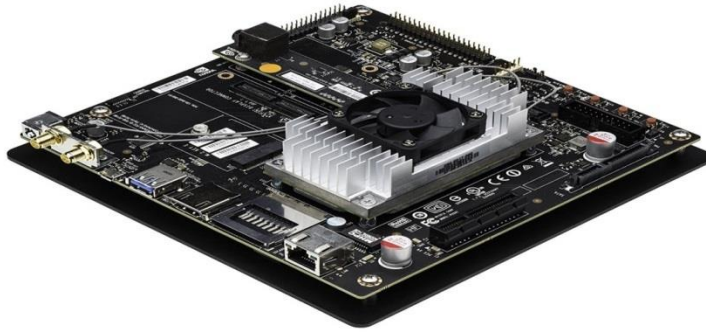
LIDAR, verilerini harici ekran sistemine göndermeden önce girerken ve bu verileri yorumlarken, denetleyici motoru kontrol ettiği için diğer sistemleri de bir mikrodenetleyici olmadan kullanmak mümkün değildir. Kısaca, motor kontrolünü sağlayıp enkoder verilerini okumak dışında LIDAR verilerini hesaplama görevini de yerine getirir.

2.2.3 L298N motor sürücü

Mikrodenetleyicilerin dc veya step motorları direk kontrol etmeleri için gereken sinyal yetersizdir. Bu yüzden mikrodenetleyicilerin çıkışlarından alınan sinyalleri yükselterek motor kontrolünü sağlayan motor sürücü entegreleri kullanılır. Bu motor sürücüleri çeşitli bağlantılar yapıldıktan sonra motorun kodlar yardımıyla kontrol etmeye yarayan entegrelerdir. Bu çalışmada kullanılan L298N motor sürücü entegresi dc motoru sürmek ve motorun hız ayarını yapmak için kullanılmıştır. Bu sürücü, yapısında iki adet H köprüsü bulundurur ve bu sayede 2 adet motor kontrol edebilir. İki kanallı olan bu motor sürücü kanal başına 2A akım verip 24V'a kadar olan motorları sürme kabiliyetine sahiptir.

2.2.4 Jetson TX1

Jetson TX1 geliştirici kiti düşük güçte yüksek hesaplama performansı gerektiren, görsel bilgi işlem hizmeti sunan ve sistemin hızlı bir şekilde çalışmasına olanak sağlayan bir geliştirme platformudur. Tasarlanan 3B lazer tarayıcı sisteminde ortamın 3 boyutlu haritasının oluşmasını sağlayan nokta bulutu verilerinin akış hızı oldukça önemlidir. Bu nokta bulutu verilerinin akış hızları ne kadar yüksek olursa o kadar net görüntü elde edilmiş olur. Bu yüzden hem yüksek güç verimliliğine sahip hem de yüksek hesaplama performansı gösteren donanımsal olarak çok güçlü bilgisayarlara ihtiyaç duyulur. Fakat yapılan araştırmalar sonucunda 3 boyutlu görüntü elde edilme aşamasında kullanılan normal dizüstü bilgisayarlar yerine en az test aşamasında kullanılan bilgisayar kadar yüksek performans sağlayan ve çok daha küçük boyutlu olması sebebiyle tasarlanan sistemi daha portatif hale getiren Jetson TX1 geliştirme platformunun kullanılmasına karar verilmiştir. Nvidia firması tarafından üretilen bu geliştirme kiti bu tip yüksek performans gerektiren birçok araştırmada kullanılmaktadır. Jetson TX1 üzerinde linux ubuntu 16.04 işletim sistemi hazır kurulmuş olarak gelir. Bilgisayar üzerinde yapılan çalışmalarda ubuntu 14.04 kullanıldığı için geliştirme platformundaki 16.04 olan işletim sistemi değiştirilerek yerine ubuntu 14.04 linux işletim sistemi kurulmuştur. Ayrıca bu geliştirme kitinin içine bütün kodsall çalışmaların yapıldığı yazılım platformu olan ROS (Robot Operating System) kurulmuştur. Fakat bu geliştirme kitine ubuntu 14.04 kurulduğunda ve kodlar bu dağıtım üzerinden çalıştırıldığında veri akış hızı ciddi şekilde düşmüştür. Bu yüzden tekrar 16.04 kurularak gerekli paketler ROS kinetic sürümüne göre güncellenmiştir. Yapılan bu değişikliklerle sistemin performansı artmıştır. Kullanılan Jetson TX1 geliştirme platformu şekil 2.8'de gösterilmiştir.

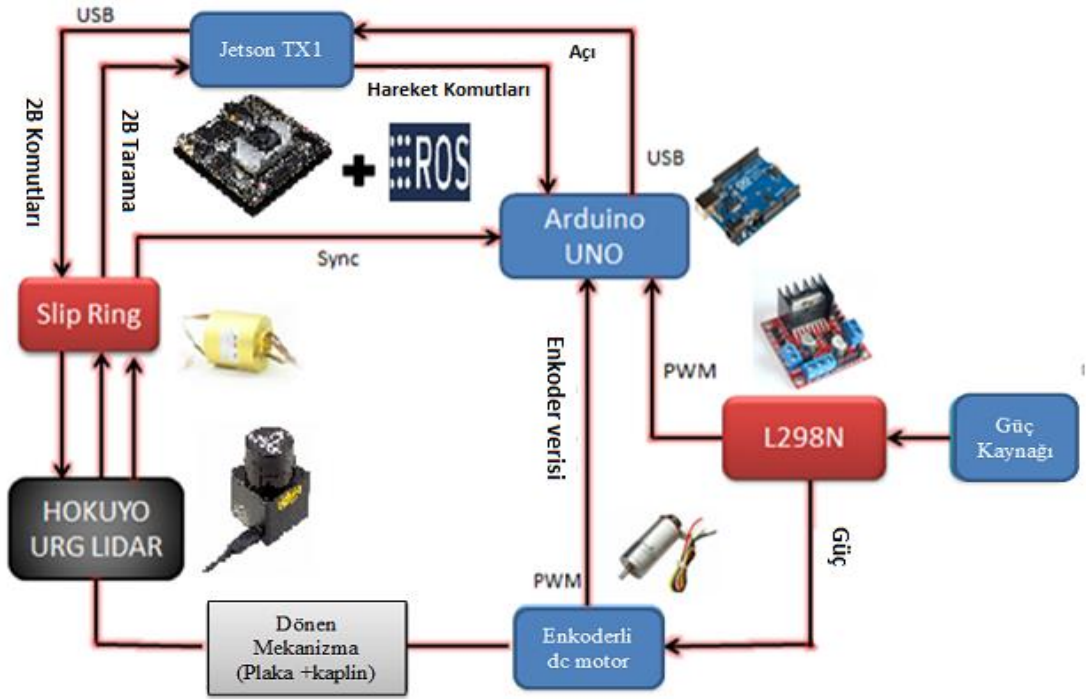


Şekil 2.8 : Jetson TX1.

2.2.5 PİL

Projede kullanılan enkoderli dc motoru beslemek için 11.1 volt 5000 mAh lityum polimer pil kullanılmıştır. Lityum polimer pil kullanılmasının sebebi NiMd ve NiMh gibi diğer pillere göre anlık daha yüksek akım üretebilmesi, daha hafif olması ve kullanım süresinin daha uzun olmasıdır. LiPo piller hücrelerden oluşur. Herbir hücrenin nominal voltajı 3.7 voltur ve bu hücreler S harfi ile gösterilir. Pil üzerinde 3S, 6S olarak gösterilen bu değerler pilin gerilim değerini ifade eder. Bu çalışmada kullanılan 11.1 voltluk pil 3S'dir.

Tasarlanan 3B lazer tarayıcı sisteminde kullanılan komponentlerin akış diyagramı şekil 2.9'da gösterilmiştir.



Şekil.2.9 : Elektronik donanımın akış diyagramı



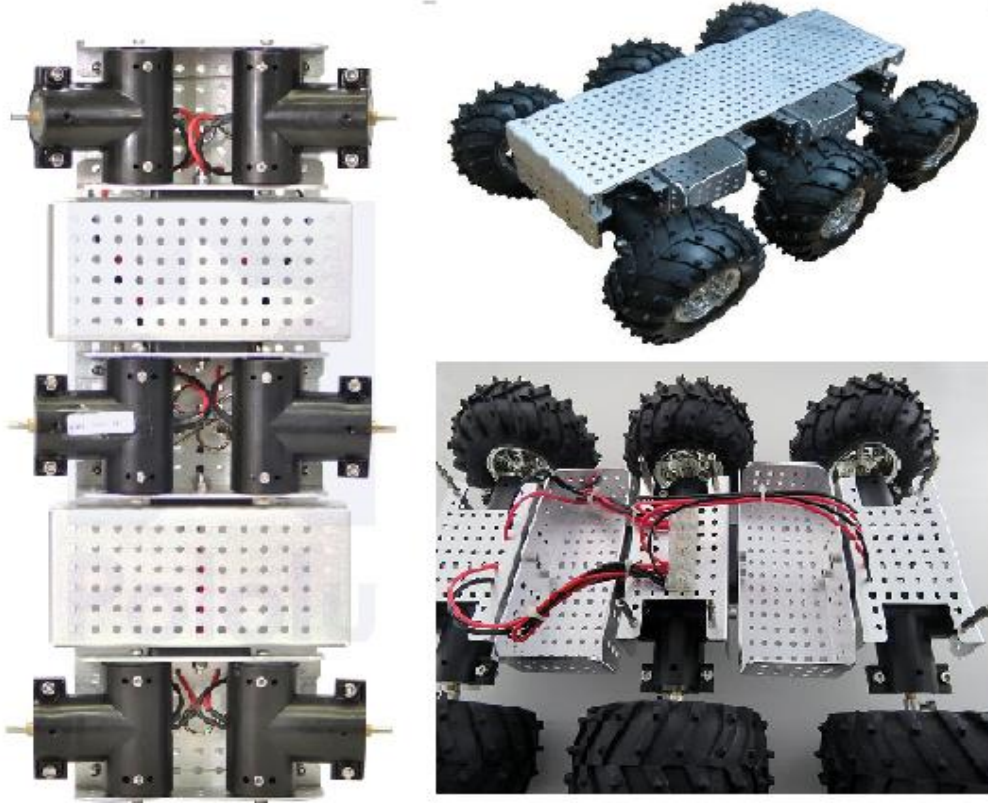
3. MOBİL ARAÇ TASARIMI

Bu tezde 3B LIDAR sistemine hareket kazandırmak ve haritalama, lokalizasyon ve engel tespiti gibi görevlerin yerine getirilmesinde kullanılmak amacıyla yarı-otonom ve otonom olarak kontrol edilebilen bir mobil araç platformu tasarlanmıştır. Araç iskeleti seçimi yapılırken maliyetinin düşük olmasına, gövde üzerinde değişiklikler yapılabilecek şekilde olmasına, üzerinde kullanılacak bileşenleri yerleştirmek için çok sayıda delik bulunmasına ve üzerine yerleştirilecek ağırlıkları rahatlıkla taşıyabilecek nitelikte olmasına dikkat edilmiştir. İstenen bu özelliklere göre DAGU firmasının 6 tekerlekli mobil araç platformunun kullanılmasına karar verilmiştir. Bu araç her biri ayrı bir motor tarafından sürülen 6 adet tekerlek ve alüminyum bir iskeletten oluşmuştur. Aracın 6 tekerlekli olmasının sebebi yük taşıma kapasitesinin artırılması ve daha dengeli bir şekilde hareketinin sağlanmasıdır. Araç, sağ üç motor ayrı sol üç motor ayrı kontrol edilmek üzere diferansiyel sürüş metoduna çok benzeyen kızaklı sürüş metodu kullanılarak hareket etmektedir. Bu metodun kullanılmasının temel sebebi tekerleklerin yön değiştirme kabiliyetinin olmamasıdır. Bu yüzden diferansiyel sürüş metoduna çok benzeyen ve tekerlekler arasındaki hız farkına bağlı olarak aracın sağa veya sola hareket etmesine dayanan kızaklı sürüş modeli kullanılmıştır. Kullanılan bu araç platformunun özellikleri çizelge 3.1'de gösterilmiştir.

Çizelge 3.1 : DAGU 6WD araç platformu özellikleri.

Parametre	Değer
Boyut	42x30x13cm
Ağırlık	2.7kg
Zeminden yükseklik	6cm
Taşıyabileceği maksimum yük miktarı	5kg
Tavsiye edilen motor gerilimi	7.2V
Yüksüz akım	420mA(motor başına)
Zorlanma akımı	6.6A(motor başına)
Yüksüz devir	160 rpm
Zorlanma torku	11 kg-cm
İskelet malzemesi	Alüminyum

Alüminyum iskelet, motor ve tekerleklerden oluşan araç platformunun görüntüsü Şekil.3.1’de gösterilmiştir.



Şekil 3.1 : DAGU araç platformu.

Araç üzerinde fırçalı doğru akım motorları kullanılmıştır ve her bir motora aracın düz olmayan bölgelerde de hareket edebilmesi için yaylı süspansiyon sistemi takılmıştır. Araçtaki motorların hiçbirinde enkoder bulunmamaktadır. Projenin ana hedeflerinden biri olan ve odometri olarak adlandırılan aracın başlangıç konumundan itibaren pozisyonunu tahmin etmek ve hızını belirlemek için enkoderlere ihtiyaç vardır. Bu enkoderler sayesinde aracın tekerleklerinin dönüş miktarına dayanan hareket kontrolü sağlanmaktadır. Bu nedenle aracın orta sağ ve sol tekerleklerindeki motorlar çıkarılarak yerine aynı özellikteki enkoderli dc motorlar entegre edilmiştir. Araç üzerinde kullanılan motorların özellikleri çizelge 3.2’de gösterilmiştir. Bu çizelgede gösterilen parametrelere göre motorun dişli oranının 75:1 seçilmesinin ana sebebi ağır yüklere karşı hareket kabiliyetinin azalmamasının istenmesidir. Motorların yüksüz durumdayken çektiği akım 420mA’dır. Fakat aracın üzerinde belli ağırlıkta yük olacağı için tasarım motorların zorlanma akımı olan 6.6A değerine göre yapılmıştır.

Çizelge 3.2 : Motor özellikleri.

Parametre	Değer
Tipi	Fırçalı dc
Çalışma voltajı	2-7.5V
Yüksüz çalışma akımı	420mA
Zorlanma akımı	6.6A
Boşta çalışma hızı	160rpm
Dişli oranı	75:1
Zorlanma torku	11 kg-cm

Mobil aracın tasarım ve kontrolünde kullanılan diğer bileşenler ve görevleri şunlardır:

3.1 BNO055 IMU

Uzayda hareket halindeki bir cismin üstünde oluşan 3 eksen ivme ve 3 eksen dönme kuvvetini ölçmek için ivmeölçer ve jiroskop olarak adlandırılan iki tip sensörden oluşan ataletsel ölçü birimi (IMU- Inertial Measurement Unit) adı verilen cihazlar kullanılır. Çalışmada kullanılan BNO055 model imu 9 serbestlik derecesine sahiptir. Bu sensörü kullanarak mutlak konum, açısal hız vektörü, ivme vektörü, doğrusal ivme vektörü gibi bilgiler elde edilebilir. Fakat bu çalışmada navigasyon işlemi için imu sensöründen sadece roll, pitch ve yaw verileri kullanılmıştır. Bu veriler USB hattını USB protokolü kullanmadan RS232 gibi seri hatta dönüştüren FTDI dönüştürücü kartı kullanılarak ROS ortamına aktarılmıştır. ROS platformunda imu/data başlığıyla yayınlanan imu verileri şekil 3.2’de gösterilmiştir.

```
ubuntu@tegra-ubuntu: ~ 80x24
header:
  seq: 9537
  stamp:
    secs: 1531387550
    nsecs: 801126956
  frame_id: "imu_link"
orientation:
  x: 16059.0
  y: 3209.0
  z: -130.0
  w: 493.0
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: 0.0
  y: 0.0
  z: 0.0011111111111111
angular_velocity_covariance: [-1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
  x: 0.007
  y: -0.006
  z: 0.006
linear_acceleration_covariance: [-1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
---
```

Şekil 3.2 : IMU verisi.

3.4 LM2596 Voltaj Düşürücü Regülatör Modülü

Regülatör farklı giriş voltajlarını belirli bir voltaj seviyesine getiren cihazlardır. Araçta kullanılan bu voltaj düşürücü regülatör kartı üzerinde anahtarlamalı gerilim regülatörü olan LM2696 entegresi bulunmaktadır ve bu kart sayesinde 4 ile 35V arasında uygulanan gerilim değerlerine göre 1.25 ile 30V arasında çıkış gerilimi elde edilir.

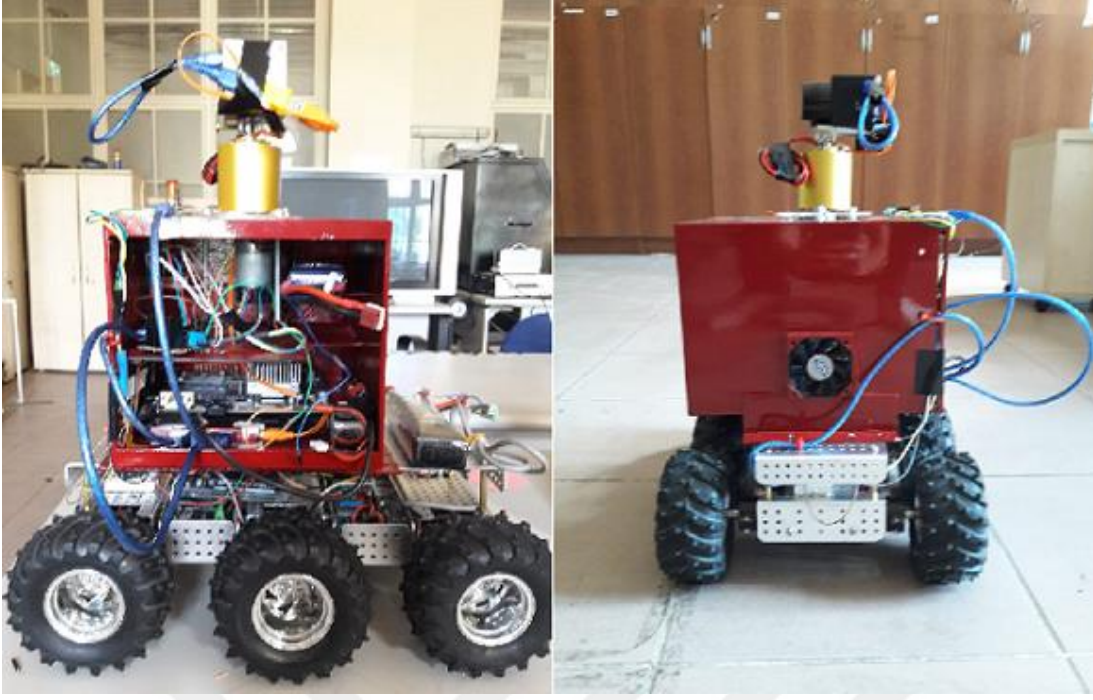
3.5 Pil

Aracın güç kaynağı olan pilin seçiminde yüksek kapasiteli olması, anlık çıkış akımının yüksek olması, pil voltajının motorların nominal voltajlarından yüksek olmaması gibi kriterler göz önünde bulundurularak 11.1V 5000mAh LiPo pil kullanılmıştır. LiPo pil kullanılmasının en önemli nedeni, LiPo pilin diğer pillere oranla kalite/ağırlık oranının daha fazla olmasıdır. Ayrıca sistemden çok yüksek akımlar çekileceği için kullanılan kablolar da bu yüksek akımlara dayanacak ölçüde seçilmiştir.

Şekil.3.4’de mobil aracın tezin amacına yönelik tasarımı, şekil 3.5’de ise 3B Lidar sisteminin mobil araca entegre edilmiş hali gösterilmektedir.



Şekil 3.4 : Mobil araç.



Şekil 3.5 : 3B Lidar sistemi ve araç entegrasyonu.

4. YAZILIM

Bu bölümde 3 boyutlu nokta bulutu verilerinin elde edilme aşamaları, 2B lazer tarayıcı kullanılarak 3 boyutlu LIDAR elde edilmesi aşamasında kullanılan yazılım platformu, 3B haritalama, navigasyon ve belirtilen bu konular üzerinde yapılan deneyler hakkında bilgi verilecektir.

4.1 Nokta Bulutu ve Geometrik Dönüşümler

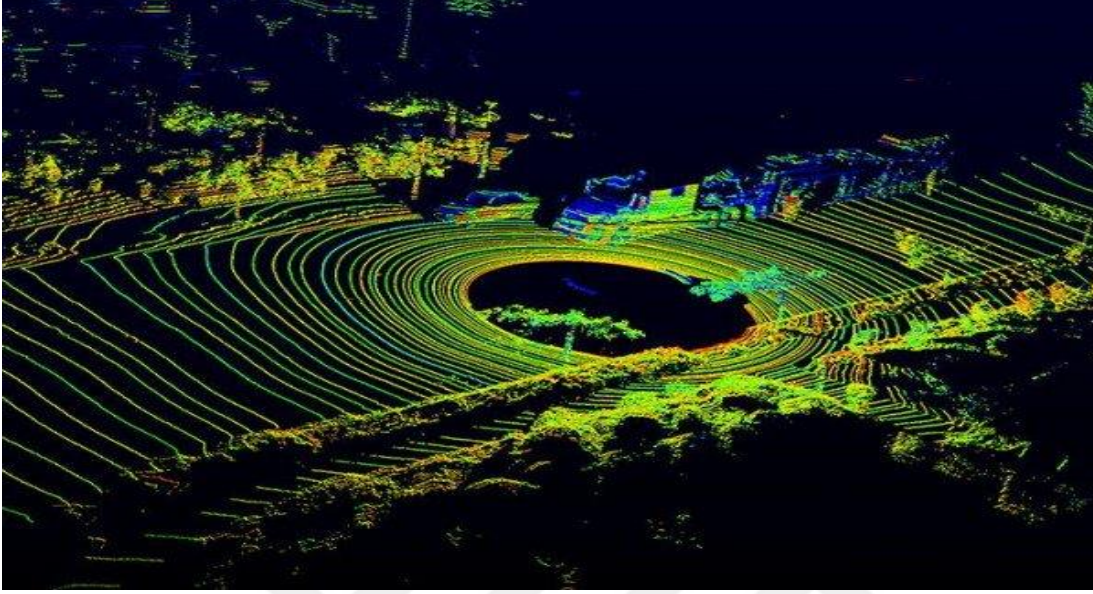
Lazer tarayıcılar kullanılarak 3 boyutlu görüntü elde edilmesinin temeli engel yüzeyine gönderilen ışınların nokta bulutu verilerine dönüştürülmesine dayanmaktadır. Çünkü bu noktalara karşılık gelen mesafeler 3 boyutlu bir harita oluşturulmasını sağlar. Nokta bulutu verileri ve bu verilerin elde edilmesini sağlayan geometrik dönüşümler 2 boyutlu lazer tarayıcıdan 3 boyutlu görüntüye geçiş aşamalarının temelini oluşturmaktadır.

4.1.1 Nokta bulutu

Nokta bulutları, genellikle çevrelerindeki nesnelerin dış yüzeylerindeki noktaları ölçen tarayıcılar, RGB-D kameralar, stereo kameralar tarafından üretilen 3 boyutlu koordinat sistemindeki bir dizi veri kümeleridir. Bir nesnenin dış yüzeyinin temsili olarak da ifade edilir. Bu nokta bulutları sayesinde bir nesnenin geometrisi ve o nesnenin ortamdaki diğer nesnelerle arasındaki çevresel ilişki hakkında bilgi edinilir.

Aynı zamanda her nokta için elde edilen mesafe verisinin doğruluğunun test edilmesinde ve tespit edilmek istenen nesne yüzeyinin özelliğinin tespitinde kullanılır. 3D tarama işlemlerinin bir çıktısı olan bu nokta bulutları, üretilen parçaların 3D modellerinin elde edilmesi, animasyon, görselleştirme gibi birçok amacın gerçekleştirilmesinde kullanılır. Otonom araçlar ve robotik alanındaki çalışmalarda ortamın haritasının çıkarılmasına çok hayati bir katkı sağlar.

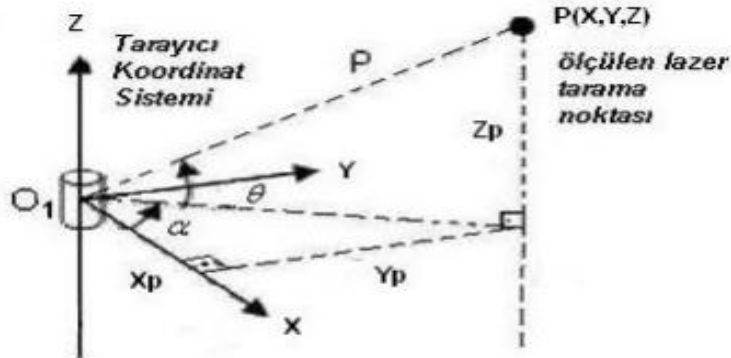
Şekil 4.1’de otonom araç üzerinde konumlanan bir velodyne LIDAR tarafından üretilen nokta bulutu verileri gösterilmektedir [46].



Şekil 4.1 : Velodyne LIDAR kullanılarak üretilen nokta bulutu [46].

4.1.2 Geometrik dönüşümler

Lazer tarayıcılar tespit edilecek nesnenin yüzeyini saniyede on binlerce noktanın X,Y,Z kartezyen koordinatlarını elde edecek şekilde belli bir açı altında yatay ve düşey yönde nokta dizileri şeklinde tararlar ve nesnenin nokta bulutu halinde görünmesini sağlarlar. Fakat düzgün bir nokta bulutu verisi elde edebilmek için tarayıcıdan elde edilen ve farklı yönlerde konumlanan noktaların tek bir koordinat sisteminde toplanmaları gerekmektedir.



Şekil 4.2 : Nokta bulutu dönüşümleri [47].

Aynı zamanda Bir XYZ bulutu elde etmek için lazer tarayıcı ve kullanılan motor tarafından elde edilen verilerin dönüştürülmesi gerekir. Şekil 4.2’de lazer nokta yakalamanın geometrik temsili gösterilmektedir. Bu şekle göre lazer tarayıcının konumlandırıldığı nokta başlangıç noktası olarak kabul edilir [47].

Şekilde gösterilen ρ değeri lazer tarayıcı ile ölçülen nesne arasındaki eğik mesafeyi temsil etmektedir. Ölçüm doğrusunun x eksenini ile yatay düzlemde yaptığı açı α , ölçüm doğrusunun yatay düzlemle yaptığı eğim açısı da θ olarak ifade edilir. Tez çalışmasında kullanılan hokuyo lazer tarayıcı 683 ölçüm değeri sunar. Bu yüzden kullanılan bu lazer tarayıcı ile ölçülen nesne arasında bu eğik mesafe değerlerini temsil eden tam 683 ölçüm değeri sağlanır. Aynı zamanda kullanılan motorun açı değeri ve kullanılan lazerin çözünürlüğü $0,352^0$ olduğu için bu lazer açı değeri de bilinmektedir. Motor bir nokta bulutunu yakaladığında lazer ölçümleri motor dönerken sürekli olarak kaydedilir. Bu şekilde 683 lazer ölçümü bir düzleme karşılık geldiğinde ilgili motor açısı da kaydedilir. Motor dönmeyi bitirdiğinde elde edilen verileri kullanarak dönüşümler yapılır. Elde edilen bu veriler arasındaki geometrik ilişki ve taranan noktaların kullanılan lazer tarayıcı merkezli XYZ koordinatları aşağıdaki denklemlerle ifade edilmektedir.

$$X=\rho.\cos(\theta).\cos(\alpha) \quad (4.1)$$

$$Y=\rho.\cos(\theta).\sin(\alpha) \quad (4.2)$$

$$Z=\sin(\theta) \quad (4.3)$$

4.2 ROS

Robot işletim sistemi olarak adlandırılan ROS, Willow Garage tarafından geliştirilen ve robot bileşenlerinin bilgisayar üzerinden kontrolünü sağlayan açık kaynak kodlu, geliştirilmeye açık ve ücretsiz bir yazılım platformudur. Yapısında yazılım geliştiricilerin ve robotik çalışmalar yürüten araştırmacıların yaptıkları robotik uygulamaların geliştirilmesine yardımcı olan çeşitli kütüphaneler ve araçlar bulunur. Gerçek bir işletim sistemi olmamasına rağmen normal işletim sistemine benzer bir işlevsellik sunar. Genellikle linux tabanlı bir işletim sisteminde çalışmaktadır. Linux dağıtımlarına (Ubuntu, Debian, Pardus) benzer şekilde ROS’un da çeşitli dağıtımları (box turtle, c turtle, diamondback, electric, fuerte, groovy, hydro, indigo, jade, kinetic) mevcuttur. Bunlardan ilk yayınlanan dağıtım box turtle, son yayınlanan

dağıtım da kinetic'tir. Bu tez çalışmasında linux işletim sisteminin ubuntu dağıtımı, robot işletim sisteminin de indigo dağıtımı kullanılmıştır. ROS, nesne tanımlama, yer tespiti, haritalama, navigasyon, nesne tanıma, nesne takip etme, robot ağları, nokta bulutu oluşturma, görüntü saptama gibi uygulama alanlarında yaygın olarak kullanılabilir. kullanılabilmektedir.

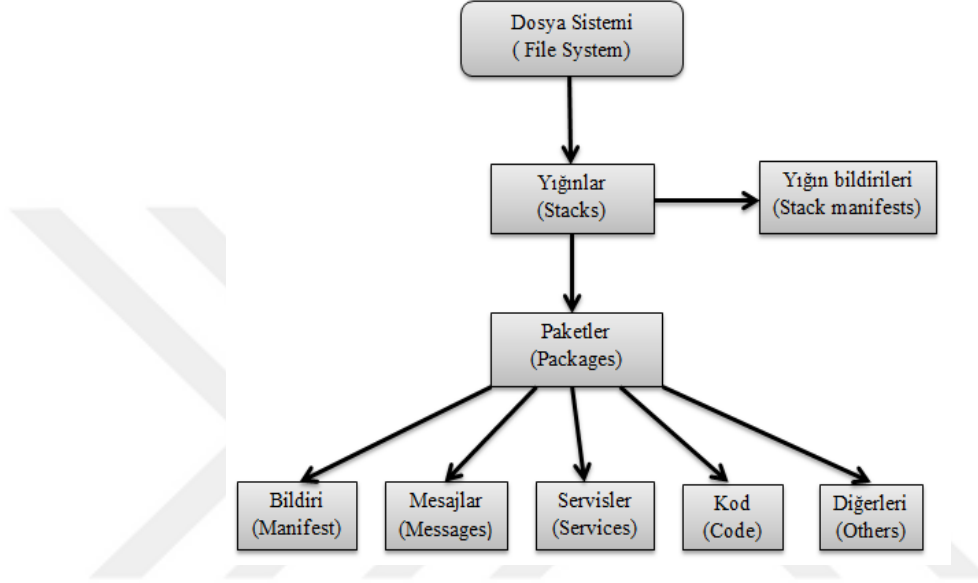
Robot işletim sisteminin amacı robot ile programcı arasındaki ilişkiyi sağlamaktır ve bu sistem çeşitli gelişmiş özellikleri sayesinde programcıların çok kolay ve rahat bir şekilde çalışmasına katkı sağlar. Robot işletim sisteminin kullanıcılarının kolay ve rahat bir şekilde çalışmasını sağlayan özellikleri aşağıda verilmiştir:

- Bir robot için yazılan kod çok az bir değişiklikle farklı bir robotta kullanılabilir.
- Robotta belli bir işlevi yerine getirmek için kullanılan bir algoritma basit birkaç işlemle değiştirilerek robot işletim sistemine uygun hale getirilebilir.
- Python, C++, Java, Lisp, Lua gibi popüler programlama dilleriyle uyumludur.
- Farklı dillerde yazılmış kodlar birlikte çalıştırılabilir. Yani C++ diliyle yazılan bir kod ile Python ile yazılan bir kod aynı anda hiçbir değişiklik yapılmadan çalıştırılabilir.
- Her işlem farklı başlıklar (topics) vasıtasıyla taşındığı için yazılımsal hataların tespit edilmesi daha kolaydır.
- Gazebo (robot fizik simulatörü) ve Rviz (görselleştirici paneli) platformlarıyla uyumlu olduğu için yapılan çalışmalar simülasyon ve görselleştirme ortamında test edilebilir.
- Çok sayıda module sahip olan ve bu modüllerin aynı anda yürütülmesi gereken büyük sistemler için oldukça uygundur.

ROS'un yapısı dosya sistemi ve hesaplama grafiği sistemi olarak ikiye ayrılır. Dosya sistemi, robot işletim sisteminin dâhili olarak nasıl oluştuğu, klasör yapısının nasıl olduğu ve çalışması için gereken minimum dosyaların ne olduğuyla alakalı bilgi verirken, hesaplama grafiği sistemi ise robot işletim sisteminin tüm işlemleri ele alması, tek bir bilgisardan daha fazlasıyla iletişim kurması ve sistemin birbirleriyle haberleşmesi hakkında bilgi verir.

4.2.1 Dosya sistemi

Robot işletim sistemi çeşitli klasörlere ayrılmıştır ve bu klasörlerin görevlerini tanımlayan bazı dosyalar vardır. ROS'un dosya sistemi paketler (packages), yığınlar (stacks), yığın bildirileri (stacks manifest), bildiriler (manifest), mesajlar (messages), servisler (services) ve kodlardan oluşur. ROS'un dosya sisteminin yapısı şekil 4.3'de gösterilmiştir.



Şekil 4.3 : Dosya sistemi grafiği.

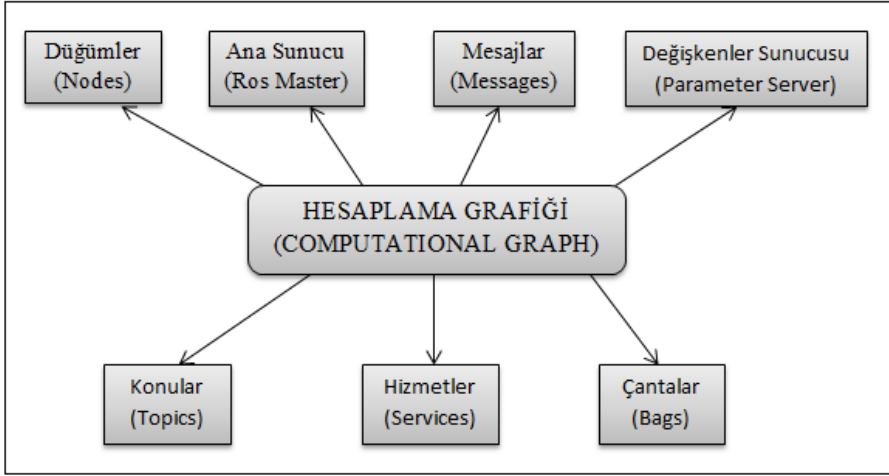
4.2.1.1 Paketler (Packages) ve Yığınlar (Stacks): Robot işletim sistemi yazılım organizasyonunun en temel birimini oluştururlar ve genel olarak robot işletim sisteminde derlenen düğümler, kütüphaneler ve veri kümeleri gibi hesaplama grafiklerini içinde bulunduran klasörlerdir. Bir paket ROS içinde bir program oluşturmak için gereken minimum içeriğe sahiptir. Yığınlar ise birçok paketin birleşmesiyle oluşan paket topluluklarıdır. Her yığın farklı görevleri yerine getirmek için kullanılır. Birlikte geliştirilen ve karşılıklı olarak birbirine bağlı olan kodları birleştirmeyi ve kod paylaşım sürecini basitleştirmeyi amaçlar. Örnek olarak navigasyon yığını bir konum tespiti paketinden, bir üst düzey robot işletim sistemi düğümünden ve birçok farklı planlayıcı paketlerinden oluşur.

4.2.1.2 Bildiriler (Manifests): Paketlerin ve yığınların içinde bulunan ve bu paket ve yığınların bağımlılıkları, lisans bilgileri ve yaptığı işler hakkında bilgi veren dosyalardır. Bir klasörün içinde bulunan manifest.xml dosyası o klasörün bir paket, stack.xml dosyası ise o klasörün bir yığın olduğunun göstergesidir.

4.2.1.3 Mesajlar (Messages) ve Servisler (Services): Bir mesaj, bir düğümün diğer düğümlere gönderdiği bilgidir. Dosya sisteminde mesajların veri yapısının tanımlandığı dosya olarak adlandırılır. ROS'un mevcut durumda önceden tanımlanmış birçok mesaj türü vardır. Servisler ise iki düğümün talep (request) ve cevap (response) şeklinde anlaşmasında kullanılan ve bu veri yapılarını tanımlayan bir sistemdir.

4.2.2 Hesaplama grafiği

ROS tüm işlemlerin birbirleriyle iletişim halinde olduğu bir ağ oluşturur ve bu ağ içerisindeki herhangi bir düğüm diğer düğümlerle etkileşime girebilir, veri alışverişi yapabilir ve gönderilen verileri görebilir. Kısaca aynı anda çalışan tüm işlemlerin eş düzeyli bir ağıdır. Şekil 4.4'de gösterilen bu hesaplama grafiğini düğümler (nodes), servisler (services), mesajlar (messages), başlıklar (topics), değişkenler sunucusu (parameter server), ana sunucu (ros master) ve verileri kaydedip başka platformlarda görüntülenmesini sağlayan çanta (bag) dosyaları oluşturur.



Şekil 4.4 : ROS hesaplama grafiği.

4.2.2.1 Ana sunucu (master) ve değişken sunucusu (parameter server): ROS'un ana düğümüdür bu yüzden ROS ana sunucu ile başlar. Bu sunucu ağdaki bilgileri depolayan ve diğer ROS düğümlerinin birbirlerini bulup haberleşmesine izin veren bir programdır. Bu ROS düğümleri birbirlerini bulduğunda birbirleriyle eşler arası iletişim kurarlar. Eğer ana sunucu olmazsa hiçbir düğüm diğer düğümlerle iletişim kuramaz. Ana sunucu düğümler arasındaki bağlanma isteklerini TCP/IP protokolünü kullanarak yollar. Ana sunucu roscore komutu kullanılarak aktif hale getirilir. Ana sunucunun bir parçası olan değişken sunucusu ise verileri merkezi bir şekilde

depolamayı sağlar ve deęişkenlerin alıřmaya bařlamadan nce kendilerini nasıl yapılandırmaları gerektięini gsterir.

4.2.2.2 Dęm (Node): Dęmler robot iřletim sisteminde hesaplamayı gerekleřtiren derlenmiř kaynak dosyalarıdır. ROS yayıncı/abone mesajlařma modelini kullanarak dięer dęmlerle iletiřim kuran dęmlerden oluřur. Bu dęmler grafik halinde bir araya getirilir ve deęiřken sunucusu, konular ve hizmetler vasıtasıyla birbirleriyle iletiřim kurarlar. Bir sistem birok dęmden oluřur ve her dęm bir grevin yerine getirilmesinden sorumludur.

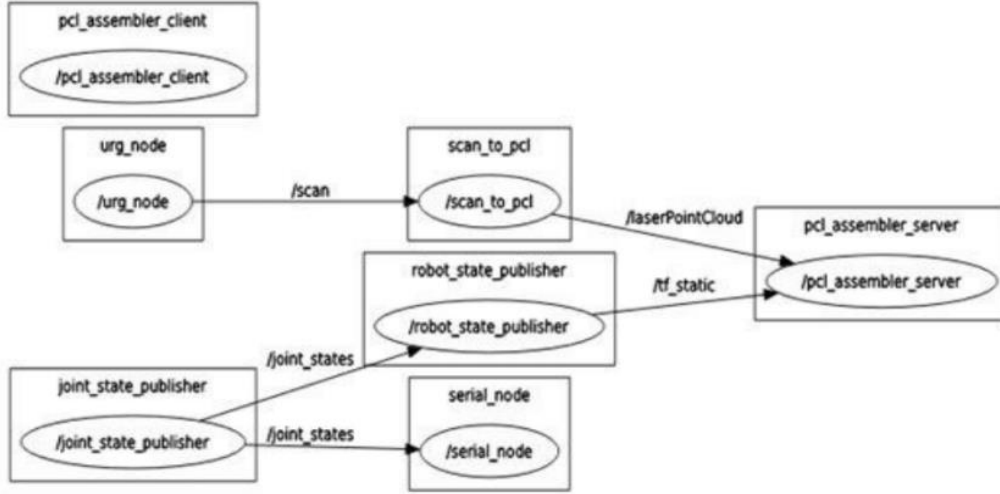
4.2.2.3 Bařlık (Topic): Dęmler, bařlıklar (topics) tarafından yayınlanan mesajlar vasıtasıyla birbirleriyle iletiřim kurarlar. Bařlıklar bir dęmden dięerine mesaj tařıyan otobsler olarak adlandırılırlar. Bir dęm bir bařlık ile mesaj gnderdięinde bu dęmn bir bařlık yayınladıęı anlamına gelir. Benzer řekilde bir dęm bir bařlık aracılıęıyla mesaj aldıęında bu dęmn bir bařlıęa abone olduęu anlamına gelir. Dęmler aynı anda birden fazla bařlık yayınlatabilir ve birden fazla bařlıęa abone olabilirler. Her bařlıęın dęmler arasında karıřıklık yaratmaması iin ayırt edici bir ismi vardır.

4.2.2.4 Mesaj (Message) ve Servis (Service): Dęmler, mesajlar vasıtasıyla birbirleriyle iletiřim kurarlar. Bu mesajlařma sistemi servislerden farklı olarak bir dęmn bir bařlık yayınlaması ve dięer dęmn o bařlıęa abone olması prensibine dayanır. Bir mesaj dięer dęmlere bilgi gnderen veriler ierir. Mesaj yapıları C programlama dilinin veri yapılarına benzer dizilerden oluřmaktadır. ROS tarafından standartlařmıř birok mesaj tipi vardır. Fakat kullanıcı isterse bu standartlařmıř mesaj tiplerini kullanmak yerine kendi mesaj tipini de oluřturabilir. Dęmler yayınlama/abone olma iletiřim trnn yanısıra bir servis aęırısının istek/yanıt letisi olarak da birbirleriyle iletiřim kurabilir.

4.2.2.5 anta (Bag): antalar, ROS mesaj verilerini kaydetmek ve bařka bir platformda oynatmak iin kullanılan bir formattır. Bir anta dosyasının iine birden ok bařlık kaydedilebilir ve uygulama esnasında kaydedilen bu veriler daha sonra bařka bir bilgisayar zerinde oynatılabilir.

ROS, rqt_graph olarak adlandırılan, ROS hesaplama grafięini grselleřtirmek iin bir GUI eklentisi sunan ve dęmler ile bařlıklar arasındaki iliřkiyi gsteren bir eklentiye sahiptir. řekil 4.5'de 2B lazer tarayıcıdan 3 boyutlu nokta bulutu

verilerinin elde edilmesini sağlayan düğüm-başlık iletişim grafiği gösterilmiştir. Dikdörtgenler düğümleri, yaylar ise başlıkları temsil etmektedir. Şekilde görüldüğü üzere bir düğüm birden fazla başlık yayınlayabilir ve birden fazla başlığa abone olabilir.



Şekil 4.5 : Düğüm-Başlık iletişim grafiği.

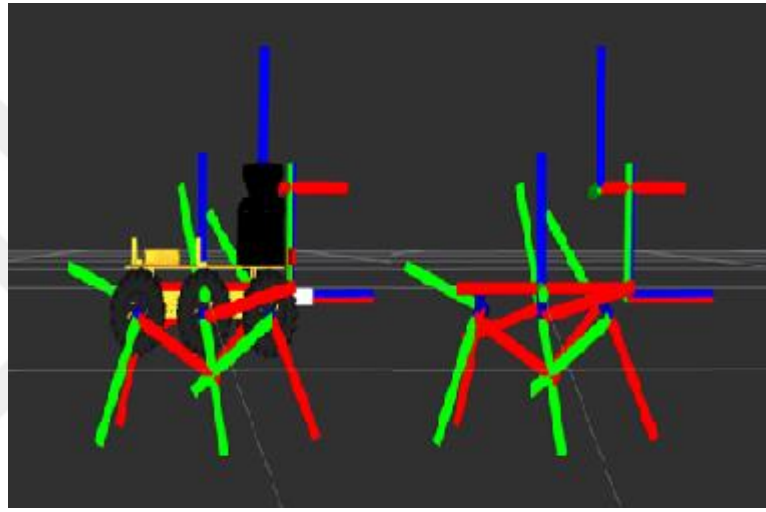
4.2.3 URDF (Universal Robot Description File) paketi

Robot tanım dosya yapısı olarak da ifade edilen bu paket, robotu oluşturan bütün altyapıları içinde barındıran .xml dosya formatına sahip bir modeldir. Robotu oluşturan bağlantıların ne kadar uzun olduğu ve bu bağlantıların pozisyon yönelimleri, eklemlerin nasıl hareket edeceği ve bu eklemlerin dönme eksenleri ve sınırları, aktuatör ve algılayıcıların pozisyon ve yönelimleri gibi birçok konuda bilgi sağlar. Ayrıca robotu oluşturan atalet, kütle gibi fiziksel özellikler hakkında bilgiye de bu model sayesinde ulaşılabilir. Kısaca bu model sistemi değiştirmeyen herşeyi açıklar. Uzantısı .urdf olan bu robot tanım dosyası robot işletim sistemi kullanan her robotta bulunur. Bu model, ROS'un gazebo simülasyon ve Rviz görselleştirici platformunda görüntülenebilir.

4.2.4 Dönüşüm çerçevesi (transform frame) paketi

Her robot belli bir koordinat çerçevesine sahiptir. Bu koordinat çerçevelerinin sayısı robotların karmaşık yapılarının düzeyine göre değişir. Kompleks robotların koordinat çerçeveleri fazla basit robotların ise daha azdır. Robota ait bir koordinat çerçevesinin diğerine göre nerede bulunduğu bilinmesi, robotun istenilen görevi hatasız bir şekilde yerine getirmesinde, hareket planlama ve engellerden sakınmasında sorun

yaşamaması için oldukça önemlidir. Bu nedenle ROS’da her bir robot bağlantısının, aktuator ve algılayıcıların merkezi koordinat çerçevesine göre hangi zamanda nerede olduğunu bildiren ve modülerliği sayesinde her robota uygulanabilen “**tf**” adında bir paket bulunmaktadır. Bu paket robot işletim sistemi kullanan her robotta bulunur. Eğer robotun bağlantıları ve eklemleri hareket etmiyorsa tf statiktir. Fakat her robotun bağlantıları ve eklemleri genellikle hareket ettiğinden çoğu zaman tf dinamiktir. Buna rağmen URDF daima statiktir. Şekil 4.6’da, tasarlanan mobil aracımız ve aracın koordinat çerçevelerinin Rviz platformundaki görüntüsü gösterilmiştir.



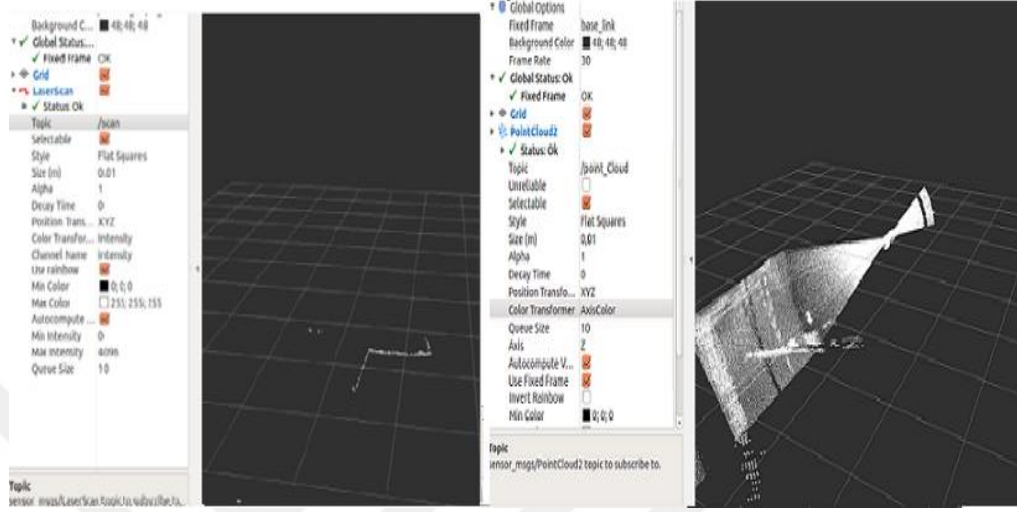
Şekil 4.6: Mobil araç dönüşüm çerçevesi (TF).

4.3 3 boyutlu nokta bulutu görüntüsünün elde edilme aşamaları

ROS yazılım platformu üzerinde 2 boyutlu veri yayınlayan lazer tarayıcıdan 3 boyutlu nokta bulutu görüntüsünün elde edilmesi sürecinde sırasıyla aşağıdaki işlemler uygulanmıştır.

- Tarama için eksen ayarlaması yapıldı. 3 boyutlu görüntü elde edilebilmesi için öncelikle tarayıcının belli bir eksen etrafında dönmesi gerekmektedir. Bu dönme ekseninin tarayıcının ölçüm noktasında uygulanması amacıyla 4 temel ölçüm metodu kullanılmaktadır. Pitching scan, rolling scan, yawing scan ve yawing scan top olarak adlandırılan bu tarama metotlarından her biri detaylı olarak incelenerek tasarlanan platformda gerçek zamanlı kullanım için en uygun metodun yawing scan top metodu olduğuna karar verildi (Şekil 4.7) [48].

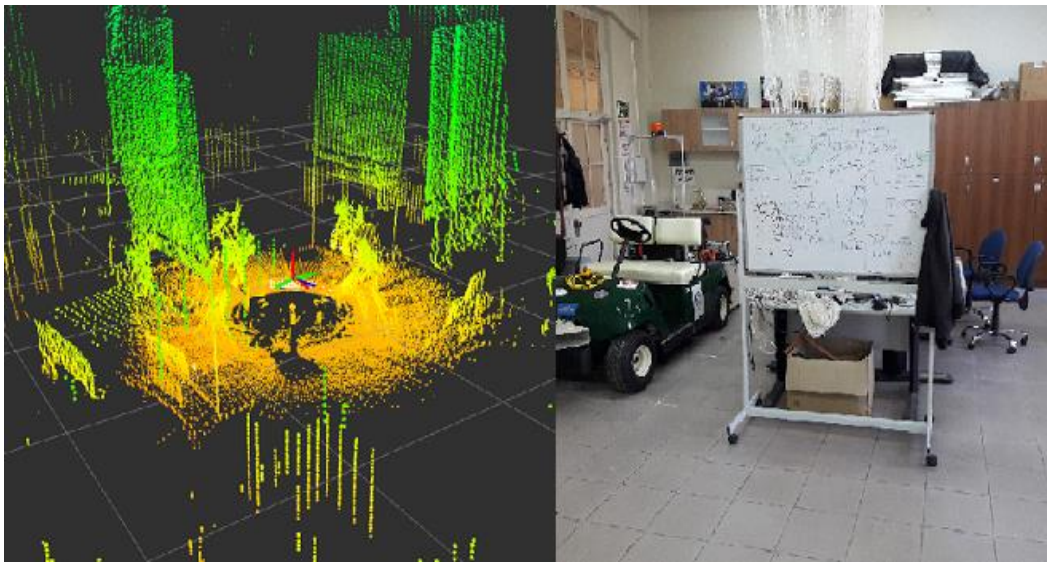
Şekil 4.8’de /scan başlığındaki LaserScan mesafe verilerinin ve /Point_Cloud başlığındaki PointCloud2 nokta bulutu verileri gösterilmiştir. Şekil 4.9’da ise lazer tarama verisinin nokta bulutu verisine dönüştürüldüğünde Rviz platformunda elde edilen görüntü gösterilmiştir.



Şekil 4.9 : Mesafe ve nokta bulutu verilerinin rviz’deki görüntüleri.

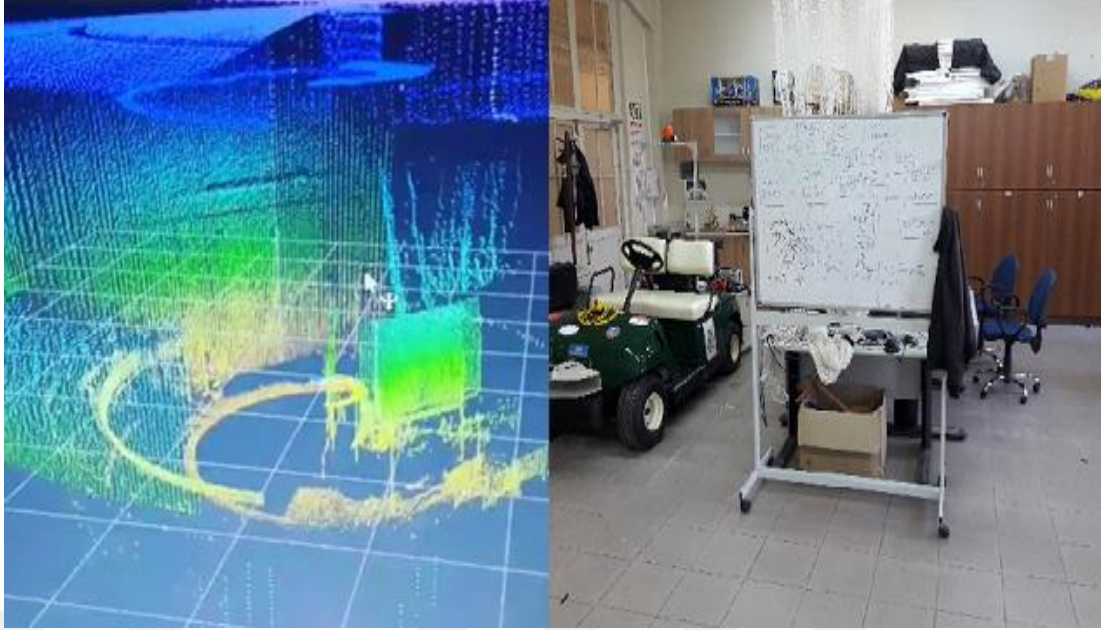
- URDF robot modeli ve TF dönüşüm çerçeveleri oluşturuldu. Bu sayede lazer tarayıcının tarayacağı eksen robot işletim sistemine aktarıldı ve base_link olarak ifade edilen lazer tarayıcının üzerinde bulunduğu sabit platform ile laser_link olarak ifade edilen lazer tarayıcı arasındaki bağlantı ilişkisi oluşturuldu.
- ROS-Arduino haberleşmesi sağlandı. Temelde lazer tarayıcının belirlenen eksen etrafında hareket etmesini sağlayan enkoderli dc motorun kontrol görevini yerine getiren arduinonun görevi bununla sınırlı değildir. Motor kontrolünü sağlayıp enkoder verilerini okumanın yanı sıra lazer tarayıcıdan yayımlanan verileri hesaplamakla da görevlidir. Bu yüzden bu mikrodenetleyici ile robot işletim sisteminin haberleşmesi gerekir.
- Farklı konumlardaki birden fazla lazer tarama verisi tek bir başlık altında nokta bulutu verisi olarak yayımlandı. Bu adımla birlikte 2 boyutlu lazer tarayıcıdan 3 boyutlu görüntü elde edilmiş oldu. Fakat servo motor yerine enkoderli dc motor kullanılması bu görüntünün anlamsız sonuçlar vermesine sebep oldu.

- ROS-Enkoder senkronizasyonu sağlandı. ROS platformunda hazır kütüphanesi bulunan servo motor yerine sistemin daha hızlı çalışmasını sağlamak amacıyla enkoderli dc motor kullanılmıştır. Servo motor kullanıldığında motorun robot işletim sisteminden yayınlanan açı verisine abone olmasıyla ROS-Servo haberleşmesi sağlanmaktadır. Aynı yöntem enkoderli dc motor için uygulandığında ise robot işletim sisteminden yayınlanan açı ile enkoder açısı arasındaki frekans farklılıklarından dolayı iki açı eşleşmemiştir. Bu yüzden lazer tarayıcının taradığı alanda bulunan bir tane nesne birden fazla nesne olarak görüntülenmiştir.



Şekil 4.10 : ROS-Enkoderli dc motor senkronizasyon sorunu.

Şekil 4.10'da tarama alanında bulunan bir tane tahtanın dört tane olarak görüntülediği gösterilmiştir. Enkoderden gelen açı ile robot işletim sisteminden gelen açının eşleşmemesinden dolayı ROS-Servo haberleşmesinde kullanılan metottan farklı bir metod denenmiştir. Bu metoda göre enkoderden elde edilen açı verileri seri haberleşme yolu ile robot işletim sistemi platformuna gönderilmiş ve gönderilen bu veri seri iletişimden okunduktan sonra robot işletim sistemi platformuna uygun hale getirilerek yayın yapılmıştır. Böylelikle hem ROS-Enkoder haberleşmesi sağlanmış hem de istenilen ölçüde 3 boyutlu görüntü elde edilmiştir (Şekil 4.11).



Şekil 4.11: Kapalı bir ortamın 3 boyutlu görüntüsü.

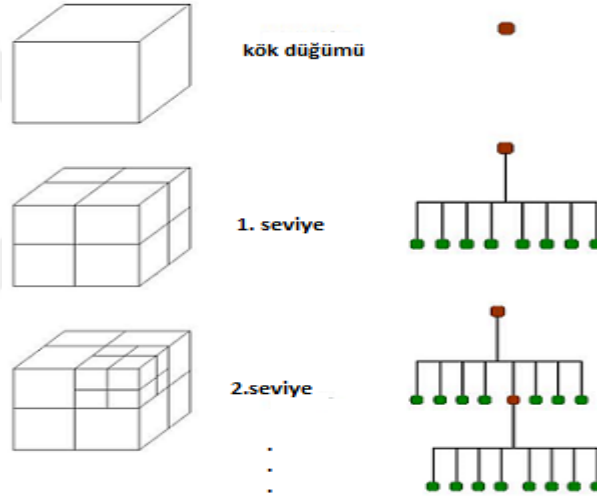
4.4 3B Haritalama (OctoMap)

2B lazer tarayıcıdan 3B nokta bulutu görüntüsü oluşturulduktan sonra mobil aracın bulunduğu ortamın 3 boyutlu haritası oluşturuldu. Nokta bulutlarını 3 boyutlu doluluk haritalarına dönüştürmek için kullanılan OctoMap, octree olarak adlandırılan harita sıkıştırma yöntemine dayanan, diğer haritalama metotlarının aksine sadece dolu ortamı değil boş ve bilinmeyen ortamı da temsil edebilen, olasılıksal doluluk tahmini kullanan, esnek ve kompakt bir 3B haritalama metodudur. Sadece nokta bulutları kullanılarak da 3 boyutlu haritalama yapılabilirdi fakat nokta bulutları büyük miktarda ölçüm noktası saklar ve bundan dolayı çok yüksek bir bellek tüketimine sebep olur. Bellek tüketimi 3 boyutlu haritalama işlemlerinde büyük bir sorundur. Bellek tüketiminin fazla olması gerçek zamanlı çalışan bu sistemlerde ciddi yavaşlamalara sebep olur. Ayrıca, engelsiz ve kullanılmayan alanlar arasında ayırım yapmaya izin vermez ve olasılıksal olarak çoklu ölçümlerin kaynaştırılması için hiçbir yol sağlamaz. Bu yüzden, bellek tüketimini minimumda tutma, boş ve dolu alanın verimli ve olasılıksal güncellemelerine olanak sağlama, verileri tutarlı bir şekilde modelleyebilme, açık kaynaklı bağımsız bir C++ kütüphanesi şeklinde kullanılabilme gibi avantajlarından dolayı bu haritalama metodu kullanılmıştır. OctoMap, Robot İşletim Sistemi (ROS) platformuna entegre edilebildiğinden bu platform üzerinde kolaylıkla kullanılmıştır. ROS platformunda pkg-config

yardımla veya cmake yapı sistemindeki find-pkg mekanizması ile derlenerek başka bir çerçeveye kolayca entegre edilebilir. Bu çalışmada kullanılan 2B lazer tarayıcı haricinde her türlü mesafe sensörü kullanılarak OctoMap ile 3 boyutlu haritalama işlemi gerçekleştirilebilir.

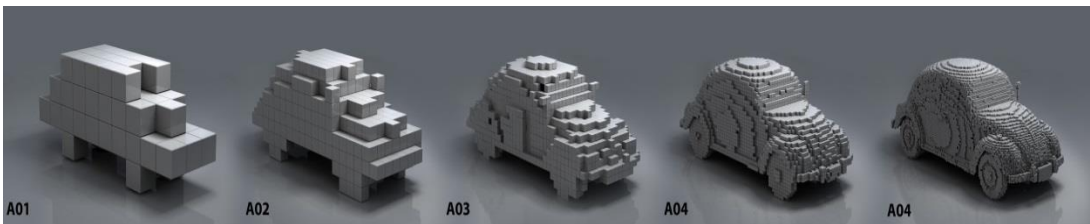
4.4.1 Octree

OctoMap'a bellek tüketimini en aza indirme özelliği kazandıran Octree, 3 boyutlu konumsal altbölüm için hiyerarşik bir veri yapısıdır. Octree'deki her düğüm genellikle voxel olarak adlandırılan kübik bir hacimde bulunan alanı temsil eder. Bu hacim şekil 4.12'de gösterildiği gibi belirli bir voxel boyutuna ulaşına kadar yinelemeli olarak sekiz alt hacime bölünür. Şekle göre soldaki hacimsel gösterimi, sağdaki ise ağaç yapısı gösterimini temsil etmektedir [49].



Şekil 4.12: Octree modelinin yapısı [49].

Minimum voxel boyutu octree'nin çözünürlüğünü belirler. Bir octree hiyerarşik bir veri yapısı olduğundan, iç düğümler buna uygun olarak muhafaza edilirse, ağaç daha iri bir alt bölüm elde etmek için herhangi bir seviyede kesilebilir. Bu sayede aynı haritanın farklı çözünürlüklerdeki görüntüleri elde edilebilir. Şekil 4.13'de bir parçanın farklı voxel çözünürlüklerindeki hali gösterilmektedir [50].



Şekil 4.13: Bir aracın farklı voxel çözünürlüklerindeki hali [50].

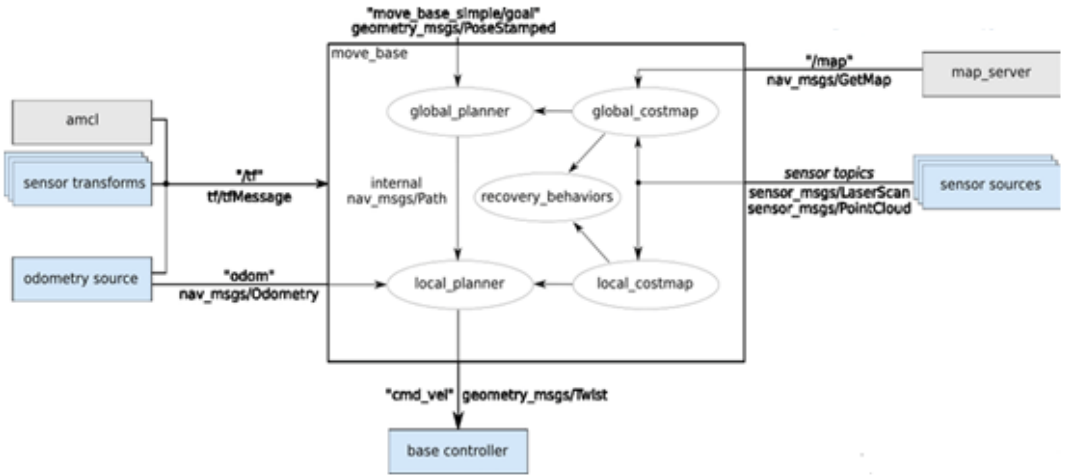
4.5 Navigasyon

Bu bölümde amaç, ROS platformu kullanarak tasarlanan mobil aracın yeni ve bilinmeyen bir ortamda navigasyon işlemini gerçekleştirmesini sağlamaktır. ROS platformu üzerinde navigasyon işlemini gerçekleştirmek için öncelikle bir haritanın oluşturulması gerekir. Harita olmadan navigasyon yapılamaz. Mobil aracın bulunduğu ortamın bir temsili olarak ifade edilen ve çevre hakkında bilgi veren harita, aracın bulunduğu konumdan başka bir konuma doğru hareket etmesi veya bir yerden bir nesneyi alıp başka bir yere götürmesi gibi eylemleri gerçekleştirebilmesi için temel bir ihtiyaçtır. Fakat iyi bir navigasyon için aracın sadece çevresi hakkında bilgi sahibi olması yeterli değildir. Aynı zamanda, yapılması istenen görevi yerine getirmek için haritaya göre hangi konumda ve hangi yönde olduğunu da bilmesi gerekir. Yerelleştirme olarak bilinen bu durum aracın oluşturulan haritaya göre konumunu tahmin etme işlemi olarak ifade edilir. Çevrenin haritasını oluşturmak ve yerelleştirme işlemini gerçekleştirmek için sistemde bir mesafe ölçüm cihazının olması gerekir. Haritalama ve yerelleştirme işlemini gerçekleştirdikten sonra aracın nereye ve nasıl gitmesini söyleyecek bir sisteme ihtiyaç vardır. Yol planlama olarak adlandırılan bu sistem temel olarak aracın mevcut yerini ve gitmek istediği konumu girdi olarak alır ve belirlenen noktaya ulaşmayı sağlayan en iyi ve en hızlı yolu çıktı olarak verir. Son olarak, mobil araç harita ve sensörlerden aldığı verileri kullanarak hareketi sırasında önünde bulunan engellere çarpmadan navigasyon işlemini gerçekleştirir. Bu bölümde navigasyon işleminin gerçekleştirilmesini sağlayan çeşitli ROS paketlerinden bahsedilecektir.

4.5.1 Navigasyon yığını (navigation stack)

Navigasyon Yığını, aracın bir noktadan diğerine otonom olarak hareket etmesi ve kaybolmadan yolunda bulabileceği tüm engellerden kaçınması için kullanılan bir dizi ROS düğüm ve algoritmasıdır. Görevi, aracın mevcut konumunu, gitmek istediği konumu, odometri verilerini (enkoder, IMU) ve lazer tarayıcı gibi sensörlerden gelen verileri almak ve bu verileri işleyerek aracı belirtilen hedefe hareket ettirmek için kullanılacak hız verilerini çıkarıp mobil tabana göndermektir. Navigasyon yığınının kullanılabilmesi için aracın ROS'u çalıştırması, bir tf dönüşüm ağacının olması ve doğru ROS ileti türleri kullanılarak sensör verilerinin yayınlanması gerekir. Ayrıca, bu paketin araç üzerinde yüksek düzeyde performans gösterebilmesi için şekil ve

dinamik olarak yapılandırılması gerekir. Şekil 4.16 navigasyon yığınının nasıl düzenlendiğini gösteren genel bir bakış sunmaktadır.



Şekil 4.16: ROS navigasyon yığını.

Şekle göre beyaz renk doğrudan otonom navigasyon için kullanılmak üzere ROS platformunda mevcut olan yığını temsil eder (move_base). Gri renkli bileşenler isteğe bağlı olarak sağlanan düğümlerdir. Mavi bileşenler ise her araç platformu için oluşturulmalıdır. Aşağıda, gezinme yığını için kullanılması gereken tüm blokların kısa açıklamaları verilmiştir.

4.5.1.1 Sensör dönüşümleri (sensor transforms): Gezinme yığını, aracın tf'yi kullanarak koordinat çerçeveleri arasındaki ilişkiler hakkında bilgi yayınlamasını gerektirir. Yani araç, ana koordinat çerçevesi ve ROS dönüşümleri kullanan farklı sensörlerin çerçeveleri arasındaki ilişkiyi yayınlamalıdır.

4.5.1.2 Sensör bilgileri (sensor sources): Sensörler navigasyonda aracın haritadaki yerini belirlemek ve engelleri tespit etmek için kullanılır. Gezinme yığını, bu sensörlerin ROS üzerinden yayınladığı sensor_msgs/LaserScan veya sensor_msgs/PointCloud mesajlarını kullanır.

4.5.1.3 Odometri bilgileri (odometry sources): Gezinme yığını, aracın dünyadaki yerini belirlemek için tf'yi kullanır ve sensör verilerini statik bir haritayla ilişkilendirir. Bununla birlikte, tf robotun hızı hakkında herhangi bir bilgi sağlamamaktadır. Bu nedenle, navigasyon yığını herhangi bir odometri kaynağının hız bilgisi içeren ROS üzerinden hem bir dönüşümü hem de nav_msgs/Odometry mesajını yayınlamasını gerektirir. Çalışmada kullanılan odometri kaynakları enkoder, ve IMU'dur.

4.5.1.4 Taban kontrolörü (base controller): Taban kontrolörünün ana işlevi, bir geometry_msgs/Twist mesajı olan navigasyon yığınının çıktısını araç için karşılık gelen motor hızlarına dönüştürmektir. Bu işlemi gerçekleştirmek için move_base paketinden çıkan “cmd_vel” başlığını dinleyen bir python kodu oluşturuldu. Buradan gelen hız verisi arduinonun anlayacağı bir protokol ile “mobil_serial_topic” başlığı altında paylaşılarak motor komutlarına dönüştürüldü.

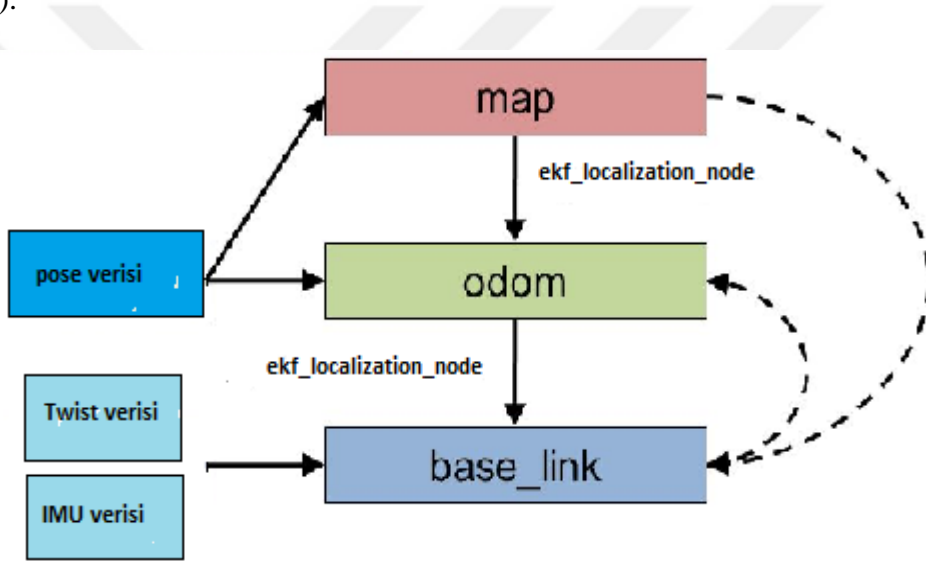
4.5.1.5 Move_base paketi: Bu paket, aracın navigasyon yığını kullanılarak istenen pozisyona taşınmasını sağlar. Yani navigasyon işleminin yapıldığı yer burasıdır. Move_base paketi ile harita üzerinden 2 boyutlu bir hedef belirlenip, aracın belirlenen yere en kısa ve engellerden kaçacak şekilde yol çizmesi (yeşil ok) ve yolu takip eden aracın hız verisinin çıktı olarak verilmesi sağlanır. Paket girdi olarak aracın yerleştirildiği konum olan “odometry/filtered” başlığını, çıktı olarak da “cmd_vel” başlığını kullanır. move_base düğümü ise navigasyon görevini gerçekleştirmek için küresel (global_planner) ve yerel (local_planner) bir planlayıcıyı birbirine bağlar. Küresel planlayıcı haritada bulunan çok uzak mesafedeki bir hedefe yollar oluşturmak için kullanılır. Yerel planlayıcı ise yakın mesafelerde yollar oluşturmak ve engellerden kaçınmak için kullanılır. ROS navigasyon yığnında yerel planlayıcı odometri mesajlarını (/odom) alır ve aracın hareketini kontrol eden hız (/cmd_vel) komutlarını çıkarır. Yerel planlayıcı olarak dwa_local_planner olarak adlandırılan ve aracın başlangıç konumundan hedef konumuna gitmesi için kinematik bir yörünge oluşturan paket kullanılmıştır. Gezinme yığını aracın bulunduğu ortamdaki engellerle ilgili bilgileri depolamak için iki çeşit maliyet haritası (costmap) kullanır. Küresel planlamada tüm çevre için uzun vadeli planlar yaratmak için bir maliyet haritası (global costmap) kullanılır. Yerel planlamada kullanılan maliyet haritasının (local costmap) amacı ise engellerden kaçınmaktır.

Bu çalışmada aracın olasılıksal olarak yerleştirmesini sağlayan AMCL (Adaptive Monte Carlo Localization), harita verilerini bir ROS servisi olarak sunan map_server ve lazer ve odometri verilerini kullanarak 2 boyutlu bir harita oluşturan Gmapping paketi kullanılmamıştır. Yapılan çalışmada Gmapping paketinin kullanılmamasının sebebi, bu paketin lazer tarayıcıdan yayınlanan scan başlığını kullanarak harita oluşturmasıdır. Bunun için nokta bulutu verisinin LaserScan verisine dönüştürülerek yatay scan verisi elde edilmelidir. Fakat bu dönüşümde elde edilen verilerin tutarsız

olmasından dolayı bu paketi kullanmak yerine navigasyon işleminin gerçekleştirildiği move_base paketinin sağladığı ve direk nokta bulutu verilerinin kullanılmasıyla elde edilen costmap haritası kullanılmıştır. ACML'nin kullanılmamasının sebebi ise bu paketin Gmapping paketinden elde edilen harita olmadan çalıştırılmamasıdır. Bundan dolayı, aracın yerleştirilmesi robot_localization adı verilen bir paket kullanıldı.

4.5.2 robot_localization paketi:

Bu paket, her biri 3 boyutlu alanda hareket eden araçlar için doğrusal olmayan bir durum tahmincisinin uygulanması olan bir durum tahmini düğümleri topluluğudur. Yerleştirme için öncelikle kullanılan sensörlerin dönüştürülmesi gerekir (Şekil 4.17).

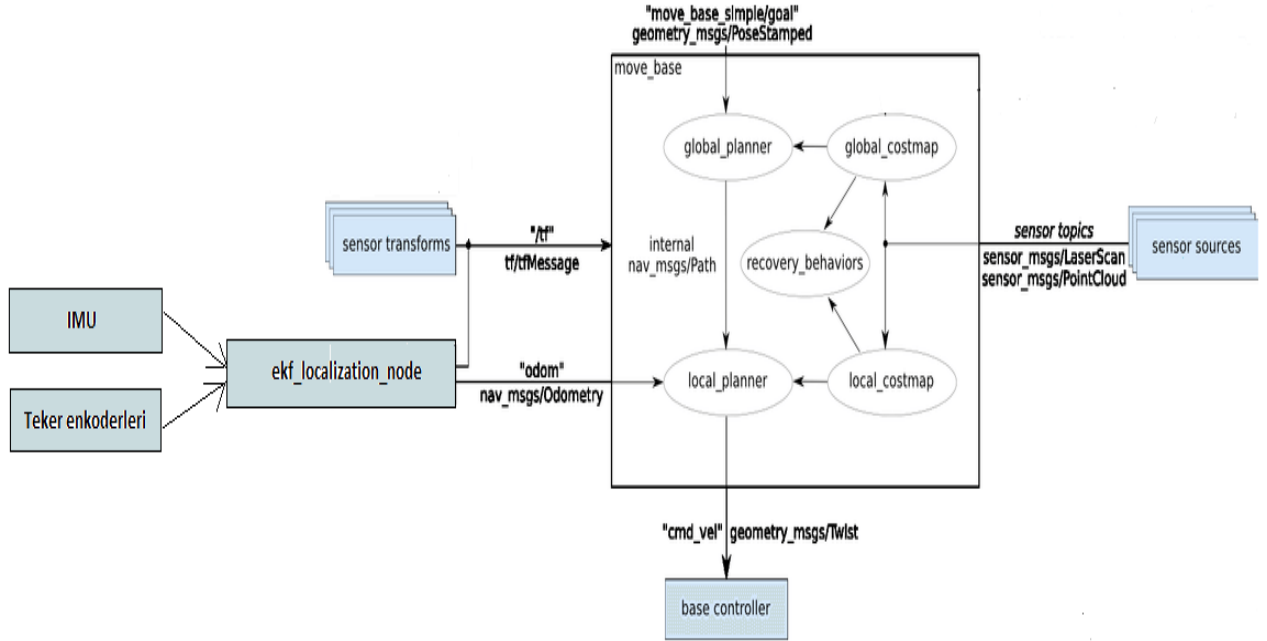


Şekil 4.17: Sensör tf dönüşümü.

Şekle göre iki adet genişletilmiş kalman filtresi (EKF) yapısı kullanıldı. Birinci yapı olan ekf-odom ile odom-base_link arasında tf oluşturularak odometry/filtered başlığında aracın odometrisi paylaşıldı. EKF-odom yapısına IMU sensöründen roll, pitch ve yaw verileri, enkoderden ise konum verisi girildi. Python kodu ile konum verisi hız verisine çevrilerek EKF'ye lineer x ve açısal z hızı sokuldu. Konum verisinin kullanılmamasının sebebi kalman filtresinin aracın hızını kullandığında daha doğru bir konumlandırma yaptığının gözlemlenmesidir. İkinci yapı olan ekf-map ile map-odom arası tf oluşturularak odometry/filtered_map adında bir başlık yayımlandı. Bu başlığın sadece tf'si kullanıldı. Bu yapıda da IMU sensörünün aynı verilerinden faydalandı. EKF-map yapısının oluşturulmasının temel amacı, nokta

bulutları kullanılarak oluşturulan maliyet haritasının (costmap) daha iyi bir konumlandırmaya sahip olmasıdır.

Sonuç olarak navigasyon yığını ve robot_localization paketinin ortak kullanımıyla mobil araç başarılı bir şekilde navigasyon işlemini gerçekleştirmiştir. Aracın başarılı bir şekilde navigasyon yapmasını sağlayan paketlerin akış şeması şekil 4.18’de gösterilmektedir .



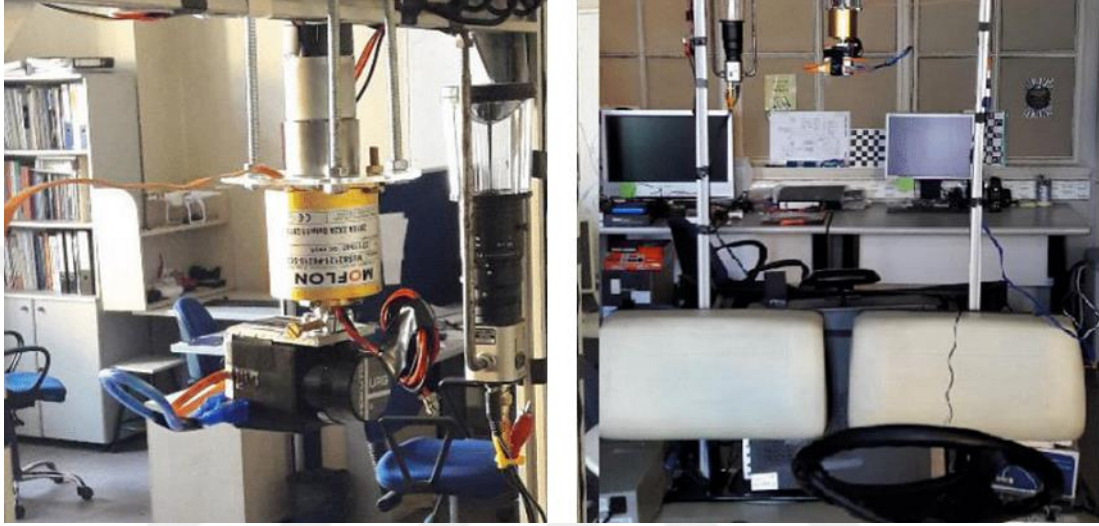
Şekil 4.18: Navigasyon akış şeması.

4.6 Deneyler

Tasarlanan 3 boyutlu LIDAR’ın tarama özelliklerini ve çevresindeki engelleri tespit edebilme kabiliyetini test etmek amacıyla öncelikle i5 işlemciye sahip bir bilgisayar kullanılmıştır. Bilgisayar kullanılarak yapılan deneylerde önce sabit bir sistem çevresindeki engeller tespit edilmiş daha sonra hareketli bir sistemde engel tespitinin sağlanması için bilgisayarı içine alabilecek düzeyde bir mobil araç tasarlanmıştır. Daha sonra aynı algoritma bilgisayara göre daha küçük boyutlu olması sebebiyle tasarlanan sistemi daha portatif hale getiren Jetson TX1 geliştirme kartı üzerinde denenmiştir. Jetson TX1 geliştirme kartı ile birlikte arduino, motor sürücü entegresi, piller ve diğer elektronik komponentleri de içinde barındıran mekanik bir sistem tasarlanıp bu sistem Şekil 3.3’te gösterilen mobil araca entegre edilmiştir ve hareketli sistem deneyleri bu mobil araç üzerinde yapılmıştır.

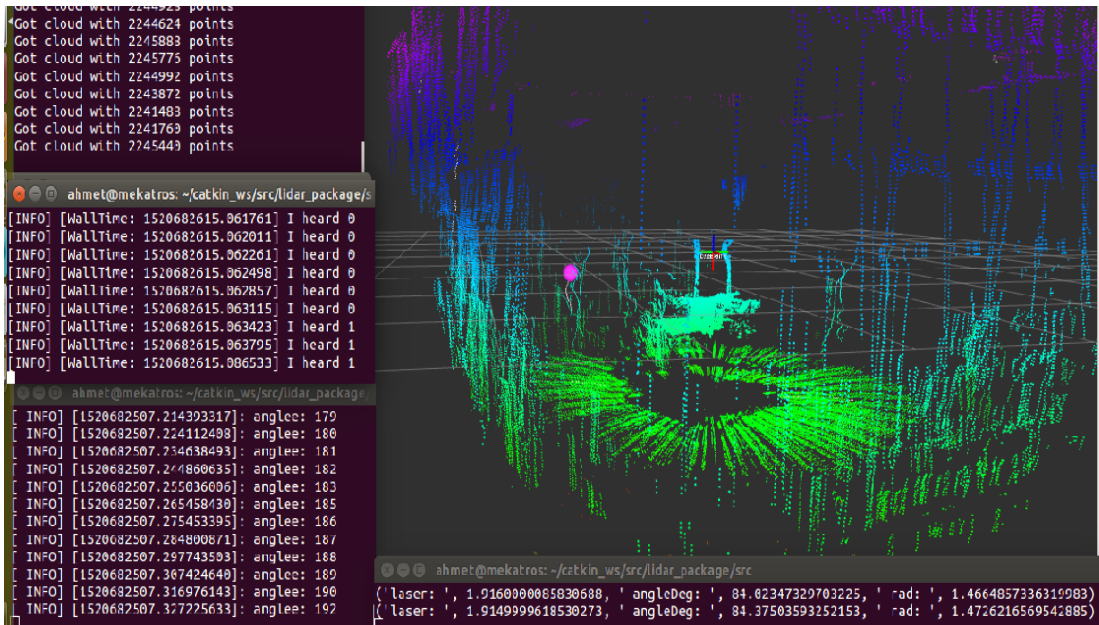
4.6.1 Bilgisayar deneyleri

Bu deneyde i5 işlemciye sahip bir bilgisayar kullanılmıştır. Geliştirilen 3B LIDAR sistemi önce otonom iç ortam golf aracı üzerine entegre edilmiştir. Bu deneydeki amaç sabit bir sistemde araca en yakın engelin mesafesinin ve açısının tespit edilmesidir. Bu yüzden sistem tam tersi yönde araca baş aşağı olacak şekilde kurulmuştur. Şekil 4.19 da engel tespitinde kullanılan deney düzeneği gösterilmiştir.



Şekil 4.19 : Otonom golf aracı üzerine entegre edilen sistem.

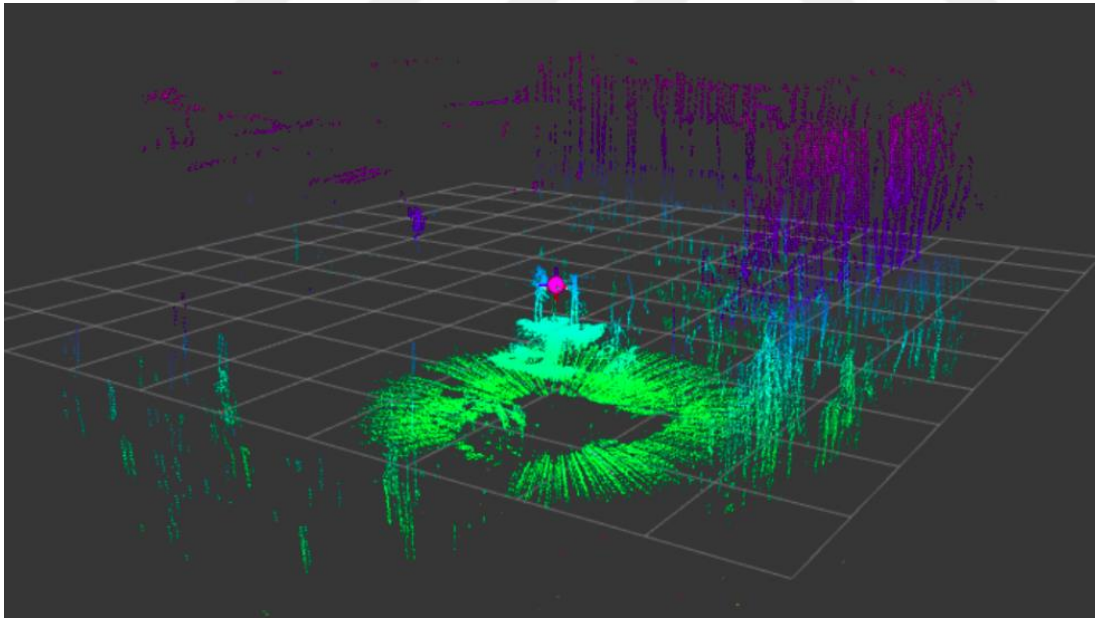
Bu deneyde öncelikle ortamın 3 boyutlu görüntüsü elde edilerek geliştirilen 3B LIDAR'ın merkezinden itibaren 5.6 metrelik mesafe aralığında bulunan bütün engellerin uzaklığı tespit edildi.



Şekil 4.20 : En yakın engel tespiti (Mor top en yakın nesneyi gösterir).

Daha sonra belirtilen aralıkta tespit edilen bütün engeller arasından sisteme en yakın engelin mesafesi ve açısı bulundu (Şekil 4.20). Fakat başlangıçta lazer tarayıcı tarafından en yakın engel olarak tarama aralığında bulunan araç ve zemin görüldü. Bu problemi çözmek için lazer tarayıcının zemin ve aracı gösteren bütün mesafe ve açı değerlerine lazerin belli açısal sınırları içindeki sensor_msgs/LaserScan noktalarını ihmal eden LaserScanAngularBoundsFilter filtrelemesi uygulandı. Bu filtreleme işleminden sonra araca en yakın engelin mesafesi ve açı değeri başarıyla tespit edildi ve bu engelin bulunduğu konum Rviz görselleştirici platformunda PointStamp noktası olarak gösterildi. Ayrıca engel tespit edildiğinde aracın stop lambasının yanması sağlanarak araca engeli uyarıcı bir özellik de kazandırılmıştır. Şekil 4.13’de Rviz platformunda PointStamp noktası olarak gösterilen en yakın engelin mesafe ve açısının yanısıra, enkoderden okunan açı değerleri, elde edilen nokta bulutu verileri ve engel algılandığında aracın stop lambasının yandığını gösteren veriler de gösterilmektedir.

Belirlenen aralıklarda eğer engel yoksa Rviz platformundaki PointStamp noktası sistemin merkezinde konumlanacak şekilde ayarlanmıştır (Şekil 4.21).



Şekil 4.21 : Engel tespit edilmediğinde PointStamp noktasının konumu.

Bu sistemin iç ortam otonom aracına entegre edilmesiyle araç artık 360⁰’lik ortamı algılayabildiğinden engel tespitinde ve engele göre karar verme mekanizması geliştirmede çok daha güvenli hale gelmiştir. Ayrıca 3m uzaklığındaki engeli 1.8cm kayma ile 3.018 metre uzaklığında algılayarak engel tespitinde %99.4’lük bir

doğruluk oranı elde edilmiştir. Bunun dışında yapılan 10cm, 50cm, 1m, 5m, 5.6m olmak üzere farklı noktadaki mesafe hassasiyet testinde de benzer doğruluk oranları elde edilmiştir.

Sabit bir platformda bilgisayar kullanılarak yapılan deneylerden sonra hareketli bir sistemde engel tespiti test edilmiştir. Bunun için bilgisayarı da içinde bulunduran bir mobil araç kullanılmıştır. Bu deneyin amacı ise hareketli bir ortamda engellerin tespit edilerek engelin konumuna göre aracın karar verme mekanizması geliştirmesidir. Şekil 4.22’ de gösterilen mobil aracın üzerine entegre edilen LIDAR sistemi golf aracının üzerine koyulduğu şekilde ters değil tarayıcının merkezi yukarıya bakacak şekilde konulmuştur.

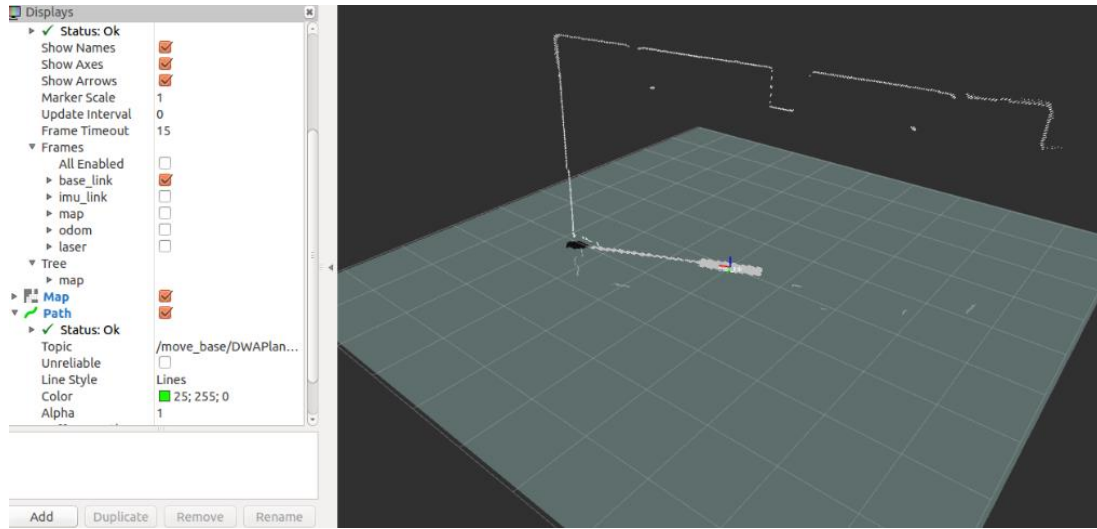


Şekil 4.22: Hareketli sistemde engel tespiti deneyinde kullanılan mobil araç.

Bu şekilde yapılan taramalarda yine en yakın engel olarak zemin ve mobil araç tespit edilmiş ve bu zemini ve mobil aracı gösteren istenmeyen mesafe değerlerinin ortadan kaldırılması için lazer tarayıcının belli açısal sınırlar içindeki değerlerini ihmal eden LaserScanAngularBoundsFilter filtrelemesi uygulanmıştır. Böylelikle mobil araç hareket halindeyken belirlenen mesafe aralığındaki en yakın engeli başarıyla tespit etmiş ve belli bir yakınlıkta tespit edilen engele göre karar verme mekanizması geliştirmesi sağlanmıştır. Ayrıca hareketli sistemde engel tespitinin doğruluğu açısından LIDAR’ın bir tam turu dönme süresi oldukça önemli olduğundan bu tarama süresi 5.5 saniyeden 3.7 saniyeye düşürülmüş ve engel tespitindeki hassasiyet arttırılmıştır.

4.6.2 Jetson TX1 deneyleri

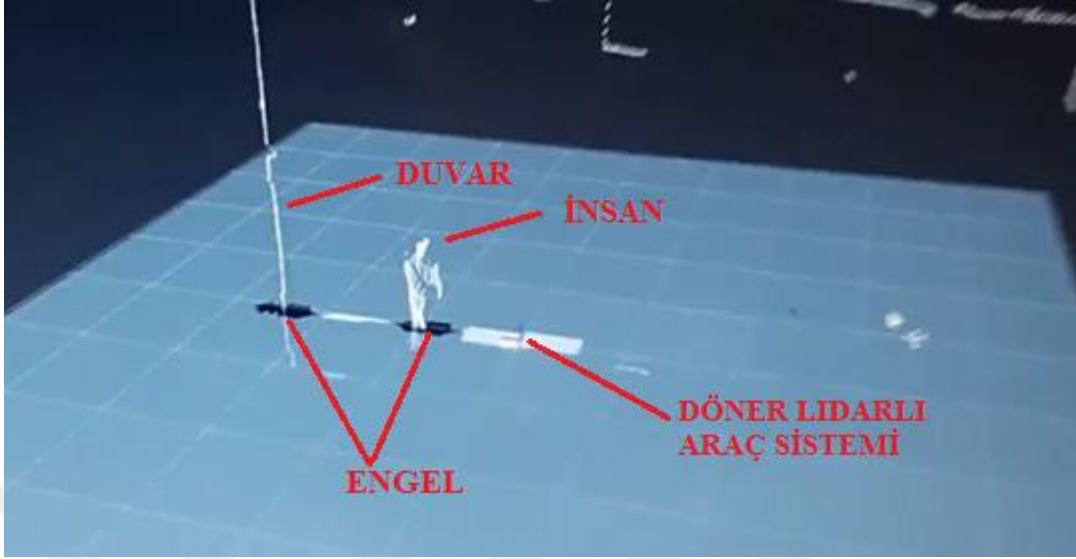
Bu deneydeki amaç, çeşitli ros yığın ve paketleri kullanılarak tasarlanan mobil araca navigasyon işleminin yaptırılmasıdır. Bu navigasyon işlemini gerçekleştirmek için 2B Lidar, tekerlek enkoderleri ve IMU olmak üzere 3 tip sensör kullanıldı. Navigasyon için gerekli olan ortamın haritası, yapısında costmap olarak adlandırılan maliyet haritası bulunduran ve aracın istenen pozisyona taşınmasını sağlayan move_base paketi kullanılarak oluşturuldu (Şekil 4.23). Şekle göre kırmızı ok aracın önünü temsil etmektedir.



Şekil 4.23: Move_base kullanılarak oluşturulan maliyet haritası (local costmap).

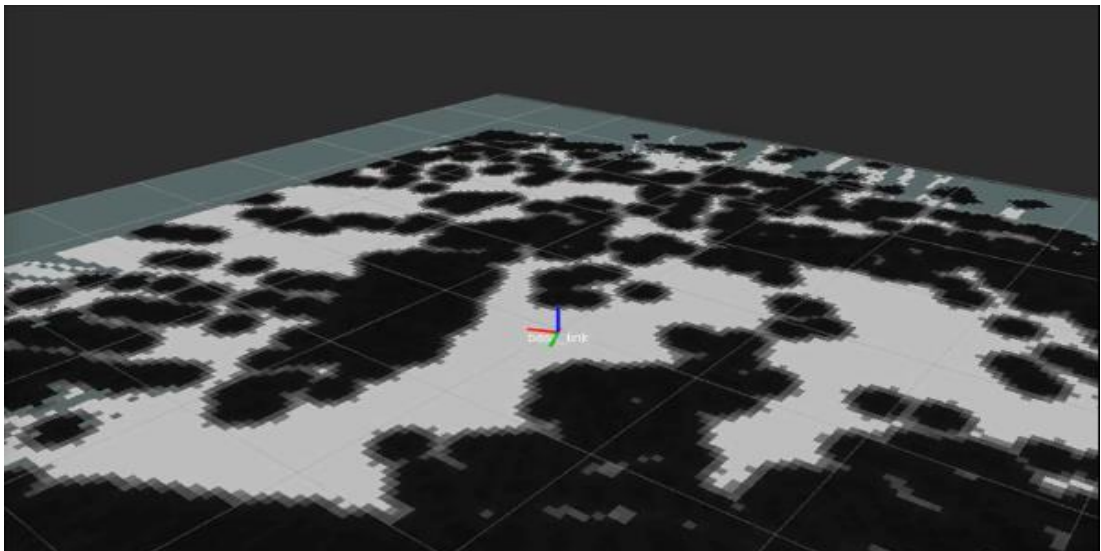
Normalde, harita oluşturma işlemi Gmapping paketi kullanılarak oluşturulur fakat Gmapping paketi lazer tarayıcının /scan başlığında yayınlanan LaserScan mesaj verisini kullanır. Bunun için, kullanılan 2B lazer tarayıcıdan elde edilen nokta bulutu verisinin LaserScan verisine dönüştürülmesi gerekir. Lazer tarayıcı sabit olduğunda doğru bir şekilde engel tespiti yapabilen bu paket, dönen bir lazer tarayıcıda kullanıldığında engellerin haritada tespit edilmesinde tutarsız sonuçlar verdi. Bundan dolayı bu paket yerine move_base paketinin içinde bulunan ve direkt nokta bulutu verisininin kullanılmasına dayanan yerel maliyet haritası (local costmap) kullanılarak sistemin haritası elde edilmiştir. Bu haritaya göre siyah renk engelleri, beyaz renk de engelsiz bir ortamı temsil eder. Tarayıcı sabitken, sistemin önünde engel olduğunda hemen engel tespit edilir ve engelin bulunduğu yerde siyah bir alan oluşur (şekil 4.24). Engel kaldırılır kaldırılmaz ise o siyah alan hiçbir gecikme olmadan ortadan kalkar. Aracın doğru bir navigasyon yapabilmesi için bu hayati öneme sahiptir. Çünkü hareketli bir sistemde engellerin hemen tespit edilmesi, aracın önündeki

engel ortadan kalktığına da hiçbir gecikme yaşanmadan haritanın kendini güncelleyerek engelin bulunduğu siyah alanı ortadan kaldırması gerekir.



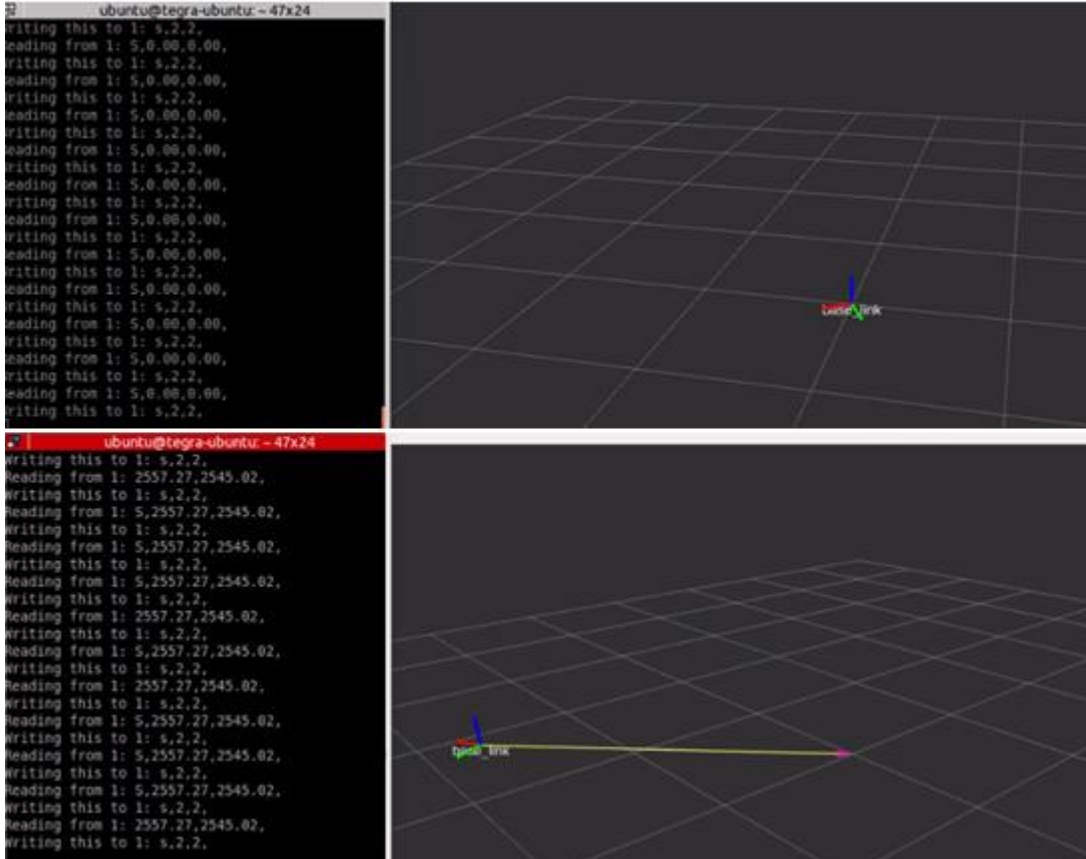
Şekil 4.24: Haritada nokta bulutlarının engeli tespit ettiğinde oluşan görüntü.

Lazer tarayıcı döndüğünde ise haritada 360° 'lik görüş alanında bulunan bütün engeller tespit edilmiştir (Şekil 4.25). Fakat mobil araç hareket etmeye başladığında güncellenen haritaya göre dönen LIDAR'ın engelleri tespit edebilmesinde yaklaşık 3 saniyelik bir gecikme yaşanmıştır. Yani, araç engeli geçtikten sonra haritada yaklaşık 3 saniye o engelin bulunduğu siyah alan ortadan kalkmamaktadır. Bu durum navigasyon işlemi için ciddi bir sorundur. Aracın hızı yavaşlatılıp, lazer tarayıcının dönme hızı artırılarak yaşanan bu gecikme süresi 1.5 saniyeye kadar indirilmiştir.



Şekil 4.25: Lazer tarayıcının dönmesiyle haritada tespit edilen engeller (siyah alanlar).

Aracın yerleştirilmesi için robot_localization paketi kullanıldı. İyi bir yerleştirmenin önemli parametrelerinden biri olan odometri, aracın sadece tekerleklerinin dönüşünden kendi konumunu ne kadar iyi tahmin edebileceğini ölçer. Yapılan çalışmada araç yerleştirilmesi için enkoderden elde edilen konum verisi kullanılmadı. Konum verisinin türevi alınıp birim dönme hızı bulunarak tekerin bu birim dönme hızından lineer x ve açısal z hızı elde edildi ve kalman filtresine gönderildi. Çünkü yerleştirme için kullanılan kalman filtresinin aracın hız parametresini kullandığında çok daha iyi bir yerleştirme yaptığı tespit edildi. İyi bir yerleştirme için araç, gittiği her 1 metre yol için 2 cm'den fazla bir hataya sahip olmamalıdır. Klavye kullanılarak araç kontrolü sağlayan turtlebot_teleop paketi kullanılarak yapılan yerleştirme deneyinde aracın gerçek ortamda aldığı mesafe ile teker enkoderlerinden okunan mesafe verisi karşılaştırılmıştır. Yapılan karşılaştırma sonucunda araç gerçekte 2.514 m mesafe katederken, teker enkoder verisinden okunan mesafe verisi ise 2.557 m olarak ölçülmüştür (şekil 4.26).

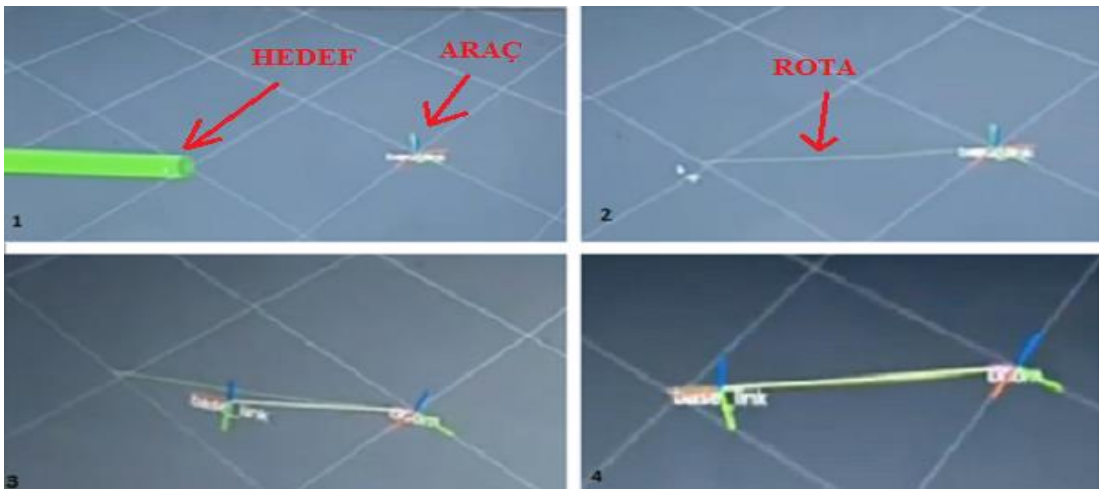


Şekil 4.26: Araç yerleştirme deneyi.

Elde edilen enkoder mesafe verileri mm cinsindedir. Sol tekerden okunan mesafe değeri (2.557m) ile sağ tekerden okunan mesafe değeri (2.545m) arasındaki

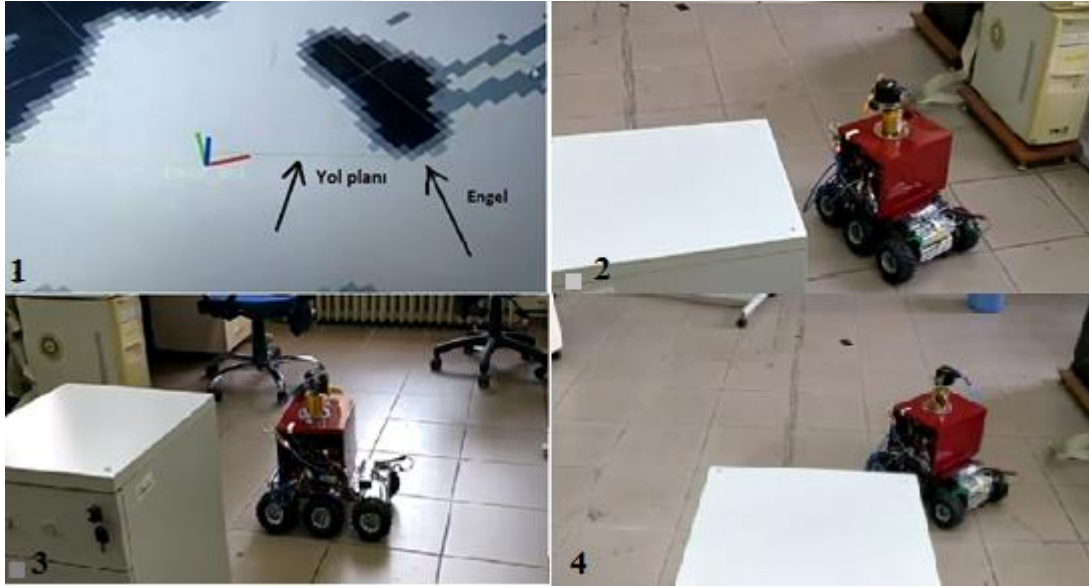
1.2cm'lik farkın sebebi sağ teker dişlisindeki boşluktan dolayıdır. Yapılan deneyden elde edilen verilere göre aracın aldığı her 1 metre yol için 1.68 cm'lik bir hata söz konusudur.

Move_base paketi kullanılarak yapılan yerelleştirme deneyinde ise yerel maliyet haritası (local costmap) üzerinde aracın gitmesi istenen hedef belirlenir. Hedef belirlendikten hemen sonra Rviz penceresinde görülebilen yeşil bir çizgi oluşur. Bu yeşil çizgi aracın mevcut konumundan hedef konumuna gitmesi için hesaplanan yoldur. Araç bu yol üzerinde hareket etmeye başlayacaktır. Yapılan deneyde aracın mevcut konumundan belirlenen hedef konumuna doğru otonom olarak doğru bir şekilde gidip gitmediği test edilmiştir. Şekil 4.27'de gösterilen 1 nolu resimde aracın gitmesi istenen hedef manuel olarak Rviz platformu üzerinde işaretlenmiştir. 2 no'lu resimde ise hedef işaretlendikten hemen sonra aracın işaretlenen hedefe nasıl gideceğini belirten yol planı elde edilmiştir. 3 ve 4 no'lu resimlerde ise aracın işaretlenen hedefe doğru hesaplanan rota üzerindeki hareketi gösterilmiştir. Şekilde görüldüğü üzere araç hedef konuma doğru başarılı bir şekilde hareket etmiş fakat belirlenen hedef ile gerçekte ulaştığı hedef arasında 4.7 cm'lik fark tespit edilmiştir. Ulaştığı hedeften tekrar mevcut konumuna geri dönmesi istendiğinde ise bu fark 9.5 cm olarak belirlenmiştir. Aracın mevcut konumuna dönerken elde edilen farkın çok daha fazla olmasının sebebinin navigasyon yığınının davranışından kaynaklandığı tespit edildi. Çünkü hedef TF orjinine göre +x yönünde ayarlandığında dwa yerel planlayıcısı kararlı plan yaptı ve araç hızlı bir şekilde hareket ederek hedefe ulaştı. Fakat hedef TF orjinine göre -x yönünde ayarlandığında dwa yerel planlayıcısı çok daha az kararlı plan yaptı ve araç çok yavaş hareket etti.



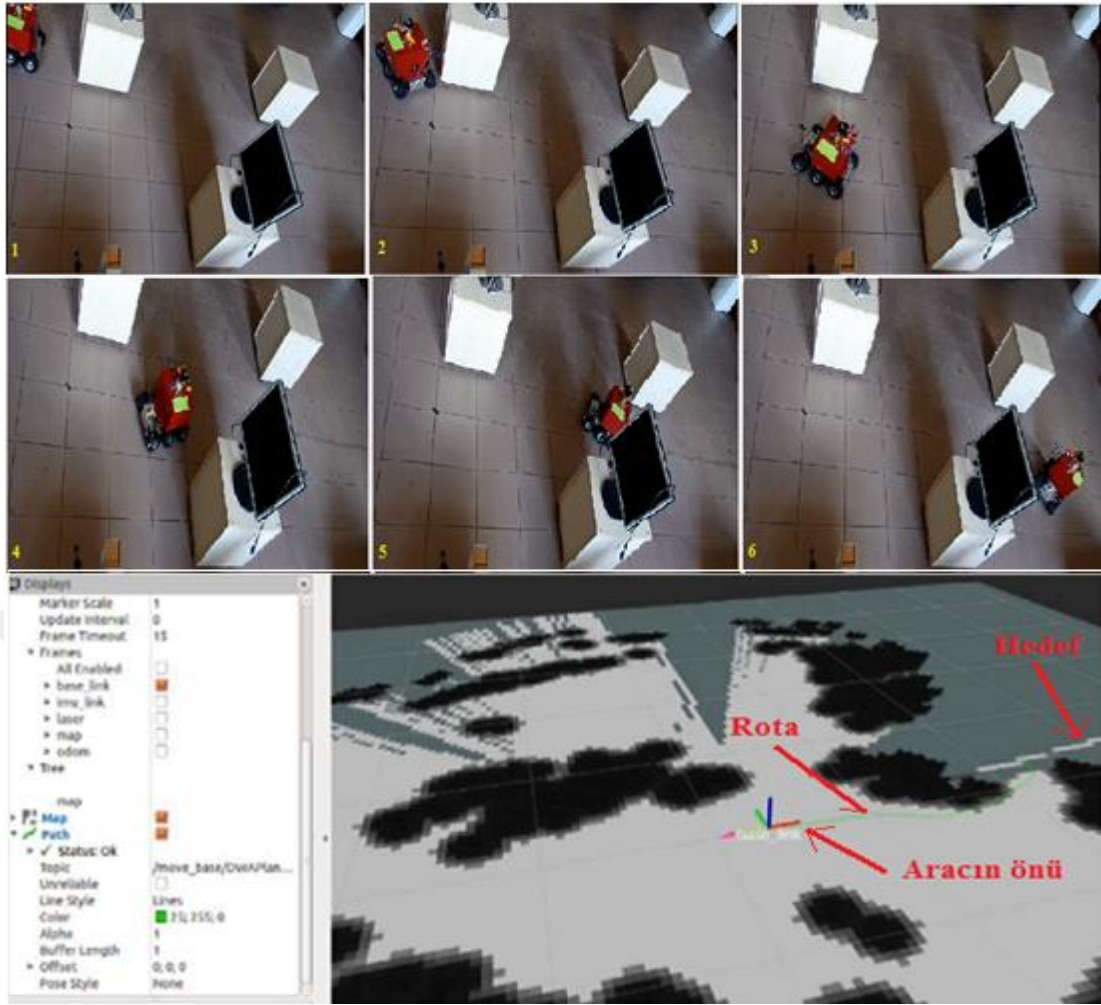
Şekil 4.27: Move_base kullanarak araç yerelleştirme deneyi.

Haritalama ve yerelleştirme deneyinden sonra araç üzerinde navigasyon testi yapıldı (Şekil 4.28). Bu navigasyon testi ile araç belirlenen hedefe önündeki engellere çarpmadan başarılı bir şekilde ulaşması için navigasyon yığımında bulunan bazı parametreler araca göre düzenlendi. Bu testteki amaç aracın bir engelin arkasında bulunan hedefe engele çarpmadan ulaşabilmesinin sağlanmasıdır. Bunun için öncelikle önünde sadece bir engel bulunan basit bir hedef belirlenmiş ve bu hedefe göre araç, 1 no'lu resimde görüldüğü gibi bir yol planı belirleyip (yeşil ok) o hedefe doğru hareket etmeye başlamıştır. Araç belirlenen hedefe engele çarpmadan ulaşmasına rağmen araç hareket ettikçe güncellenen haritaya göre dönen LIDAR'ın engelin konumunu tespit edebilmesinde yaşanan saniyelik gecikmeler tam verimli bir navigasyon işleminin gerçekleşmesine engel teşkil etmiştir. Bundan dolayı aracın hızlı hareket ettiğinde LIDAR'ın engeli güncellenen haritaya göre belirlemede yetersiz kaldığı sonucuna varılmıştır. Fakat aracın hızı yavaşlatılıp LIDAR'ın dönüş hızı artırıldığında LIDAR'dan yayınlanan nokta bulutu sayılarınının da düşürülmesiyle navigasyon işleminde ciddi iyileştirmeler sağlanmıştır.



Şekil 4.28: Navigasyon deneyi 1.

Şekil 4.29'da ise daha karmaşık ve birden fazla engelin olduğu bir hedef belirlenmiş ve belirlenen bu hedefe göre aracın hareketi test edilmiştir. Aracın yerel maliyet haritasındaki hareketi, yol planı ve gerçek ortamdaki hareketi şekilde gösterilmektedir. Laboratuvar ortamında yapılan bu deneye göre de aracın önündeki engellere çarpmadan belirlediği rota çerçevesinde işaretlenen hedefe ulaştığı görülmektedir.



Şekil 4.29: Navigasyon deneyi 2.



5. SONUÇ VE ÖNERİLER

5.1 Sonuçlar

Son yıllarda otonom araçlar üzerinde yapılan çalışmalar oldukça artmaktadır ve engel tespiti, haritalama, lokalizasyon, yol planlaması gibi alanlarda ciddi çalışmalar yürütülmektedir. Bu çalışmalarda kullanılan LIDAR, IMU, GPS, Radar, kamera gibi algılayıcılar araçların otonomluğuna ciddi katkı sağlamaktadır.

Bu algılayıcılardan biri olan LIDAR, engelleri tespit etmek ve çevre hakkında bilgi sağlamak amacıyla kullanılan yüksek çözünürlüklü, yansıtıcı ve zor yüzeylerle başa çıkabilme ve çok küçük nesnelere bile tespit edebilme özelliğine sahiptir. Fakat çok yüksek maliyetli ve büyük boyutlarda olmalarından dolayı Google, Apple gibi büyük şirketler dışında kullanılmaları çok zordur.

Bu tez çalışmasında aracın çevre ile olan iletişimde büyük avantaj sağlayan yüksek maliyetli büyük boyutlu 3 boyutlu LIDAR'lar yerine 2 boyutlu lazer tarayıcı kullanılarak daha düşük maliyetli daha küçük ve 360⁰ görüş alanına sahip yeni bir 3 boyutlu LIDAR tasarımı yapılmıştır. 2 boyutlu lazer tarayıcıdan 3 boyutlu LIDAR elde edilme aşamaları hakkında bilgi verilmiştir ve bu LIDAR tasarımı esnasında kullanılan komponentler hakkında bilgi verilerek kullanım amaçları açıklanmıştır. Elde edilen bu yeni 3 boyutlu LIDAR sistemiyle kapalı bir ortamın 3 boyutlu görüntüsü elde edilerek ortam hakkında bilgi edinilmiştir. Geliştirilmeye çalışılan yeni LIDAR sistemini istenen verimlilikte daha istikrarlı ve güvenli bir şekilde çalıştırabilmek, yazılımsal ve donanımsal haberleşmeyi yönetmek amacıyla ROS olarak adlandırılan robot işletim sistemi kullanılmıştır.

Yapılan deneylerde öncelikle i5 işlemci hızına sahip bir bilgisayar kullanılmıştır. Çünkü 3 boyutlu görüntü elde edilebilmesi için gereken nokta bulutu verilerinin akış hızının yüksek olması elde edilen görüntünün daha net olmasını sağlar. Bilgisayar kullanılarak geliştirilen bu sistem otonom kapalı ortam golf aracına entegre edilmiş ve hareketsiz bir ortamda araca en yakın engelin mesafe ve açısı tespit edilmiştir. Tarayıcının engel olarak araç veya zemini algılamaması tarayıcı üzerinde çeşitli

filtrelemeler uygulanmıştır. Yapılan bu filtrelemelerden sonra tarayıcının minimum veri kaybıyla bir tam turu dönüş süresinin azaltılması sağlandı. En yakın engelin tespit edilmesinde % 99.4 mesafe doğruluk oranı elde edilmiştir. Geliştirilen 3 boyutlu LIDAR sistemi hareketsiz kapalı ortam golf aracında kullanıldıktan sonra sisteme hareket vermek amacıyla bir mobil araç tasarlandı. Bu hareketli araç vasıtasıyla en yakın engelin konumunun hareketli bir ortamda tespit edilmesi amaçlanmıştır. Bu çalışmada da lazer tarayıcının engel olarak araç ve zemini görmemesi için tarayıcının belli sınırları arasına çeşitli filtrelemeler uygulanmıştır. Bu filtrelemeler sonucunda LIDAR sistemi belirlenen aralıktaki en yakın engeli tespit etmiş ve bu engele göre karar verme mekanizması geliştirmiştir.

Bilgisayar kullanılarak elde edilen 3 boyutlu nokta bulutu verisi ve yapılan deneylerden sonra küçük boyutundan dolayı sistemin daha portatif hale gelmesine imkân sağlayan yüksek performanslı bir geliştirme kartı olan Jetson TX1 kullanıldı. Jetson TX1 geliştirme kartını ve 3 boyutlu görüntü için gereken bütün komponentleri içinde barındıran mekanik bir model tasarlandı ve daha portatif bir 3B LIDAR sistemi elde edildi. Bilgisayarı taşımak için kullanılan mobil araçtan sonra yeni oluşturulan portatif LIDAR platformuna hareket kazandırarak navigasyon işleminde kullanmak amacıyla 6 tekerlekli diferansiyel sürüş modeline sahip ve ortadaki iki tekerinde enkoderli dc motor bulunan yeni bir mobil araç tasarlandı. 3B LIDAR sistemi tasarlanan bu mobil araç üzerine entegre edildi. Daha sonra birçok robotik sisteme kolayca entegre edilebilen OctoMap paketi sayesinde nokta bulutu verileri kullanılarak ortamın 3 boyutlu haritası elde edildi.

Navigasyon için Navigasyon Yığını (Navigation Stack) olarak adlandırılan, aracın bir noktadan diğerine otonom olarak hareket etmesi ve hareket esnasında önüne çıkan bütün engellerden kaçınması için kullanılan bir dizi algoritma ve ros düğümü kullanıldı. Mobil araca navigasyon işleminin yaptırılabilmesi için öncelikle araç üzerinde bulunan sensörlerden elde edilen veriler kullanılarak aracın bulunduğu ortamın bir haritası oluşturuldu. Haritalama işleminden sonra ROS platformunda yerleştirme olarak ifade edilen, aracın haritaya göre konum ve yön bilgisi veren konum kestirimi işlemi gerçekleştirildi. Daha sonra aracın mevcut yerini ve gitmek istediği konumu girdi olarak alan ve belirlenen noktaya en iyi ve en hızlı bir şekilde gidilmesini sağlayan yolu çıktı olarak veren yol planlaması işlemi gerçekleştirildi. Son olarak kullanılan gezinme yığını algoritması sayesinde aracın odometri verilerini

(IMU ve Enkoder) ve lazer tarayıcı verisini kullanarak engellere çarpmadan ve kaybolmadan belirlenen hedefe ulaşması sağlanmıştır. Fakat dönen 3B LIDAR kullanılarak gerçekleştirilen bu navigasyon deneyinde araç hareket halinde iken güncellenen haritaya göre LIDAR'ın engelleri tespit etmesinde yaşanan gecikmelerden dolayı mevcut durumda araç hızlı hareket ettiğinde tam verimli bir navigasyon işlemi gerçekleştirilememiştir. Aracın hızının yavaşlatılıp, LIDAR'dan yayınlanan nokta bulutu verilerinin sayısı düşürülüp, LIDAR'ın dönme hızı artırılarak yaşanan bu gecikmelerden doğan sorunlar minimuma indirilmiş fakat yine de iyileştirilme ihtiyacı olan bir navigasyon işlemi gerçekleştirilmiştir.

Yapılan bu deneyler sonucunda, geliştirilen bu sistem sayesinde otonom araç, çevresindeki engelleri tüm yönlü olarak tespit edebilecektir. Son olarak yüksek maliyetlerinden dolayı otonom araç araştırmalarında kullanımı mümkün olmayan 3B LIDAR'lar yerine üretilen bu daha ucuz ve portatif 3B LIDAR sisteminin kullanımının yaygınlaşması öngörülmektedir. Çünkü bu sistem 360° görüş alanında engel tespiti için ekonomik ve etkili bir çözümdür.

5.2 Öneriler

Kullanılan lazer tarayıcı elde edilen 3 boyutlu görüntünün netliğinde, engellerin yüksek hassasiyette tespit edilmesinde oldukça önemlidir. Bu yüzden bu tez çalışmasında lazer tarayıcının yerine hem daha net görüntü elde edilebilmesi hem de engellerin çok daha yüksek hassasiyette ve mesafe aralığında tespit edilebilmesi için daha yüksek çözünürlüğe, daha geniş tarama açısına, daha büyük mesafe aralığına ve daha düşük tarama süresine sahip 2B lazer tarayıcı kullanılmalıdır. Ayrıca çalışmada kullanılan lazer tarayıcı sadece iç ortamda çalışmaktadır. Belirtilen bu özelliklerin dışında dış ortamda da çalışma yapabilmek için hokuyo firmasının ürettiği UTM-30LX algılayıcısı kullanılabilir.

3B LIDAR elde edilmesinde kullanılan dc motor ve slip-ring'in aynı özelliklere sahip daha küçük modelleri kullanılırsa tasarlanan sistem daha hafif ve kullanışlı hale gelebilir.

Dikdörtgensel bir yapıda tasarlanan mobil araç, kare veya dairesel olması durumunda çok daha yüksek bir navigasyon işlemi gerçekleştirebilir.

5.3 Gelecek Çalışmalar

Gelecekteki çalışmalar gerçekleştirilen navigasyon işleminin iyileştirilmesini ve geliştirilen sisteme engeli tanıma özelliği kazandırılmasını içermektedir.



KAYNAKLAR

- [1] **Url-1** <<https://www.army-technology.com/projects/irobot-510-packbot-multi-mission-robot/>>, erişim tarihi: 28.03.2018
- [2] **Url-2** <<https://spectrum.ieee.org/tech-history/heroic-failures/selfdriving-cars-were-just-around-the-cornerin-1960/>>, erişim tarihi: 28.03.2018
- [3] **Url-3** <https://alumni.stanford.edu/get/page/magazine/article_id=67438>, erişim tarihi: 29.03.2018
- [4] **Url-4** <<https://www.web2carz.com/autos/car-tech/6396/the-vamors-was-the-worlds-first-real-deal-autonomous-car>>, erişim tarihi: 29.03.2018
- [5] **Url-5** <https://www.cs.cmu.edu/~tjochem/nhaa/navlab5_details.html>, erişim tarihi: 29.03.2018
- [6] **Url-6** <<https://www.herox.com/crowdsourcing-news/159-the-drive-for-autonomous-vehicles-the-darpa-grand>>, erişim tarihi: 29.03.2018
- [7] **Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C.M., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.E., Koelen, C., Markey, C., Rummel, C., Niekirk, J.V., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A.V. ve Mahoney, P. (2006).** Stanley: the robot that won the DARPA Grand Challenge, Journal of Field Robotics 23(9), 1-43.
- [8] **Url-7** <<https://inhabitat.com/googles-self-driving-koala-cars-hit-california-roads/google-koala-car/>>, erişim tarihi: 30.03.2018
- [9] **Url-8** <<https://www.safaribooksonline.com/library/view/ros-robotics-projects/9781783554713/ch10s02.html>>, erişim tarihi: 30.03.2018
- [10] **Url-9** <<https://www.itv.com/news/central/update/2015-01-01/how-does-a-driverless-car-work/>>, erişim tarihi: 01.04.2018
- [11] **Url-10** <www.gpsworld.com/urban-gps-navigation-improved-50-90-percent-researchers-say/>, erişim tarihi: 02.04.2018
- [12] **Url-11** <www.velodynelidar.com/terms_of_use.html>, erişim tarihi:01.04.2018
- [13] **Ebrahim, M. (2016).** 3D Laser Scanners: History, Applications and Future, LAP LAMBERT Academic Publishing, Saarbrücken, 104 pages.
- [14] **Chen, X., Ma, H., Wan, J., Li, B., Xia, T. (2017).** Multi-View 3D Object Detection Network for Autonomous Driving, IEEE Conference on

Computer Vision and Pattern Recognition (CVPR), (pp.1907-1915). Honolulu, Hawaii, United States, July 21-26.

- [15] **Li, B., Zhang, T., Xia, T.** (2016). Vehicle Detection from 3D Lidar Using Fully Convolutional Networks, Conference: Robotics: Science and Systems XII, (pp. 1608-1617). United States: University of Michigan, June 18-22.
- [16] **Kim, J.U., Min, J., Kong, H.B.,** (2017). 3D Object Detection Method Using Lidar Information in Multiple Frames, International Conference on Image Analysis and Processing, (pp. 276-286). Italy: Catania, September 11-15.
- [17] **Abdul Rachman, A.S.** (2017). 3D- Lidar Multi Object Tracking for Autonomous Driving. (Yüksek Lisans Tezi). Delft Teknoloji Üniversitesi, Makine Mühendisliği Fakültesi, Delft.
- [18] **Chen, X., Deng, Z.** (2015). Detection of Road Obstacles Using 3D Lidar Data via Road Plane Fitting, Proceedings of the 2015 Chinese Intelligent Automation Conference, (Vol. 338, pp. 441-449). Retrieved from http://www.springer.com/gp/book/9783662464656?wt_mc=GoogleBooks.GoogleBooks.3.EN&token=gbgen#otherversion=9783662464663
- [19] **Asvadi, A., Garrote, L., Premebida, C., Peixoto, P., Nunes, U.J.,** (2017). DepthCN: Vehicle detection using 3D LIDAR and ConvNet, IEEE 20th International Conference on Intelligent Transportation Systems, (pp. 1-6), Japan: Yokohama, October 16-19.
- [20] **Li, J., Bao, H., Han, X., Pan, F., Pan, W., Zhang, F., Wang, D.,** (2017). Real time self-driving car navigation and obstacle avoidance using mobile 3D laser scanner and GNSS, Multimedia Tools and Applications, 76, 21, pp. 23017-23039.
- [21] **Zhao, M., Stasinopoulos, S., Yu, Y.,** (2017). Obstacle detection and avoidance for autonomous bicycles, 13th IEEE Conference on Automation Science and Engineering, (pp. 1310-1315), China: Xi'an, August 20-23.
- [22] **Liang, W., Zhang, Y., Wang, J.,** (2017). Map- Based Localization Method for Autonomous Vehicles Using 3D-LIDAR, IFAC-PapersOnLine, 50 (1), 276-281.
- [23] **Dube, R., Gawel, A., Sommer, H., Nieto, J., Siegwart, R., Cadena, C.,** (2017). An online multi-robot SLAM systems for 3D LIDARs, IEEE/RSJ International Conference on Intelligent Robots and Systems, (pp. 1004-1011), Canada: Vancouver, September 24-28.
- [24] **Pierzchala, M., Giguere, P., Astrup, R.,** (2018). Mapping forests using an unmanned ground vehicle with 3D LIDAR and graph SLAM, Computers and Electronics in Agriculture, 145, 217-225.
- [25] **Wolcott, R.W., Eustice, R.M.,** (2014), Visual localization within LIDAR maps for automated urban driving, IEEE/RSJ International Conference on Intelligent Robots and Systems, (pp.176-183), USA: Chicago, September 14-18.

- [26] **Kagami, S., Thompson, S., Samejima, I., Hamada, T., Kato, S., Hatao, N., Nihei, Y., Egawa, T., Takeda, K., Takemura, H., Mizoguchi, H.,** (2013). Autonomous Vehicle Navigation by Building 3D Map and by Detecting Human Trajectory using LIDAR, IEEE International Conference on Cyber-Physical-Systems, Network and Applications, pp. 3-6.
- [27] **Pfrunder, A., Borges, P.V.K., Romero, A.R., Catt, G., Elfes, A.,** (2017). Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D LIDAR, IEEE International Conference on Intelligent Robots and Systems, (pp. 2601-2608), Canada: Vancouver, September 24-28.
- [28] **Naes, T.,** (2017). Ego-localization Navigation for Intelligent Vehicles using 360⁰ LIDAR Sensor for Point Cloud Mapping, (Yüksek Lisans Tezi), Gürcistan Güney Üniversitesi, Makina Mühendisliği Fakültesi, Gürcistan.
- [29] **Yoshioka, M., Suganuma, N., Yoneda, K., Aldibaja, M.,** (2017). Real-time object classification for autonomous vehicle using LIDAR, IEEE International Conference on Intelligent Informatics and Biomedical Sciences, (pp. 210-211), Japan: Okinawa, November 24-26.
- [30] **Stiller, C., Ziegler, J.,** (2012). 3D perception and planning for self-driving and cooperative automobiles, 9th International Multi-Conference on Systems, Signals and Devices, (pp. 1-7), Germany:Chemnitz, March 20-23.
- [31] **Choi, J.,** (2014). Hybrid map-based SLAM using a Velodyne laser scanner, IEEE 17th International Conference on Intelligent Transportation Systems, (pp. 3082-3087), China:Qingdao, October 8-11.
- [32] **Morales, J., Martinez, J.L., Anthony, M., Reina, A.J., Seron, J., Cerezo, A.G.,** (2013). Improving 3D scan matching time of the coarse binary cubes method with fast spatial subsampling, 39th Annual Conference of the IEEE Industrial Electronics Society, (pp. 4168-4173), Austria: Vienna, November 10-13.
- [33] **Yin, H., Berger, C.,** (2017). Mastering data complexity for autonomous driving with adaptive point clouds for urban environments, IEEE Intelligent Vehicle Symposium, (pp.1364-1371), USA: Los Angeles, June 11-14.
- [34] **Url-12** <<https://cleantechnica.com/2016/07/29/tesla-google-disagree-lidar-right/>>, erişim tarihi: 05.04.2018
- [35] **Martinez, J.L., Morales, J., Reina, A.J., Mandow, A., Boter, A.P., Cerezo, A.G.,** (2015). Construction and calibration of low-cost 3D laser scanner with 360⁰ field of view for mobile robots, IEEE International conference on Industrial Technology, (pp. 149-154), Spain: Seville, March 17-19.
- [36] **Morales J., Mrtinez, J.L.,Mandow, A., Boter, A.P., Cerezo, A.G.,** (2011). Design and Development of a Fast and Precise Low-Cost 3D Laser Rangefinder, IEEE International Conference on Mechatronics, (pp. 621-626), Turkey: Istanbul, April 13-15.

- [37] **Maurelli, F., Droeschel, D., Wisspeintner, T., May, S., Surmann, H.,** (2009). A 3D laser scanner system for autonomous vehicle navigation, International Conference on Advanced Robotics, (pp. 1-6), Germany: Munich, June 22-26.
- [38] **Kurnianggoro, I., Hoang, V.D., Jo, K.H.,** (2015). Calibration of a 2D Laser Scanner System and Rotating Platform using a Point-Plane Constraint, Computer Science and Information Systems, 12 (1), 307-322.
- [39] **Wang, C., Meng, L., She, S., Mitchell, I.M., Li, T., Tung, F., Wan, W., Meng, M.Q.H., Silva, C.W.,** (2017). Autonomous mobile robot navigation in uneven and unstructured indoor environments, IEE/RSJ International Conference on Intelligent Robots and Systems, (pp.109-116), Canada: Vancouver, September 24-28.
- [40] **Quinley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibz, J., Berger, E., Wheeler, R., Ng, A.,** (2009). ROS: an open-source Robot Operating System, ICRA workshop an open source software, 3 (1), 1-6.
- [41] **Ocando, M.G., Certad, N., Alvarado, S., Terrones, A.,** (2017). Autonomous 2D SLAM and 3D mapping of an environment using a single 2D LIDAR and ROS, Latin American Robotics Symposium (LARS) and Brazilian Symptoisum on Robotics (SBR), (pp. 1-6), Brazil: Curitiba, November 8-11.
- [42] **Chen, M., Yang, S., Yi, X., Wu, D.,** (2017). Real-time 3D mapping using a 2D laser scanner and IMU aided visual SLAM, IEEE International Conference on Real-time Computing and Robotics (RCAR), (pp.297-302), Japan:Okinawa, July 14-18.
- [43] **Zhang, A., Hu, S., Chen, Y., Liu, H., Yang, F., Liu, J.,** (2008). FAST CONTINUOUS 360 DEGREE COLOR 3D LASER SCANNER, The International Archives of Photogrammetry, Remote Sensing and Information Sciences, 37 (B1), 409-414.
- [44] **Url-12** <<https://autonomoustuff.com/product/hokuyo-urg-04lx-ug01/>>, erişim tarihi: 23.04.2018
- [45] **Url-13** <<https://www.robotshop.com/media/files/pdf/hokuyo-urg-04lx-ug01-specifications.pdf> >, erişim tarihi: 23.04.2018
- [46] **Url-14** <<https://arstechnica.com/cars/2017/01/googles-waymo-invests-in-lidar-technology-cuts-costs-by-90-percent/> >, erişim tarihi: 24.04.2018
- [47] **Url-15** <https://anibal.gyte.edu.tr/hebe/Ab1Drive/74611106/w/Storage/106_010_2_621_74611106/Downloads/lidarverilerininyenilenmesi-ve-analizidersnotlar.pdf?>>, erişim tarihi: 25.04.2018
- [48] **Walf, O., Wagner, B.,** (2003). Fast 3D Scanning Methods for Laser Measurement Systems, Preceedings of the International Conference on Control Systems and Computer Science, 312-317.
- [49] **Url-16** <https://www.codeproject.com/Articles/108761/Octrees>, erişim tarihi: 11.07.2018
- [50] **Url-17** <www.bilderzucht.de/blog/3d-pixel-voxel/>, erişim tarihi: 11.07.2018

ÖZGEÇMİŞ



Ad-Soyad : Ahmet KAĞIZMAN
Doğum Tarihi ve Yeri : 07/02/1988 - SİVAS
E-posta : kagizman@itu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans** : 2014, Okan Üniversitesi, Mühendislik-Mimarlık Fakültesi, Mekatronik Mühendisliği

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN YAYINLAR

Kağızman, A., Altuğ, E. (2018). Acquiring 3D point cloud data and object detection using 2D laser scanner (LIDAR) for autonomous vehicles. OTEKON 2018, 9th International Automotive Technologies Congress, 7-8 Mayıs, Bursa, Türkiye.

